PEDECIBA Informática
Instituto de Computación – Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

Tesis de Maestría
en Informática

Metodología de desarrollo para aplicaciones con enfoque SOA (Service Oriented Architecture)

Andrea Delgado Cavaliere

Supervisor de maestría y orientador de tesis:
Dr. Raúl Ruggia

Montevideo, Uruguay
2007

Metodología de desarrollo para aplicaciones con enfoque SOA (Service Oriented Architecture)
Delgado Cavaliere, Andrea
ISSN 0797-6410
Tesis de Maestría en Informática
Reporte Técnico RT 07-21
PEDECIBA
Instituto de Computación – Facultad de Ingeniería
Universidad de la República.
Montevideo, Uruguay, 2007

PEDECIBA Informática
Instituto de Computación (InCo)
Facultad de Ingeniería
Universidad de la República

Tesis de Maestría en Informática

Metodología de desarrollo para aplicaciones con enfoque SOA (Service Oriented Architecture)

Andrea Delgado Cavaliere

Supervisor de maestría y orientador de tesis: Dr. Raúl Ruggia

Montevideo, Uruguay
2007

ACTA DE TESIS DE MAESTRIA EN INFORMATICA

TITULO: Metodología de desarrollo para aplicaciones con enfoque SOA

AUTOR: Andrea Verónica Delgado Cavaliere

SUPERVISOR y ORIENTADOR de Tesis: Dr. Raúl Ruggia

INTEGRACION DEL TRIBUNAL:

- Dr. Francisco Ruiz González (Revisor)
- Dra. Nora Szasz
- Dr. Alberto Pardo

FALLO DEL TRIBUNAL: Aprobado con calificación de Excelente

FECHA: 20 de diciembre de 2007

"El hombre no es más que lo que hace de sí mismo."

Jean Paul Sartre

RESUMEN

El área de Tecnología Informática (TI) en las Organizaciones actuales se puede caracterizar por tener diversidad de sistemas que tienen entre sí dependencias complejas, que han ido creciendo en forma separada y heterogénea a lo largo de los años. Un desafío que se plantea es poder integrarlos para reaccionar ágilmente a los cambios en los requerimientos del negocio, principalmente en dos aspectos: los procesos de la Organización y las tecnologías disponibles.

Service Oriented Architecture (SOA) es un estilo de Arquitectura de Software basado en la definición de servicios reutilizables, con interfaces públicas bien definidas, donde los proveedores y consumidores de servicios interactúan en forma desacoplada para realizar los procesos de negocio. Los servicios representan grupos lógicos de operaciones relacionadas con algún concepto del negocio, y los procesos del negocio se realizan mediante secuencias definidas de invocaciones a servicios, en orquestación o coreografías de servicios. La definición y disponibilidad de estos servicios para toda la Organización es la base del enfoque SOA.

El Grupo de Ingeniería de Software (Gris) del Instituto de Computación tiene un programa de construcción y prueba de modelos de proceso en el marco del cual se han adaptado y probado diversos procesos, metodologías y enfoques de desarrollo de software. El principal proceso con que se cuenta es una adaptación del Rational Unified Process (RUP), para el cual se definió una metodología para el desarrollo de aplicaciones SOA como parte central de este trabajo .

Esta metodología fue integrada al proceso base adaptación del RUP definiendo una extensión al mismo dada por un conjunto núcleo de elementos que se deben incorporar para este tipo de desarrollos. Esta extensión denominada Extensión SOA, incluye la definición entre otros de Disciplinas, Actividades, Entregables y Roles, adecuados para guíar un desarrollo SOA cumpliendo con las características que define este enfoque.

La metodología fue probada en el marco del curso “Proyecto de Ingeniería de Software” para la construcción de una aplicación de Help-Desk para el proyecto Link-all del InCo, y ajustada y mejorada en base a los resultados obtenidos. Como principal evaluación de su utilización se puede concluir que constituye una guía importante para realizar desarrollos SOA, habiendose obtenido un producto que cumple con las funcionalidades y aspectos de calidad definidos por el cliente, así como con las características que plantea el enfoque SOA.

Se realizó una generalización de la metodología SOA propuesta con miras a su aplicación en la industria del software, en el marco del proyecto COMPETISOFT - Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software de Iberoamérica, definiendo un Perfil SOA para extender el proceso de desarrollo de software que integra el modelo de procesos definido, como propuesta base de trabajo para la incorporación del enfoque SOA al modelo de procesos.

PALABRAS CLAVE: Ingeniería de Software, Procesos y Metodologías de desarrollo, Diseño y Arquitectura de Software, Service Oriented Architecture (SOA), Procesos de Negocio.

Agradecimientos

En primer lugar a mi tutor y supervisor de maestría Dr. Raúl Ruggia por el apoyo brindado durante toda la realización de la maestría, la guía académica en el área, las revisiones y correcciones realizadas a este trabajo y el continuo aliento.

A la Comisión Académica de Posgrado de la Facultad de Ingeniería por haberme becado para finalizar esta tesis en su último período de trabajo, fue realmente un incentivo necesario, a mi tutor nuevamente y a la Dra. Regina Motz por las cartas de recomendación escritas para la presentación a la beca, sobre mi trabajo y persona, realmente las agradezco.

Al Instituto de Computación por haberme dado la oportunidad de ingresar a la carrera docente, y el espacio y las capacidades para iniciarme en la investigación y demás tareas asociadas, lo que se ha constituído en parte importante de mi vida. Al PEDECIBA por la oportunidad de realizar la Maestría en Informática.

Al Centro de Ensayos de Software, Grupo de Ingeniería de Software y a la Comisión Sectorial de Investigación Científica por haberme permitido con sus aportes presentar parte de este trabajo en Congresos de Informática e Ingeniería de Software, realmente fueron experiencias muy enriquecedoras. Al Director del Instituto de Computación Dr. Héctor Cancela por las cartas de referencia que me ayudaron a obtener el financiamiento.

A mis compañeras del Grupo de Ingeniería de Software: Beatriz Pérez y Mónica Wodzislawski, por los intercambios que nos hacen crecer, el apoyo continuo, las alegrías y tristezas compartidas, es un gusto compartir el trabajo y más con uds.

A los grupos del curso Proyecto de Ingeniería de Software que contribuyeron a este trabajo: grupo 5 del curso 2005, todo un aprendizaje pero muy divertido, grupos 1 y 2 del curso 2006 por el gran esfuerzo realizado, a los proyectos de grado 2006 por su trabajo, intercambios de ideas y aportes a esta tesis.

A mis compañeros docentes del curso Proyecto de Ingeniería de Software esos años: Raquel Abella, Federico Piedrabuena, Joaquín Goyoaga, Doris Correa y Beatriz Pérez, por el buen trabajo y los aportes. A los compañeros del proyecto Link-All que me apoyaron y aportaron en la aplicación de la metodología SOA en el curso.

Por último pero no por eso menos importante, sino todo lo contrario, a mi familia toda, pero especialmente a mis padres Alicia y Luis, sin cuyas guías y ejemplos de vida no podría haber transitado este camino, a mis herman@s Marcela, Ana Paula, Patricia y Rodrigo que son mi apoyo continuo, mis referencias, mi alegría, a mis cuñad@s Leonardo, Alejandro, Federico y Paula por estar en la familia y ser también mis hermanos, a mis sobrinas Chiara chiquita divina, y Lucía mi ahijada del alma que es el sol de mi vida, los quiero muchisimo a todos !

A mis amigos que son parte invalorable de mi vida, por su apoyo, comprensión y consejos, a todos, pero sobre todo, a Pablin, Beita y Martincho, que son también mis hermanos, por el aguante, la fuerza y siempre estar, los quiero un montón !

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

1.1. Contexto de trabajo

En los últimos años se han experimentado grandes cambios en el área de la computación, tanto en la proliferación de nuevas tecnologías, metodologías y enfoques de desarrollo que han repercutido en las Organizaciones actuales, como a la inversa, cambios en los requerimientos y necesidades a nivel Organizacional han repercutido en la forma de hacer y ejecutar software. La explosión del uso de internet por las Organizaciones, plantea varias ventajas y desafíos para la forma en que éstas realizan su Negocio, y la forma en que informatizan sus procesos e interactúan con otras Organizaciones. Lo que se hace notorio es que una necesidad que antes pudo ser medianamente satisfecha por el área de Tecnología Informática (TI) mediante la aplicación de diversidad de enfoques y tecnologías, actualmente está requiriendo respuestas más integradas.

Una característica que generalmente se ha dado en el área de TI en las Organizaciones es el pensamiento de "silo", desarrollo de aplicaciones como islas de funcionalidad provista por la integración vertical de un conjunto de componentes construidos para ese propósito. Cuando estas aplicaciones deben ser integradas en procesos más complejos de la Organización, se observa que en general lo único que comparten es la base de datos, requiriendo entonces un mayor esfuerzo de implementación para lograr interactuar entre si en otros niveles. En este contexto, las Organizaciones tienen hoy una diversidad de sistemas más bien independientes, que tienen entre sí dependencias complejas, y que han ido creciendo en forma separada y heterogénea a lo largo de los años.

Un desafío que se plantea es poder integrarlos para reaccionar ágilmente a los cambios en los requerimientos del negocio, principalmente en dos aspectos: los procesos de la Organización y las tecnologías disponibles. La definición y disponibilidad de estos servicios para toda la Organización es la base del enfoque Service Oriented Architecture(SOA), que surge entonces como forma de cambiar la visión del diseño de las aplicaciones en las Organizaciones, permitiendo la reutilización de los elementos base que la componen llamados servicios en forma horizontal por las distintas aplicaciones que ahora se pueden integrar fácilmente utilizando estos servicios.

Una característica que generalmente acompaña el surgimiento de nuevos paradigmas y tecnologías de desarrollo, cuando éstos concitan la atención de la comunidad del software, es que rápidamente proliferan interpretaciones, implementaciones y herramientas, nuevas o adaptaciones de herramientas existentes, que "soportan" o "conforman" a estos paradigmas y tecnologías. El caso de Service Oriented Architecture (SOA) es un ejemplo de esto. Desde su aparición a fines de los años noventa, múltiples autores, investigadores y empresas de desarrollo de software de alcance mundial, han brindado sus definiciones del enfoque, promoviendo su adopción como "la solución" a los problemas mencionados. Sin embargo, las iniciativas tomadas por las empresas en incorporar estos nuevos enfoques de desarrollo o tecnologías sin tener claro como realizar

esta adopción o como mantener el negocio mediante estos enfoques, puede llevar al fracaso de las mismas, como es el ejemplo en los años noventa de las empresas punto com.

Otra característica común es que luego de la cresta de popularidad que se genera por la amplia adopción en las empresas de software de estos nuevos paradigmas y tecnologías de desarrollo, se comienzan a detectar los problemas que trae aparejada la interpretación e implementación distinta dada por las herramientas y formas de adopción recomendadas, y la propuesta se "estanca", hasta que finalmente se resuelven estas discrepancias en base a esfuerzos de acuerdos entre las grandes empresas proveedoras de herramientas de desarrollo acompañadas generalmente de alguna Organización de estandarización internacional, como son en este caso [OMG], [OASIS] y [W3C], con el apoyo de las investigaciones académicas en el área y prácticas realizadas en la industria de software. Difícilmente se pueda introducir y mantener un nuevo enfoque de desarrollo de software en una Organización, basado únicamente en herramientas y tutoriales de las herramientas, es necesario también un enfoque disciplinado para su comprensión y utilización, y estándares que los soporten.

La Ingeniería de Software es una disciplina emergente como se menciona en Guide to the Software Engineering Body of Knowledge [SWEB04] "es especialmente cierto al compararla con otras disciplinas de la Ingeniería mucho más establecidas". El término Ingeniería de Software fue utilizado por primera vez en una conferencia realizada en Alemania en el año 1968 [INSW68]. La IEEE en [IEEE90] define la Ingeniería de Software como: "1) la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; esto es, la aplicación de la ingeniería al software; 2) el estudio de los enfoques como en 1)". Se cuenta actualmente con un cuerpo de conocimiento definido, con referencias importantes en las distintas áreas que lo componen, establecido en [SWEB04] que aporta a la madurez del área. Pero como se indica en dicha guía, "la Ingeniería de Software continúa siendo infundida con nuevas tecnologías y nuevas prácticas. La aceptación de nuevas técnicas crece y técnicas más viejas son descartadas" lo que determina que el área esté en continua evolución.

Por lo que, desde la disciplina de Ingeniería de Software, una posición que puede tomar frente a los nuevos paradigmas y tecnologías que emergen, está dada por la definición presentada: estudiar su aplicación al desarrollo, operación y mantenimiento de software en forma sistemática, disciplinada y cuantificable. En el caso del enfoque de desarrollo SOA, resulta importante contar con una metodología de desarrollo que permita la adopción del enfoque SOA por parte de las distintas Organizaciones, en forma disciplinada, basada en la conceptualización y estandarización de las prácticas asociadas con el enfoque y no en una herramienta o interpretación en particular, maximizando el valor de la comprensión y aplicación del paradigma al desarrollo de software.

En el Grupo de Ingeniería de Software (Gris) del Instituto de Computación de la Facultad de Ingeniería de la Universidad de la República [GRIS06] una línea principal del trabajo de investigación que se realiza, está dada por el programa de construcción y prueba de modelos de proceso. El programa inició en el año 2000 y tiene como objetivo central el desarrollo y prueba de modelos de proceso que puedan resultar adecuados para su transferencia a la industria, identificando buenas prácticas que puedan ser incorporadas por las distintas empresas. Para esto se utiliza como banco de pruebas de los procesos el curso "Proyecto de Ingeniería de Software" que se dicta en cuarto año de la carrera de Ingeniería en Computación, cuyas características son consistentes con el objetivo del programa. [ADBP06]

Entre los objetivos principales del curso se plantea que los estudiantes contrasten la teoría con su aplicación práctica en proyectos sometidos a restricciones análogas a las de la industria, para lo que, entre otras características, se cuenta con clientes externos al curso. Estos clientes son en general grupos de investigación, proyectos o áreas de la misma facultad, o empresas del medio, quienes plantean el desarrollo de sistemas de mediano porte que en general presentan desafíos tecnológicos importantes. Los proyectos se realizan por equipos de desarrollo compuestos por grupos de estudiantes de entre ocho y doce personas, con una duración fija en semanas y horas por semana a realizar, siguiendo un proceso de desarrollo con Fases e hitos establecidos y con Disciplinas, Actividades, Entregables y Roles definidos. [ADBP06]

El programa cuenta con tres hitos principales: obtener un proceso definido, obtener un proceso validado y obtener un proceso ajustado y mejorado. Para esto se definen los procesos en forma previa al curso "Proyecto de Ingeniería de Software", realizando la prueba de estos procesos en dicho curso por parte de grupos de estudiantes que desarrollan las aplicaciones planteadas por los distintos clientes. Luego se evalúa la aplicación de los procesos en los proyectos y los resultados obtenidos, identificando mejoras a los procesos definidos, obteniendo procesos ajustados y mejorados que pueden ser puestos en práctica al siguiente año. [ADBP06]

Desde su puesta en marcha y hasta la fecha se han estudiado y adaptado varios modelos de proceso de las distintas corrientes existentes, tanto procesos "pesados" como "ágiles", siendo el representante más importante de los primeros el Rational Unified Process (RUP) [RUP] y de los segundos el eXtreme Programming (XP) [BECK99]. De estos dos procesos se han realizado adaptaciones que se han puesto a prueba en distintas ediciones del curso, definiéndose como proceso base del programa la adaptación realizada del RUP cuya concepción surge a partir de un proyecto de grado realizado en el año 2000 [ADBP00] que es sobre la cual se agregan otros enfoques y metodologías de desarrollo. Una descripción completa de la concepción y evolución del programa puede verse en [ADBP06].

En el contexto presentado surge entonces la oportunidad para la realización del trabajo de tesis de maestría que se presenta en este informe: la definición de una metodología de desarrollo para aplicaciones con enfoque Service Oriented Architecture (SOA).

1.2. Aportes

El primer aporte en la realización de este trabajo consiste en investigar y conceptualizar el enfoque de desarrollo de software Service Oriented Architecture (SOA), estilo de Arquitectura de Software a partir del cual es posible construir aplicaciones distribuidas que cumplen con determinadas características establecidas en el mismo. SOA es un paso más en la evolución de los paradigmas de desarrollo de software para proveer soluciones a las necesidades de las Organizaciones actuales, en aspectos como interoperabilidad de aplicaciones desarrolladas con distintas tecnologías, reutilización de conocimiento, diseño y código existente en la misma Organización o fuera de ésta, y enfoque de desarrollo basado en el Negocio, esto es, en los procesos del Negocio que realiza la Organización para alcanzar sus objetivos estrátegicos, ayudando a cerrar la brecha que en general existe entre las áreas del Negocio y de Tecnología Informática.

En este sentido, un enfoque con el cual los conceptos en SOA están muy relacionados es el de Business Process Management (BPM), que trata con la gestión de los procesos del Negocio en una Organización, incluyendo definición, monitoreo de su desempeño y mejora, para cumplir con los objetivos del negocio definidos. Con los objetivos de lograr interoperabilidad y portabilidad de aplicaciones desarrolladas con distintas tecnologías, y reutilización de conocimiento, diseño y código existente, surge el enfoque de desarrollo basado en modelos o Model Driven Architecture (MDA), el cual se relaciona también con el enfoque SOA en la obtención de aplicaciones con las características definidas.

Los conceptos por detrás del enfoque SOA se apoyan en el conocimiento y evolución de las disciplinas de Diseño y Arquitectura de Software, en las que se establecen varios de los principios básicos para el desarrollo de software. Entonces, para entender cómo y porqué se define y utiliza el enfoque SOA, se deben tener en cuenta otros aspectos relacionados con el mismo. Para esto, se presenta el estado actual del conocimiento en las disciplinas de Diseño y Arquitectura de Software con una visión amplia de sus elementos, para ubicar el contexto de SOA. Se presenta el estado actual del conocimiento en el enfoque SOA que constituye uno de los aspectos centrales de este trabajo, y su relación con los enfoques BPM y MDA mencionados, de los que se provee una visión más reducida.

En segundo lugar, el principal aporte de la realización de este trabajo, definir una metodología para el desarrollo de aplicaciones con enfoque SOA teniendo en cuenta las características

que presenta el enfoque y los objetivos que persigue, estableciendo un enfoque disciplinado para su utilización, que sirva de guía para la obtención de dichos objetivos. Este objetivo es el más importante de este trabajo de tesis, e incluye, entre otros elementos, la definición de actividades y productos a generar durante el proceso de desarrollo, orientados a la obtención de un producto con las características establecidas en el enfoque SOA. Como base para esto, se presenta el estado actual del conocimiento en procesos y metodologías de desarrollo de Software, incluyendo conceptos, procesos de desarrollo y de mejora de procesos de referencia en el área. Adicionalmente se debía validar la metodología propuesta mediante su aplicación en un proyecto de desarrollo de software con enfoque SOA.

Se define en el marco del programa de construcción y prueba de modelos de procesos, una metodología para el desarrollo de aplicaciones con enfoque SOA como una extensión del proceso base adaptación del Rational Unified Process (RUP) [RUP] con que cuenta el Grupo de Ingeniería de Software (Gris) [GRIS06], denominando dicho proceso como Extensión SOA. Este proceso es probado en el curso "Proyecto de Ingeniería de Software" durante el segundo semestre del año 2005, para el desarrollo de una aplicación de Help-Desk que se debía integrar en la plataforma SOA desarrollada en el marco del Proyecto Link-all [LKAL05]. Para la aplicación de la metodología en dicho curso se define un conjunto núcleo de elementos primarios a incorporar a la Extensión SOA, y se realiza un Sitio Web para hacer disponible la información asociada [WTAD05].

En tercer lugar, la metodología debía ser ajustada y mejorada en base a la prueba realizada, identificando oportunidades de aplicación de la misma en la industria de software o como parte de otros procesos de desarrollo definidos para su aplicación en la industria de software. En base a los resultados obtenidos en la prueba de la metodología SOA realizada en el curso "Proyecto de Ingeniería de Software", la metodología es ajustada y mejorada, extendiendola mediante el agregado de otros elementos que complementan los elementos núcleo definidos inicialmente. En el marco del proyecto COMPETISOFT - Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software de Iberoamérica [COMP06], se define el perfil SOA integrado por los elementos núcleo definidos en la metodología, para su incorporación al modelo de procesos provisto como parte del Marco Metodológico del proyecto, el cual se remite para su consideración a los integrantes del mismo.

Al momento de la definición de la metodología SOA cabe destacar que no existían metodologías de desarrollo completas para el enfoque SOA, sino más bien guías y procedimientos aislados que no brindaban una visión completa del desarrollo de software con este enfoque, como [ENDR04], [KRAF05] y [ERLT05]. En el transcurso de realización de este trabajo, fueron apareciendo otras propuestas, que sirven por un lado, para confirmar aspectos definidos en la metodología SOA propuesta que constituyen "buenas prácticas" identificadas para este tipo de desarrollos, y por otro lado, para revisar aspectos no contemplados o resueltos con una visión distinta. Sin embargo, no era un objetivo de este trabajo realizar una comparación con otras propuestas que hubieran podido surgir durante su realización, por lo que únicamente se describe brevemente una de las más importantes asociada al RUP.

Otro aporte a mencionar es la generación, publicación y presentación de artículos en distintos Congresos del área durante los años 2006 y 2007. Dos de estos artículos corresponden a la línea principal del trabajo realizado como parte de esta tesis de maestría:

- "Desarrollo de aplicaciones con enfoque SOA (Service Oriented Architecture)", Delgado A., González L., Piedrabuena F., en V Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento (JIISIC'06), Actas, ISBN 970-94770-0-5, pp. 237-244, Puebla, México, Febrero de 2006.
- "Metodología para desarrollo de aplicaciones con enfoque SOA (Service Oriented Architecture)", Delgado A., en XXXII Conferencia Latinoamericana de Informática (CLEI'06), Sesión 4, Artículo Nro. 265, Libro de resúmenes pp. 99, Santiago de Chile, Chile, Agosto de 2006.

Otros dos artículos corresponden a la exploración de aspectos menos centrales para el trabajo de tesis, pero relacionados con la investigación que se realiza en el área, que se presentan a continuación:

- "Extensión MDA (Model Driven Architecture) para proceso basado en RUP (Rational Unified Process)", Delgado A., Carballal N., Rapetti C., en VI Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento (JIISIC'07), Actas, ISBN 978-9972-2885-1-7, pp. 247-255, Lima, Perú, Febrero de 2007.
- "Evaluación de Arquitecturas de Software con ATAM (Architecture Tradeoff Analysis Method): un caso de estudio", Delgado A., Castro A., Germán M., en VI Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento (JIISIC'07), Actas, ISBN 978-9972-2885-1-7, pp. 151-159, Lima, Perú, Febrero de 2007.

Otro artículo corresponde a un trabajo recientemente aceptado pero que aún no ha sido publicado ni presentado, ya que el Congreso se realizará en setiembre de este año, como se describe a continuación:

- "Desarrollo de Software con enfoque en el Negocio", Delgado A., en I Taller sobre Procesos de Negocio e Ingeniería de Software (PNIS'07), a realizarse en Zaragoza, España, 11 de setiembre de 2007, Workshop realizado en el marco de las XII Jornadas de Ingeniería del Software y Bases de Datos (JISBD'07) en el II Congreso Español de Informática (CEDI 2007).

1.3. Organización del documento

El presente documento, que corresponde al informe del trabajo de tesis de maestría realizado, se encuentra organizado como se describe a continuación:

En el Capítulo 2 se presenta el estado actual del conocimiento en el área de procesos y metodologías de desarrollo, introduciendo conceptos asociados al ciclo de vida del software, procesos del ciclo de vida del software, procesos de desarrollo de software y procesos de evaluación y mejora de procesos, que establecen las bases para la definición del proceso en el que se enmarca la metodología SOA definida.

En el Capítulo 3 se presenta el estado actual del conocimiento en las Disciplinas de Diseño y Arquitectura de Software, donde se describen los principales conceptos y definiciones asociados así como la evolución del área, y procesos definidos para su realización. Se presenta la importancia que tiene la Arquitectura de Software en el desarrollo, y enfoques para su documentación y evaluación.

En el Capítulo 4 se presenta el estado actual del conocimiento en el enfoque Service Oriented Architecture (SOA), definiendo el contexto, conceptos y elementos asociados al mismo. Se presentan también los principales enfoques de desarrollo que se tuvieron en cuenta para la definición de la metodología SOA propuesta. Se presenta el enfoque Business Process Management (BPM) y se muestra su relación con el enfoque SOA. Se presenta el enfoque Model Driven Architecture (MDA) y su relación con el enfoque SOA.

En el Capítulo 5 se presenta la metodología SOA propuesta, incluyendo el proceso base adaptación del RUP que se extiende para desarrollo con enfoque SOA en la Extensión SOA definida. Se describen las Disciplinas, Actividades, Entregables y Roles definidos por la metodología SOA propuesta, incluyendo el modelado gráfico de los elementos por Actividad definida, y la relación de las Disciplinas con las Disciplinas en el proceso base.

En el Capítulo 6 se presenta la aplicación de la metodología SOA realizada mediante la prueba de la Extensión SOA en el curso "Proyecto de Ingeniería de Software" para el desarrollo de la

aplicación de Help-Desk en el marco del proyecto Link-all [LKAL05]. Se muestra el desarrollo y seguimiento realizado de dicha aplicación, los resultados obtenidos y las conclusiones de la prueba.

En el Capítulo 7 se presentan los ajustes, mejoras y generalización realizados sobre la metodología SOA propuesta en base a los resultados de la aplicación de la metodología SOA y al agregado de elementos ya previsto al momento de la definición. Se describe el contexto de la generalización realizada de la metodología SOA propuesta como perfil SOA para el proyecto COMPETISOFT [COMP06] y sus principales aportes. Se presenta además una breve descripción de una propuesta de reciente surgimiento para desarrollo SOA con RUP, realizando una comparación con la propuesta de este trabajo.

En el Capítulo 8 se presentan las conclusiones y líneas de trabajo a futuro que quedan abiertas a partir de lo realizado en esta tesis de maestría.

En el Capítulo 9 - Anexo A se presenta el Sitio Web realizado para la aplicación de la metodología SOA propuesta en el curso "Proyecto de Ingeniería de Software", como entorno para la prueba realizada.

En el Capítulo 10 - Anexo B se incluyen las plantillas definidas para los entregables que define la metodología SOA propuesta, como parte integral de la misma para la obtención de los productos asociados.

En el Capítulo 11 - Anexo C se presentan los principales aspectos de la generalización realizada para COMPETISOFT nombrada Perfil SOA (PSOA).

Finalmente se presentan las referencias Bibliográficas que fueron utilizadas para la realización de este trabajo.

Capítulo 2

Procesos y metodologías de desarrollo

2.1. Introducción

En este capítulo se presenta el estado actual del conocimiento en procesos y metodologías de desarrollo, relacionadas con el trabajo de tesis realizado, por lo que, no se cubre en forma exhaustiva la vasta variedad de enfoques existentes en el área sino únicamente aquellos que de una forma u otra están relacionados con este trabajo. En particular se provee el contexto para los procesos de desarrollo que han sido adaptados en el marco del programa de definición y prueba de procesos del Gris [GRIS06], que fueron evaluados para la definición de la metodología SOA propuesta, así como modelos de mejora de procesos cuyos enfoques se encuentran incorporados en dichas adaptaciones y/o que por su importancia constituyen un aporte en el área.

Se presenta en primer lugar en la sección 2.2 conceptos asociados a los modelos y procesos del ciclo de vida del software, referencia fundamental para la definición de procesos de desarrollo de sofware. En la sección 2.3 se presenta una selección de procesos de desarrollo como representantes más importantes de los distintos enfoques existentes, referidos generalmente como "pesados" y "livianos" o ágiles. En la sección 2.4 se presentan modelos de mejora de procesos que proveen guías para la evaluación y mejora de los procesos en las Organizaciones.

2.2. Ciclo de Vida del Software

Según [PFLE02] "El proceso de desarrollo de software suele denominarse ciclo de vida del software, porque describe la vida de un producto de software desde su concepción hasta su implementación, entrega, utilización y mantenimiento. ... un proceso de desarrollo de software debe describirse de manera flexible que permita que las personas diseñen y construyan el software utilizando las herramientas y técnicas preferidas; un modelo de proceso puede exigir que el diseño se realice antes de la codificación, pero debe permitir la utilización de diferentes técnicas de diseño. ... Todo modelo de proceso de desarrollo de software incluye los requerimientos del sistema como entrada y un producto entregado como salida."

En [SWEB04] se definen y establece la diferencia entre modelos del ciclo de vida del software y procesos del ciclo de vida del software como sigue: "Los modelos del ciclo de vida del software sirven como definición de alto nivel de las fases que ocurren durante el desarrollo. No tienen como objetivo proveer definiciones detalladas sino resaltar las actividades claves y sus interdependencias.", ordenando la ejecución de dichas fases en el tiempo, por otro lado, "Las

definiciones de procesos del ciclo de vida del software tienden a ser más detalladas que los modelos del ciclo de vida del software. Sin embargo, los procesos del ciclo de vida del software no intentan ordenar sus procesos en el tiempo. Esto significa que, en principio, los procesos del ciclo de vida del software pueden ser organizados para acomodarse a cualquiera de los modelos del ciclo de vida del software." A continuación se presentan modelos del ciclo de vida del software en 2.2.1 y procesos del ciclo de vida del software en 2.2.2.

2.2.1. Modelos del Ciclo de Vida del Software

Los modelos del ciclo de vida del software como se mencionó, definen el orden y relación entre las distintas etapas del desarrollo de software. Estas etapas en general se definen como análisis de requerimientos, diseño de la solución, implementación de los programas, pruebas de los programas, entrega del sistema y mantenimiento. En [PFLE02] se describen como modelos del ciclo de vida del software, el modelo en cascada, el modelo V, el modelo de prototipos, el modelo de especificación operacional, el modelo de transformación, el modelo en fases con desarrollo iterativo y desarrollo incremental y el modelo en espiral. Algunos de estos modelos se mencionan como ejemplo de modelos del ciclo de vida también en [SWEB04]. A continuación se describen brevemente los modelos en cascada y de fases iterativo e incremental, conceptos que se utilizan en el proceso base de la metodología SOA propuesta.

2.2.1.1. Modelo en cascada y cascada con prototipos

El modelo en cascada según [PFLE02] es uno de los primeros modelos propuestos, donde las etapas se representan en cascada desde una etapa hacia la siguiente. Cada etapa de desarrollo debe completarse antes de dar comienzo a la siguiente, por ejemplo, cuando todos los requerimientos han sido identificados, analizados por integridad y consistencia, y documentados, el equipo de desarrollo puede pasar a la siguiente fase de diseño. Presenta una visión clara de la sucesión de etapas en el desarrollo y sugiere la secuencia de eventos a realizar. Asociados con cada actividad del proceso se encuentran los hitos y las entregas para realizar seguimiento del proyecto. El mayor problema con este enfoque es que no refleja la manera en que realmente se realiza el desarrollo, donde existe un alto grado de repetición de actividades de una y otra etapa, por ejemplo cuando la solución debe actualizarse para reflejar cambios. Para contrarrestar estos aspectos se incluye el prototipado en las etapas de análisis y diseño, dando lugar al modelo en cascada con prototipos. En la figura 2.1 se muestran el modelo en cascada y el modelo en cascada con prototipos.

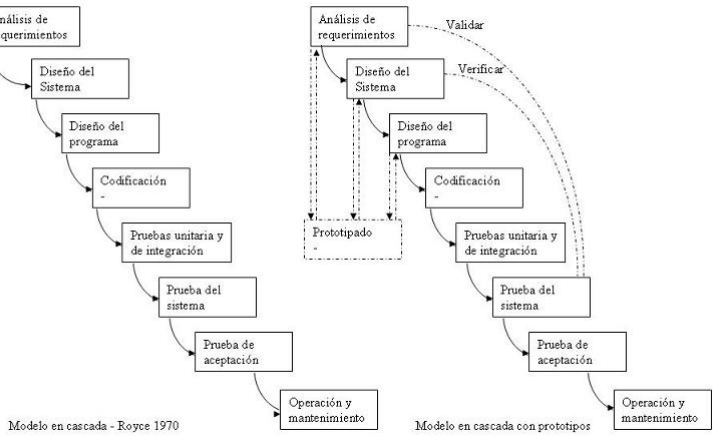


Figura 2.1: Modelo del ciclo de vida en cascada y cascada con prototipos de [PFLE02]

2.2.1.2. Modelo en fases con desarrollo iterativo e incremental

Como se indica en [PFLE02], para ayudar a reducir el ciclo de vida de los proyectos, que en los primeros años del desarrollo de software podían ser años entre la especificación de requerimientos y el momento en que se entregaba el software, se desarrollaron nuevos modelos de proceso. Una de las maneras para lograr la reducción del ciclo de vida, es el desarrollo por fases, donde el sistema se diseña para ser liberado en versiones sucesivas, en las que los usuarios disponen de determinadas funcionalidades en operación mientras en desarrollo se sigue construyendo la versión siguiente que agregará funcionalidades nuevas o mejorará funcionalidades existentes en la versión anterior o ambas.

Dos enfoques populares para organizar el desarrollo de esta manera son el desarrollo incremental y el desarrollo iterativo. En el desarrollo incremental el sistema es particionado en subsistemas de acuerdo con su funcionalidad, y las versiones se desarrollan comenzando con un subsistema funcional pequeño y agregando funcionalidades con cada nueva versión. En el desarrollo iterativo se entrega en la primer versión un sistema completo en cuanto a cubrimiento de funcionalidades en estado básico, y luego en cada nueva versión se cambia y/o mejora la funcionalidad de cada subsistema basado en el uso del sistema. En general estos enfoques se utilizan combinados resultando en un desarrollo iterativo incremental donde una nueva versión puede incluir funcionalidades nuevas, pero la funcionalidad que ya existía en la versión anterior también puede haber sido mejorada. En la figura 2.2 se muestra el modelo por fases, el desarrollo incremental y el desarrollo iterativo.

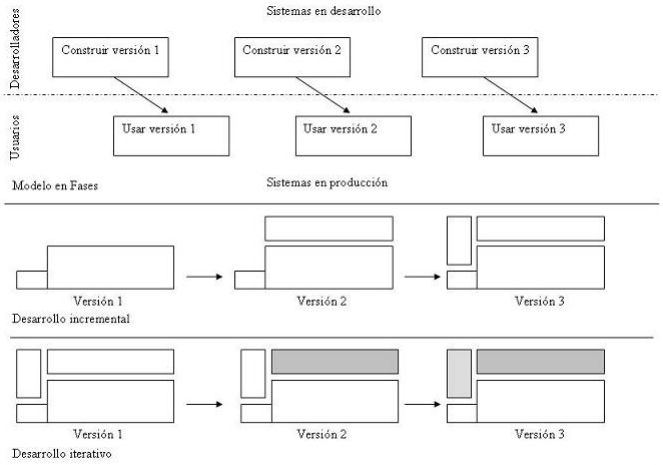


Figura 2.2: Modelo del ciclo de vida en Fases y desarrollo incremental e iterativo de [PFLE02]

2.2.2. Procesos del Ciclo de Vida del Software

Según [SWEB04] "los procesos del ciclo de vida del software se definen por una cantidad de razones, incluyendo incrementar la calidad del producto, facilitar la comprensión humana y comunicación, soportar la mejora de procesos y la gestión de procesos, proveer guías automatizadas al procesos y soporte de ejecución automatizada. Los tipos de definición de procesos requeridos dependerán, por lo menos parcialmente, en la razón de la definición. Variables importantes a considerar incluirán la naturaleza del trabajo (por ejemplo, mantenimiento o desarrollo), el dominio de aplicación, el modelo del ciclo de vida y la madurez de la Organización." La referencia más importante para la definición de procesos del ciclo de vida del Sofware es la norma ISO/IEC 12207 - Procesos del Ciclo de Vida del Software, que se presenta en 2.2.2.1.

2.2.2.1. Norma ISO/IEC 12207 - Procesos del Ciclo de Vida del Software

La Norma ISO/IEC 12207 Procesos del Ciclo de vida del Software [ISO95] es un estándar internacional que establece un marco común para los procesos de ciclo de vida de software, con terminología bien definida, que puede ser referenciado por la industria de software. Aplica a la adquisición de sistemas y productos y servicios de software, a la provisión, desarrollo, operación y mantenimiento de productos de software, realizados interna o externamente a una Organización. Contiene un conjunto de procesos, actividades, y tareas diseñadas para ser adaptadas por los proyectos de software.

Describe la arquitectura del ciclo de vida de los procesos pero no especifica los detalles de como implementar o realizar las actividades y tareas incluidas en los procesos, ni tampoco prescribe nombre, formato o contenido explícito de la documentación a ser producida, ni tampoco un modelo de ciclo de vida del software específico ni metodología de desarrollo de software a utilizar. Agrupa las actividades que deben ser realizadas durante el ciclo de vida del software en cinco procesos primarios, ocho procesos de soporte y cuatro procesos organizacionales. Cada proceso del ciclo de vida es dividido en un conjunto de actividades, y cada actividad es dividida a su vez en un conjunto de tareas.

- Los procesos primarios incluyen las actividades principales del ciclo de vida del software y son los de: Adquisición, Provisión, Desarrollo, Operación y Mantenimiento del software.
- Los procesos de soporte contribuyen al éxito y calidad del proyecto de software y son los de: Documentación, Gestión de Configuración, Aseguramiento de Calidad, Verificación, Validación, Revisión conjunta, Auditoría y Resolución de problemas.
- Los procesos organizacionales sirven a la Organización para establecer e implementar la estructura base a partir de los procesos del ciclo de vida asociados y mejorar en forma continua esta estructura y los procesos, y son los de: Gestión, Infraestructura, Mejora y Formación (o entrenamiento).

Brinda también un proceso de adaptación en el anexo A que se debe seguir para adaptar el estándar en forma específica para cada Organización, y en el anexo B se provee una guía de elementos a tener en cuenta para realizar dicha adaptación. En la figura 2.3 se muestra la estructura de procesos definida en el estándar.

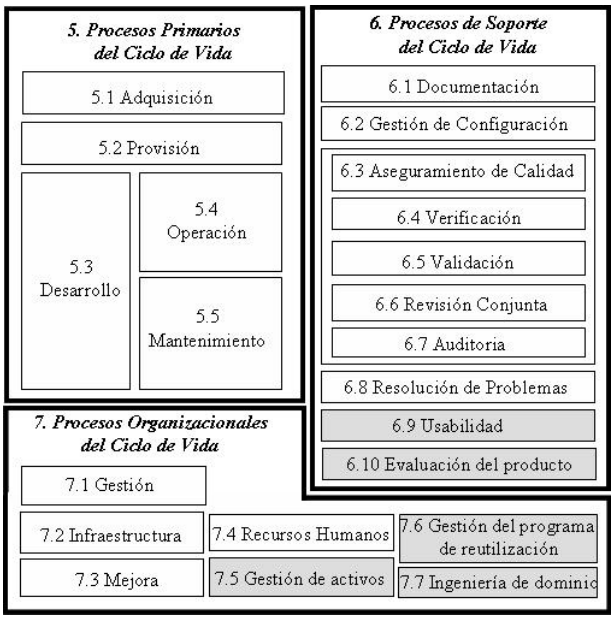


Figura 2.3: Estructura de procesos del Ciclo de Vida definida en ISO/IEC 12207 de [ISO95]

Las enmiendas 2002 y 2004 realizan cambios sobre la guía básica agregando diversos Anexos para extender la norma del año 1995. Se agregan los nuevos procesos de Usabilidad, Evaluación de producto, Gestión de activos, Gestión del programa de reutilización e Ingeniería del dominio, y se renombra el proceso de formación por Recursos Humanos. En el Anexo G se incluye la información asociada con cada nuevo proceso agregado, siguiendo la forma establecida en la norma base y numerándolos en forma consecutiva a los ya existentes. En la figura 2.3 se muestran los procesos agregados en los recuadros sombreados en gris. En el Anexo F se incluye un modelo de procesos de referencia, que define la totalidad de los procesos de la norma, según los requerimientos de evaluación de procesos establecidos en la norma ISO/IEC 15504-2, por lo que, para cada proceso se brinda un propósito y salidas esperadas, asegurando la consistencia entre ambos estándares. Al momento de escribir este trabajo el estándar se encuentra en una etapa de revisión mayor por parte de ISO para la liberación de una nueva versión.

2.2.3. Notaciones para modelado de procesos de Software

Como se indica en [SWEB04] existe una variedad de notaciones para definir procesos, donde una diferencia clave entre ellos es el tipo de información que se define, captura y utiliza, por ejemplo en la norma ISO/IEC 15504 en términos de procesos y sus salidas, en la norma ISO/IEC 12207 como una lista de procesos que se descomponen en las actividades y tareas que los conforman, en lenguaje natural, y otras como máquinas de estado y redes de petri. En el año 2002 el OMG [OMG] publicó el estándar Software Process Engineering Metamodel (SPEM) para modelado de procesos, con la intención de armonizar las notaciones existentes, el cual se presenta en 2.2.3.1.

2.2.3.1. Software Process Engineering Metamodel (SPEM) de OMG

Según [SPEM05] el metamodelo es utilizado para describir un proceso concreto de desarrollo de software o una familia de procesos relacionados de desarrollo de software. Como se indica en [SPEM05] "SPEM es un metamodelo para definir procesos y sus componentes. Una herramienta basada en SPEM debería ser una herramienta para definir y adaptar procesos. ... Esta especificación está limitada a la definición de un conjunto mínimo de elementos de modelado de procesos necesario para describir cualquier proceso de desarrollo de software ... SPEM se define como un metamodelo MOF y como un perfil UML, para permitir a los modeladores SPEM la utilización de UML como notación concreta, y para permitir el intercambio de modelos definidos con SPEM en formato XMI, como metadata que conforma un metamodelo MOF."

El Object Management Group (OMG) [OMG] utiliza una arquitectura de cuatro capas de modelado sobre la cual está basada la definición de sus estándares, y el concepto de metamodelado definido como: el uso de modelos (en una capa de abstracción) para describir modelos (en una capa inferior siguiente). Estas capas se notan M0, M1, M2 y M3; donde M0 representa instancias, M1 representa modelos, M2 representa metamodelos y M3 representa metametamodelos. En este contexto en la capa M0 se ubican los procesos reales en producción, en la capa M1 se encuentra la definición de dicho proceso, por ejemplo el RUP, en la capa M2 se ubica el metamodelo que define SPEM, que sirve como template para los procesos del nivel M1. En esta capa se ubica también la definición de UML.

Entre los elementos más importantes que se definen en SPEM se encuentran los relativos a la estructura de los procesos, componentes de los procesos y ciclo de vida de los procesos. En cuanto a la estructura, se definen elementos como *producto de trabajo* y *tipo de producto de trabajo*, que proveen una categorización para los productos obtenidos como ser "documento de texto" o "modelo UML", *definición de trabajo* que especifica el trabajo realizado en el proceso, con especializaciones como *actividad* y *paso*, y *realizador del proceso*, cuya especialización *rol del proceso* se relaciona con *productos de trabajo* y *definiciones de trabajo*.

Entre los componentes de los procesos se pueden mencionar los elementos: *proceso* que define el proceso completo, y *disciplina* que agrupa *actividades* de un proceso de acuerdo a un tema

común. El ciclo de vida de los procesos se modela con elementos principales como *fase*, especialización de *definición de trabajo* con un criterio de entrada y un criterio de salida, que se realizan en forma secuencial en un proceso, *ciclo de vida* como una secuencia definida de fases para alcanzar un objetivo específico, e *iteración* como *definición de trabajo* con un hito menor. En la figura 2.4 se muestra el diagrama de clases de la estructura de los procesos definida.

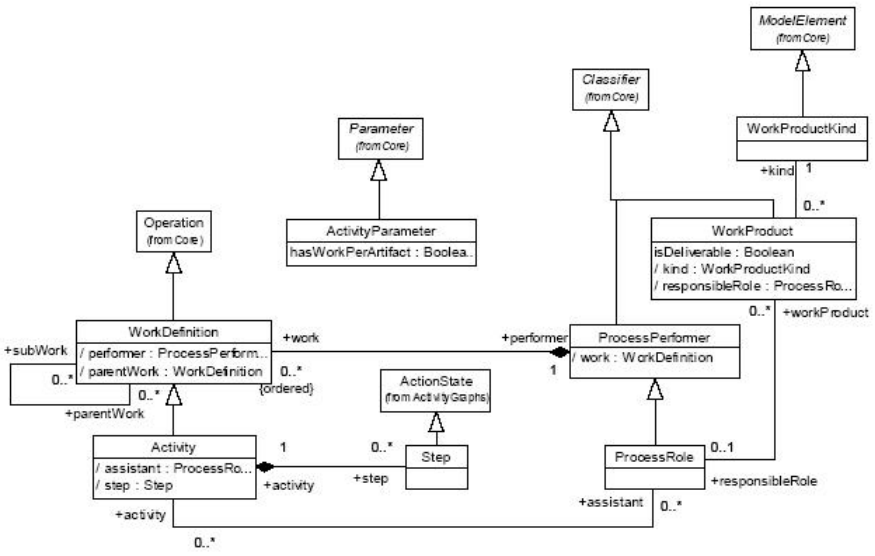


Figura 2.4: Diagrama de clases de la estructura de los procesos en SPEM de [SPEM05]

Para la mayoría de los elementos definidos se proveen iconos sugeridos alternativos a los existentes en UML, que se sugiere reemplazar por los provistos; estos iconos se proveen en OMG en varios formatos y pueden consultarse en [SPEM05]. Sin embargo, para la metodología SOA definida se utilizó la notación del RUP, que tiene elementos en común con SPEM en base a los esfuerzos de unificación realizados.

2.3. Procesos de desarrollo de Software

Los procesos de desarrollo de software existentes en el área de la Ingeniería de Sofware, se pueden clasificar actualmente como procesos "pesados" y "livianos" o ágiles. Según [XPIN] una metodología "pesada" se puede identificar por tener muchas reglas, prácticas y documentos, requiriendo disciplina y tiempo para seguirla correctamente mientras una metodología "liviana" o ágil tiene solo unas pocas reglas y prácticas que se plantea son más fáciles de seguir. Como representante más importante de los procesos "pesados" se destaca el Rational Unified Process (RUP), y de los procesos "livianos" o ágiles se destaca el eXtreme Programing (XP).

De estos procesos se cuenta con adaptaciones en el marco del programa de definición y prueba de procesos del Gris [GRIS06], que fueron evaluados para la definición de la metodología SOA propuesta. En 2.3.1 se presenta el Rational Unified Process (RUP) seleccionado como proceso base de la metodología SOA definida y en 2.3.2 se describen brevemente los aspectos principales de eXtreme Progaming (XP).

2.3.1. Rational Unified Process (RUP)

Según [RUP] "un proceso es un conjunto de pasos parcialmente ordenados que intentan alcanzar un objetivo. En la Ingeniería de Software, ese objetivo es la construcción de un producto de

software o la mejora de uno existente. En la Ingeniería de procesos, el objetivo es desarrollar o mejorar un proceso." El Rational Unified Process (RUP) en su versión completa está catalogado como proceso "pesado", si bien el proceso general es instanciable a las necesidades de cada Organización para crear sus propios procesos basados en RUP que pueden ser menos "pesados" que el original.

Es un proceso adaptable tanto a las distintas Organizaciones como a los diferentes dominios y tamaños de los proyectos en desarrollo, y desde su primera versión está en constante evolución. Provee un enfoque disciplinado para la asignación de tareas y responsabilidades en la Organización de desarrollo, así como para la ejecución y seguimiento de los proyectos en la Organización. En esta sección se presenta el Rational Unified Process (RUP) describiendo en 2.3.1.1 las características principales que lo identifican, luego en 2.3.1.2 se presentan las dos dimensiones de modelado y elementos que lo definen.

2.3.1.1. Principales características

El Rational Unified Process (RUP) presenta cuatro características principales: es iterativo, incremental, guiado por Casos de Uso y centrado en la Arquitectura. Sobre estas características principales está construido el proceso y la metodología que plantea el RUP para el desarrollo de Software. Se basa fuertemente en la obtención de entregables o artefactos, que pueden ser ejecutables, código, y documentación de los varios aspectos del desarrollo, los que se van obteniendo en los hitos que plantea el proceso. Estos hitos corresponden a la definición de Fases e iteraciones en el desarrollo, y se obtienen mediante la realización de Actividades definidas en las distintas Disciplinas especificadas, por parte de los roles establecidos. A continuación se describen las características principales del RUP.

Iterativo e incremental

El desarrollo iterativo e incremental se ha constituido en los últimos años en un aporte importante para asegurar la adecuación del desarrollo a los requerimientos planteados. Este tipo de desarrollo permite la construcción del sistema mediante sucesivas liberaciones que constituyen incrementos en las funcionalidades requeridas, a partir de un conjunto de funcionalidades y prestaciones básicas definidas conjuntamente con el cliente, que se van obteniendo mediante iteraciones en el desarrollo, en las que se realizan las actividades de las disciplinas definidas, para el subconjunto bajo desarrollo en cada iteración. Para cada iteración se define el subconjunto de funcionalidades y prestaciones que integrarán la liberación de la iteración, y para éstas se realizan las actividades correspondientes a las disciplinas definidas.

El desarrollo iterativo incremental ayuda entre otros aspectos a que la comprensión del problema se realice en forma creciente realizando sucesivos refinamientos de los requerimientos planteados, facilitando la inclusión de cambios en los mismos al ir iterando sobre éstos, la identificación y mitigación de riesgos asociados en cada etapa, el control de los productos intermedios liberados en cuanto a calidad y también en cuanto a la obtención de información importante del estado del proyecto en cada momento, propiciando un mayor involucramiento del cliente mediante la validación constante de productos y por lo tanto obtención continua de feedback sobre el desarrollo.

Guiado por Casos de Uso

La metodología que propone el RUP está basada en la especificación de las funcionalidades del sistema como Casos de Uso, los que muestran la interacción entre los actores identificados y el sistema, según los flujos definidos para cada Caso de Uso. Cada Caso de Uso se compone del flujo básico o principal de ejecución y los flujos alternativos que se definen frente a condiciones que no se cumplen en el flujo principal. También se especifican pre y post condiciones para la ejecución del Caso de Uso y los actores involucrados en el mismo.

Los Casos de Uso guían todo el desarrollo ya que a partir de la especificación de los mismos se define y planifica todo el desarrollo, comenzando por la identificación de Casos de Uso relevantes

para definir la Arquitectura de Software, definiendo en el Alcance del proyecto para cada fase e iteración que Casos de Uso integrarán la liberación asociada, así como que Casos de Uso compondrán la versión final a obtener. Para cada iteración los Casos de Uso bajo desarrollo serán analizados, diseñados, implementados y verificados, incrementando funcionalidades sobre la liberación anterior, hasta obtener la versión final del producto.

Centrado en la Arquitectura

Un aspecto clave en la metodología propuesta por el RUP es el enfoque en la definición de la Arquitectura de Software, la cual se realiza en forma temprana en las Fases Inicial y de Elaboración a partir de la identificación de Casos de Uso relevantes a la Arquitectura. Estos Casos de Uso son aquellos que al ejecutar ejercitan aspectos importantes a tener en cuenta en el diseño de la Arquitectura de Software (por ej. acceso a software externo, a Bases de Datos, control de seguridad de acceso al sistema o seguridad en intercambio de datos, entre otros), o que son importantes desde el punto de vista del cliente como funcionalidad básica que da valor al sistema.

El primer subconjunto de funcionalidades y prestaciones a ser construido está constituido por estos Casos de Uso relevantes a la Arquitectura, lo que aporta a la mitigación de riesgos asociados al desarrollo en forma temprana, así como a descubrir aspectos que no han sido del todo resueltos en el diseño planteado o que no se han resuelto de forma aceptable según los requerimientos. Permite también que el resto de las funcionalidades sean agregadas en la fase de Construcción sobre un esqueleto base que viene siendo desarrollado y probado desde etapas tempranas lo que aporta a la estabilidad del diseño e implementación asociada.

La metodología para la definición y documentación de la Arquitectura de Software que plantea el RUP se presenta con mayor detalle en el capítulo 3 de Diseño y Arquitectura de Software.

2.3.1.2. Modelado en dos dimensiones

El RUP se modela mediante dos dimensiones: la dimensión del Tiempo que muestra los aspectos dinámicos del proceso y define Fases e iteraciones para el desarrollo, y la dimensión de las Disciplinas que muestra los aspectos estáticos del proceso y define Disciplinas, actividades, entregables y roles para el desarrollo. En la figura 2.5 se muestran las dos dimensiones de modelado del RUP.

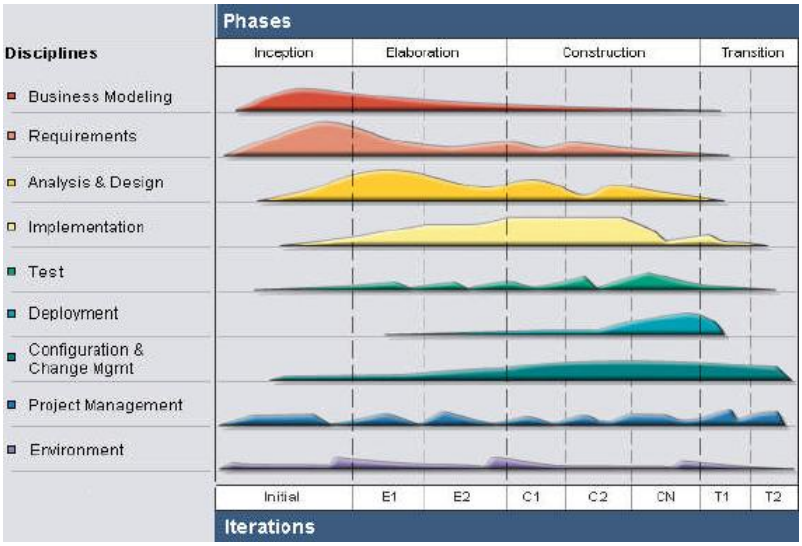


Figura 2.5: Dimensiones de modelado del RUP de [RUP]

La gráfica en la figura 2.5 muestra como el énfasis del desarrollo varía en el tiempo. Por ejemplo, en las iteraciones tempranas se utiliza más tiempo en la Disciplina de Requerimientos, mientras que en iteraciones más tardías se utiliza más tiempo en la Disciplina de Implementación. A continuación se describen las dos dimensiones del RUP.

Dimensión del Tiempo

En la dimensión del tiempo se definen ciclos, fases e iteraciones. Un ciclo constituye una realización completa de un proyecto de desarrollo con una liberación de una versión final del sistema en construcción, que constituye una generación del producto. Cada ciclo se divide a su vez en cuatro fases que se nombran como Inicial, Elaboración, Construcción y Transición, cada una de las cuales tiene objetivos que deben cumplirse en cuanto a realización de principales actividades y obtención de entregables más importantes. Estos objetivos son evaluados al finalizar la fase según lo planificado, si son obtenidos es posible realizar el pasaje a la fase siguiente, en caso de no ser obtenidos se deberá replanificar el proyecto extendiendo la fase que corresponda hasta que éstos objetivos sean cumplidos. Para esta replanificación será necesario definir si es posible o no absorber el atraso en fases siguientes o será necesario extender la fecha de finalización del proyecto.

Fases y sus principales Objetivos

En la Fase Inicial el foco del desarrollo se encuentra en la comprensión global del problema y el relevamiento de principales requerimientos, así como la planificación inicial del proyecto y la identificación de los riesgos del proyecto. Los criterios de evaluación al finalizar la Fase incluyen acuerdo con el cliente en la definición de alcance y estimaciones de costo y agenda para el proyecto, el conjunto adecuado de requerimientos fue capturado y existe una comprensión compartida de dichos requerimientos, las estimaciones de costo y agenda, identificación de prioridades y riesgos y el proceso de desarrollo definidos son apropiadas, todos los riesgos han sido identificados y existen estrategias de mitigación para cada uno. Si los objetivos son conseguidos, se deberá realizar la planificación en detalle de la Fase de Elaboración siguiente. Si alguno de los criterios no es alcanzado, el proyecto deberá ser replanificación-pensado considerablemente o abortado.

En la Fase de Elaboración el principal objetivo consiste en la construcción de la llamada Línea Base de la Arquitectura, compuesta por los Casos de Uso más importantes o escenarios principales de los mismos, según los Casos de Uso relevantes a la Arquitectura identificados, lo que constituye el esqueleto de la aplicación sobre el cual se construirán el resto de las funcionalidades. Los criterios de evaluación de la Fase incluyen que los requerimientos y la Arquitectura son estables, la evaluación y testing de los prototipos ejecutables ha demostrado que los elementos de mayor riesgo han sido tratados y resueltos, la planificación en detalle de la Fase de construcción permiten la continuación del trabajo, los involucrados acuerdan que la visión del sistema puede ser alcanzada al ejecutar el plan para desarrollar el sistema completo, en el contexto de la Arquitectura definida. Si no se alcanzan los objetivos el proyecto deberá ser repensado o incluso abortado.

En la Fase de Construcción el objetivo más importante es la construcción y testeo del sistema según lo acordado, obteniendo paralelismo en el desarrollo y mejorando la productividad en el área de implementación, así como realizando el testeo completo de las sucesivas liberaciones que se van obteniendo en las iteraciones intermedias. Los criterios de evaluación para la Fase de Construcción incluyen que toda la funcionalidad haya sido desarrollada y el testing alpha se haya completado, así como el manual de usuario y la descripción de la liberación actual, esta liberación es estable y puede ser entregada a la comunidad de usuarios, y todos los involucrados están preparados para dicha transición. Si alguno de estos criterios no es alcanzado la Fase de Transición deberá posponerse.

En la Fase de Transición el foco se encuentra en la Disciplina de Implantación en la cual se implanta el producto en el ambiente del cliente y se realizan las pruebas de aceptación sobre el sistema definidas con el cliente. Los criterios de evaluación para la Fase de Transición incluyen la revisión y aceptación de los entregables del proyecto por parte del cliente, así como la

satisfacción del mismo con el resultado obtenido. Luego de este último hito el producto es puesto ten producción y el ciclo de mantenimiento post liberación comienza. Esto puede involucrar comenzar un nuevo ciclo de desarrollo o alguna liberación de mantenimiento adicional.

Cada fase se compone de iteraciones cuya duración está definida según la planificación del proyecto realizada al inicio del mismo. En dichas iteraciones se realiza un recorrido completo por todas las Disciplinas del proceso, en forma de mini-cascada, para el subconjunto de funcionalidades del producto bajo desarrollo en dicha iteración, expresadas como Casos de Uso y que componen la liberación asociada a esa iteración. Se realizan las actividades definidas para cada Disciplina según la Fase a la que corresponde la iteración, generando los entregables asociados según la importancia de los mismos para la obtención de los objetivos definidos para la Fase.

Dimensión de las Disciplinas

En la dimensión de las Disciplinas se describen los aspectos estáticos del proceso, en base a cuatro elementos que hacen a la definición de proceso de desarrollo: quién está haciendo qué, de que forma (como) y en qué momento (cuando). Las Disciplinas describen cuando y agrupan actividades en forma lógica, donde las Actividades definen el como; los Roles describen quién y las responsabilidades asociadas, y los Entregables describen qué, cuales son los artefactos o productos que serán generados como resultado de la realización de las actividades.

Disciplinas, Actividades, Entregables y Roles

Las Disciplinas definidas por el RUP son: Modelado del Negocio, Requerimientos, Análisis&Diseño, Implementación, Testing, Deployment, Gestión de Configuración y Cambios, Gestión de Proyecto, y Entorno o Ambiente. Para cada Disciplina se definen las actividades y entregables que corresponden a la misma, la participación de los roles en las actividades y entregables correspondientes, y cuando se deben realizar (Fase e iteración) las actividades definidas. Algo que puede llamar la atención es que no exista una Disciplina para el Aseguramiento de la Calidad. El RUP es consciente de este hecho y argumenta que las actividades que podrían incluirse en dicha Disciplina se realizan en cada una de las otras Disciplinas mediante revisión de artefactos, trazabilidad entre los modelos y verificación de elementos. Una visión del RUP y este tema se puede ver en [RSQA05].

Una actividad se define como el conjunto de acciones en una disciplina para crear o actualizar uno o varios entregables, y consta de pre-condiciones que son los entregables de entrada, y post-condiciones que son los entregables de salida, así como de una descripción de las tareas a realizar en el marco de la actividad y los roles participantes en la misma.

Un entregable o artefacto se define como producto tangible del proyecto que es entrada y salida de las actividades, tiene una descripción del contenido a generar y en general se provee una plantilla (o template) como guía para su realización.

Un rol define el comportamiento y las responsabilidades de la/s persona/s que lo cumplen, el comportamiento está dado por las actividades en las que participa, y las responsabilidades están dadas por los entregables que tiene asociados. Un rol puede ser cumplido por más de una persona, y una persona puede cumplir más de un rol en un proyecto de desarrollo.

Cada flujo de trabajo que describe una Disciplina se modela como Diagrama de Actividad de UML mostrando la secuencia y/o paralelismo para la realización de las actividades involucradas en la Disciplina. Luego cada Actividad se modela definiendo los entregables de entrada y salida asociados, los roles participantes y sus responsabilidades, y si aplica, alguna herramienta de soporte para la realización de la actividad. Para los Entregables se brindan plantillas (templates), listas de verificación sobre el contenido y forma de los mismos, y reportes ejemplo si aplica. En la figura 2.6 se muestra como ejemplo los elementos de modelado para la Actividad Análisis arquitectónico.

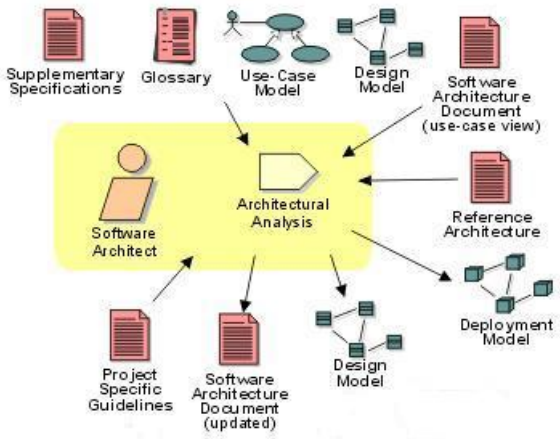


Figura 2.6: Ejemplo modelado de actividad Análisis Arquitectónico en el RUP de [RUP]

Como se puede observar varios de los entregables y de salida son los mismos, esto es porque al ser un proceso iterativo incremental, los entregables no se realizan completos en cada ejecución de la actividad sino que se va iterando también de acuerdo a lo que se va definiendo en cada momento. Específicamente la primera vez que se realiza esta actividad, el Modelo de Casos de Uso tendrá algunos de los Casos de Uso más importantes descriptos con mayor detalle que otros que solamente estarán esbozados, los Modelos de Diseño y Deployment no contendrán casi información, al salir de la actividad se habrá incrementado la información en todos los Modelos y en el Documento de la Arquitectura en las vistas correspondientes según la iteración.

En cuanto a los roles, el RUP define más de treinta, agrupados según la siguiente clasificación: Analistas, Desarrolladores, Gestores, Producción y Soporte, Testers y Roles adicionales, los que no se corresponden en relación uno a uno con las personas en el desarrollo, esto es, una persona puede realizar más de un rol en determinados momentos y un rol puede ser cumplido además por más de una persona a la vez. Un ejemplo claro es el de los roles clásicos de los primeros grupos mencionados, donde se encuentran los roles de Analista de Sistemas, Diseñador, Arquitecto de Software e Implementador, entre otros.

2.3.2. eXtreme Programming (XP)

Las metodologías ágiles se basan en los principios establecidos en el llamado "Manifiesto para Desarrollo de Software Ágil" [AGMA01] escrito y firmado en el año 2001 por varios autores, entre ellos Kent Beck, Martin Fowler, Steven Mellor, Alistair Cockburn. El principio fundamental definido en dicho manifiesto establece que: "Estamos descubriendo nuevas formas de desarrollar software haciéndolo y ayudando a otros a hacerlo. A través de este trabajo hemos llegado a valorar: Individuos e interacciones sobre procesos y herramientas, Software funcionando sobre documentación comprensible, Colaboración con el Cliente sobre negociación de contratos, Respuestas al cambio sobre seguimiento de un plan. Esto es, mientras que hay valor en los ítems de la derecha, valoramos más los ítems en la izquierda."

Como procesos ágiles se pueden mencionar entre otros, SCRUM [SCRU], DSDM [DSDM], y XP [BECK99]. Según [XPIN] "eXtreme Programming (XP) es una de varias metodologías "livianas" o ágiles. XP tiene unas pocas reglas y un número modesto de prácticas, las que son fáciles de seguir. XP es un entorno limpio y conciso desarrollado mediante la observación de que hace que el desarrollo de software sea más rápido y que hace que sea más lento. Es un entorno en el cual los programadores se sienten libres de ser creativos y productivos mientras que a la vez permanecen organizados y enfocados." XP es apropiado principalmente para equipos de desarrollo de pequeño tamaño, recomendado hasta diez personas, realizando el desarrollo en un mismo lugar físico, donde el cliente forma parte del equipo de desarrollo.

2.3.2.1. Principales características

eXtreme Programming (XP) [BECK99] se basa en un conjunto de consideraciones sobre los problemas que se enfrentan al desarrollar software: los riesgos que deben ser tratados en todos los niveles del proceso de desarrollo, las cuatro variables involucradas: costo, tiempo, calidad y alcance, promoviendo el uso de la variable alcance como mejor forma de control del desarrollo, el costo del cambio donde como premisa se toma que el costo del cambio no crece exponencialmente con el tiempo como es la asunción tradicional de la Ingeniería de Software, sino que lo hace mucho más lento, alcanzando eventualmente una asíntota "achatando" este costo y manteniéndolo constante, los cuatro valores pilar del desarrollo: comunicación, simplicidad, feedback y coraje, que son la base para cubrir las necesidades tanto humanas como comerciales en un proyecto, enfatizando las cuatro actividades básicas y más importantes en cualquier desarrollo: implementar, testear, escuchar, diseñar; al final del día con XP tiene que haber un programa funcionando.

Además se identifican principios fundamentales y adicionales donde cada principio cubre los valores mencionados y son la base de las doce prácticas definidas en XP, los principios fundamentales se definen como: *feedback rápido*, *asumir simplicidad*, *cambio incremental*, *abrazar el cambio* y *trabajo de calidad*, y los principios adicionales son: *enseñar a aprender*, *inversión inicial pequeña*, *jugar para ganar*, *experimentos concretos*, *comunicación abierta y honesta*, *trabajar con los instintos de la gente no en contra de éstos*, *aceptar la responsabilidad*, *adaptación local*, *viajar ligero* y *mediciones honestas*, y finalmente. Entre estos principios se puede destacar el de viajar ligero como identificatorio de la filosofía XP, que pone el énfasis en que los artefactos a realizar y mantener debieran ser: pocos, simples, valiosos; la idea es ser "nómades intelectuales" preparados para levantar campamento y seguir, por ejemplo frente a un cambio en el diseño o en los requerimientos, llevando siempre lo de mayor valor para el cliente: tests y código.

2.3.2.2. Elementos de modelado

eXtreme Programming define doce prácticas como las actividades básicas del desarrollo que deben realizarse, las cuales proveen soluciones a las consideraciones sobre el desarrollo que plantea el enfoque, contando con el supuesto de la curva de costo "achatada" que permite la definición y ejecución de las prácticas propuestas. Estas prácticas se ejecutan en las Fases e iteraciones que define XP para el desarrollo, donde el resultado de cada iteración es un sistema probado listo para poner en producción. XP define también roles para los distintos involucrados en el desarrollo. A continuación se presentan brevemente los elementos de modelado que se definen en XP.

Las doce prácticas

Las doce prácticas que define XP cubren las consideraciones presentadas en 2.3.1.1 sobre el desarrollo de software, y proveen un enfoque de trabajo para tratarlas. Todas estas prácticas componen las distintas estrategias utilizadas por XP para la gestión, planificación, implementación, diseño y testing del sistema, y se apoyan entre si, donde una práctica es débil, la fortaleza de otras prácticas cubrirá esta debilidad. A continuación se presentan brevemente las doce prácticas de XP.

El juego de la planificación: tiene como objetivo maximizar el valor del software producido por el equipo, con la estrategia de invertir lo menos posible para poner en producción lo antes posible las funcionalidades más valiosas, en conjunción con las estrategias de diseño y programación para minimizar los riesgos. El juego de planificación se compone de tres fases: exploración, compromiso y dirección, donde el elemento de planificación son las historias de usuarios que describen lo que debe hacer el sistema según el cliente.

Liberaciones pequeñas: cada liberación debe ser lo más pequeña posible conteniendo los requerimientos de mayor valor para el negocio, pero debe tener sentido como un todo. Los ciclos de liberación deben planificarse lo más cortos posibles dentro del contexto del proyecto.

Metáfora: representa el mismo concepto que la Arquitectura de Software, donde se enfatiza el objetivo de la Arquitectura, que es darle a cada involucrado una historia coherente en la cual trabajar, una historia que pueda ser fácilmente compartida por los involucrados técnicos y del negocio en el proyecto, y que pueda comunicarse con facilidad.

Diseño simple: el diseño correcto del software en cada momento es el que cumple: ejecuta todos los tests, no tiene lógica duplicada, establece cada intención importante para los programadores, y tiene la menor cantidad posible de clases y métodos. Cada pieza en el diseño debe poder justificar su existencia en esos términos. El objetivo es diseñar lo que se necesita cuando se necesita.

Testing: cualquier característica del programa que no tiene un test automatizado asociado simplemente no existe. Los programadores escriben tests de sus programas y los clientes escriben test funcionales, y todos se vuelven parte del programa. El programa resultante se vuelve cada vez más confiable y más capaz de aceptar cambios.

Refactoring: cada vez que se agrega una característica se intenta ver si hay una forma de cambiar el programa existente para agregar más fácilmente la característica, luego de agregarla se ve si es posible entonces encontrar una forma de hacer el programa más simple, y que los tests sigan ejecutando. Este enfoque se llama refactoring, y va de la mano con el diseño simple.

Programación en pares: todo el código en producción es escrito por dos programadores en una máquina, con un único teclado y un único mouse. En cada par hay dos roles: uno con el teclado y el mouse está pensando como implementar de mejor forma el método en cuestión, el otro está pensando en forma más estratégica sobre si ese enfoque funcionará, si hay otros tests que puedan no ejecutar, si hay forma de simplificar el sistema entero para solucionar el problema puntual. Los pares son dinámicos, pueden cambiar en el día inclusive.

Propiedad colectiva: todos los programadores toman responsabilidad por el sistema completo, si bien no todos tienen conocimiento en detalle de todas las partes si tienen conocimiento de todas las partes. Cualquiera que vea una oportunidad de agregar valor a una porción de código es requerido que lo haga en cualquier momento. Si un par programando ve una oportunidad de mejorar el código, deben mejorarlo si es la mejor opción.

Integración continua: el código es integrado y testeado cada pocas horas, como mucho al final del día de trabajo. Una estrategia para poder hacer esto es tener una máquina dedicada para la integración, donde los pares van agregando sus cambios a la liberación actual, chequeando y resolviendo cualquier colisión, y ejecutando los tests hasta que el 100 % se ejecuten en forma correcta.

40 horas por semana: trabajar más de 40 horas por semana difícilmente permite que las personas estén frescas y puedan realizar sus tareas en forma creativa, con confianza y cuidado. Para XP el sobretiempo es un síntoma de problemas serios en el proyecto, la regla en XP es no trabajar dos semanas seguidas con sobrededicación, esto es una semana con horas extras bien, pero no se puede mantener para la siguiente. En el caso que se requiera hay que revisar el problema puesto que no es resoluble trabajando más horas.

Cliente en el lugar: un cliente de los que va a utilizar el sistema cuando esté en producción debe estar con el equipo de desarrollo, disponible para responder preguntas, resolver disputas, y establecer prioridades de pequeña escala. Generalmente estos usuarios del sistema son demasiado valiosos para dárselos al equipo de desarrollo, en ese caso se debe evaluar si entonces tener el sistema funcionando es tan importante, y dará más valor al negocio que una persona más trabajando.

Estándares de programación: para poder realizar las prácticas de refactoring, propiedad colectiva y programación de a pares, se debe tener el código estandarizado, no es posible tener distintas prácticas de programación. Este estándar debe requerir la menor cantidad de trabajo posible, debe ser consistente con la regla de no tener código duplicado, debe enfatizar la comunicación y debe ser adoptado en forma voluntaria por todo el equipo.

Fases e iteraciones

El ciclo de vida en XP también se modela con Fases e iteraciones. Las Fases que define XP son cinco: Exploración, Planificación, Producción, Mantenimiento y Muerte. A continuación se comentan brevemente los objetivos principales para cada una de las fases definidas.

En la Fase de Exploración todos los involucrados deben practicar y aprender sobre sus tareas. Los programadores deben explorar posibilidades para la arquitectura del sistema, deben probar y evaluar las tecnologías que se usarán en el desarrollo y producción del sistema, estimar las distintas tareas de programación que realicen en la fase, entre otras. Los clientes tienen que practicar la escritura de las historias para que éstas les sean útiles a los programadores tanto para implementarlas como para estimar el tiempo de implementación.

La Fase de Planificación tiene como principal objetivo que los clientes y los programadores acuerden una fecha en la cual el conjunto más pequeño y valioso de historias sea realizado, poniendo en práctica el juego de la planificación. La agenda acordada luego se divide en iteraciones de entre una y cuatro semanas, donde las historias construidas tienen asociadas un conjunto de tests funcionales. La primer iteración pone énfasis en la Arquitectura; eligiendo historias que fuercen a crear todo el sistema aunque sea en forma de esqueleto. El resto de las historias se van eligiendo según las prioridades del cliente, y al finalizar cada iteración los tests funcionales deberían ejecutar. Al finalizar esta Fase el sistema está listo para entrar en producción.

En la Fase de Producción las iteraciones serán más cortas, posiblemente de una semana, donde diariamente se tendrán reuniones para que todos los integrantes del equipo sepan en que está trabajando el resto. Se realizarán nuevos tests y se ajustará la performance del sistema, con el conocimiento del diseño y las estimaciones más realistas sobre la carga del sistema en producción y el hardware de producción disponible. La introducción de cambios será más lenta y posiblemente sea mejor realizar una lista de cambios posibles para implementar luego de que la liberación sea puesta en producción.

La Fase de Mantenimiento es el estado normal de un proyecto XP. Se produce nueva funcionalidad, se mantiene ejecutando el sistema existente, se incorporan nuevas personas al equipo y se despiden otras, simultáneamente. Cada nueva liberación comienza con la Fase de exploración, donde se incorporan refactorings, se prueban nuevas tecnologías, se experimentan nuevas ideas sobre la Arquitectura y el cliente escribe nuevas historias. Sin embargo, los cambios a un sistema en producción se deben realizar con más cuidado, interrumpiendo el desarrollo para atender problemas de producción, por lo que la velocidad de implementación del equipo se reduce.

La Fase de Muerte aparece cuando el cliente no tiene más nada que agregar al sistema, está satisfecho con el mismo, por lo que es importante en ese momento escribir un documento del sistema que ayude a incorporar en un futuro cambios o agregados nuevos. Otra razón para entrar en esta Fase puede ser que el sistema ya no está funcionando como debe, no es posible económicamente agregar nuevas características y la tasa de defectos crece en forma intolerable. Aunque con los proyectos XP se espera no llegar a ese momento, si sucede se debe dejar morir al sistema.

Roles

En XP no se definen tantos roles como en el RUP, sino más bien unos pocos. Los roles definidos son: programador, cliente, verificador, encargado de registrar, entrenador, consultor y gerente.

2.4. Modelos de mejora de procesos

Los modelos de mejora de procesos utilizan o proveen modelos de procesos para los procesos a realizar en una Organización, modelos de evaluación para determinar la capacidad de estos procesos, siguiendo distintos enfoques para la evaluación de estas capacidades, y métodos de evaluación para realizar las evaluaciones. Trabajan con la premisa de que "la calidad de un producto de software es influenciada principalmente por el proceso utilizado para desarrollarlo"

[CMMI06], [ISO98], por lo que para mejorar la calidad de un producto de software es importante mejorar la calidad del proceso de software que se utiliza para su desarrollo. Para mejorar los procesos de desarrollo debe ser posible evaluar la capacidad actual de los procesos y plantear un camino de mejora a seguir para incrementar dicha capacidad.

A continuación se presentan tres modelos de mejora de procesos, los dos primeros por haber aportado en la primer definición del proceso base [ADBP00] utilizado para la propuesta de la metodología SOA realizada, y el tercero por la importancia que su definición tiene para la industria de software en iberoamérica. En la sección 2.4.1 se presenta la norma Norma ISO/IEC 15504 - Evaluación de Procesos de Software, en la sección 2.4.2 se describe el Capability Maturity Model Integration (CMMI) del SEI [SEI], y finalmente en la sección 2.4.3 se introduce el marco metodológico COMPETISOFT que es una iniciativa en desarrollo para las PYMES iberoamericanas, para el cual se definió el Perfil SOA (PSOA) como generalización de la metodología SOA propuesta.

2.4.1. Norma ISO/IEC 15504 - Evaluación de procesos de Software

La norma ISO/IEC 15504 - Evaluación de procesos de Software es un estándar en cinco partes que como se establece en [ISO98] parte 1, "define un marco para la evaluación de procesos de software. Este marco puede ser utilizado por organizaciones involucradas en la planificación, gestión, monitoreo, control y mejora de la adquisición, provisión, desarrollo, operación, evolución y soporte de software. La evaluación de procesos examina los procesos utilizados por una Organización para determinar si éstos son efectivos en alcanzar sus objetivos. La evaluación caracteriza las prácticas actuales en una unidad organizacional en términos de la capacidad de los procesos seleccionados. Los resultados pueden ser utilizados para guiar actividades de mejora de procesos o determinación de capacidad de procesos analizando los resultados en el contexto de las necesidades de negocio de la Organización, identificando fortalezas, debilidades y riesgos inherentes a los procesos".

La capacidad de los procesos se mide según el cumplimiento de atributos definidos para cada nivel de capacidad, que son valorados en una escala porcentual que se define. La evaluación de procesos se realiza en términos de los propósitos y salidas esperadas definidos para los procesos, por lo que lo que se mide es lo que debe alcanzarse sin establecer como debe hacerse. Los niveles de capacidad proveen un camino a seguir para la mejora de los procesos en la Organización, a medida que los procesos van incrementando el cumplimiento de los atributos definidos en cada nivel. Además define el proceso de evaluación que debe seguirse, indicando los elementos que debe presentar, así como los roles y productos de entrada y salida para el mismo, definiendo las habilidades que deben presentar los asesores en la evaluación.

La norma ISO/IEC 15504 comenzó con el proyecto Software Process Improvement and Capability dEtermination (SPICE) a principios de los años 90, que resultó en el ISO/IEC-TR-15504 (Reporte Técnico) en el año 1998 que constaba de nueve partes. En el año 2003 concluye SPICE comenzando la liberación de partes de la norma con ISO/IEC 15504-2 - Realización de una evaluación, que cancela las partes 2 y 3 del TR del 98, dado que el modelo de referencia de procesos fue incorporado a la norma 12207 junto con otras modificaciones realizadas en las enmiendas 2002 y 2004 como se presentó en 2.2.2.1. ISO/IEC 15504 -2 - Realización de una evaluación, establece los requerimientos para los modelos de procesos de referencia y los métodos de evaluación a utilizar, sin especificar ninguno en particular, y define el marco para la medición de la capacidad de los procesos.

En la primer versión del proceso base [ADBP00] en ese momento adaptación del Unified Process(UP) [USDP99], definida en el marco del programa de construcción y prueba de procesos del [GRIS06] en el año 2000, se tuvieron en cuenta varios elementos de SPICE para complementar dicha adaptación, principalmente en lo que tiene que ver con las Disciplinas de Soporte como Gestión de Calidad y Gestión de Configuración. A continuación se describen los niveles de capacidad de ISO/IEC 15504 en la sección 2.4.1.1 para evaluación de la capacidad de los procesos según los atributos y escala de valoración definidas.

2.4.1.1. Niveles de capacidad en ISO/IEC 15504

Se plantea un modelo en dos dimensiones para la evaluación de los procesos, donde en la primera versión se incluían en una dimensión los procesos del modelo de referencia de procesos provisto como parte de la norma, y en la otra dimensión la evaluación de la capacidad para cad proceso mediante la valoración de los atributos de los procesos definidos, según la escala definida. Actualmente la medida de la capacidad de los procesos es aplicable a cualquier modelo de referencia de procesos compatible con el incorporado en la norma ISO/IEC 12207.

Se definen nueve atributos de procesos, agrupados en niveles de capacidad, que representan una evolución incremental para la mejora de la capacidad de los procesos. Los atributos son características de los procesos que pueden ser evaluados en una escala de cumplimiento, que proveen una medida de la capacidad de los procesos y son aplicables a todos los procesos. Los niveles de capacidad definidos son seis numerados desde el 0 hasta el 5. Para ser evaluados, los procesos deben describirse estableciendo el propósito del proceso y las salidas esperadas de su implementación. A continuación se presentan los atributos de procesos y niveles de capacidad definidos.

- Nivel 0 - Incompleto: no se alcanza el propósito del proceso. Las salidas del proceso y los productos de trabajo obtenidos son pocos o no son fácilmente identificables.
- Nivel 1 - Realizado: el propósito del proceso es alcanzado en general. El cumplimiento puede no ser planificado ni monitoreado, los individuos en la Organización reconocen cuando una acción debe realizarse y hay acuerdo general en que se realiza cuando y como debe hacerse. Hay productos de trabajo del proceso identificables que testifican el cumplimiento del proceso.
- Nivel 2 - Gestionado: el proceso libera productos de trabajo siguiendo procedimientos especificados y es planificado y monitoreado. Los productos de trabajo cumplen estándares y requerimientos establecidos. El proceso libera productos de trabajo que cumplen completamente con los requerimientos de calidad establecidos en tiempo y con los recursos necesarios.
- Nivel 3 - Establecido: el proceso se realiza y es gestionado siguiendo un proceso definido basado en buenos principios de Ingeniería de Software. Las implementaciones individuales del proceso utilizan versiones aprobadas y adaptadas de estándares, procesos documentados, para alcanzar las salidas del proceso. Se cuenta con los recursos necesarios para establecer la definición del proceso.
- Nivel 4 - Predecible: el proceso definido es realizado consistentemente en la práctica con límites de control definidos, para alcanzar los objetivos del proceso. Se recolectan y analizan mediciones detalladas del desempeño del proceso, lo que lleva a una comprensión cuantitativa de la capacidad del proceso y mejora la habilidad de predecir y gestionar el desempeño. La calidad de los productos de trabajo es conocida en forma cuantitativa.
- Nivel 5 - Optimizante: el desempeño del proceso es optimizado para cubrir las necesidades del negocio actuales y futuras, y el proceso es repetible en el cumplimiento de sus objetivos de negocio definidos. En base a estos se establecen objetivos de efectividad y eficiencia del proceso, que es monitoreado contra sus objetivos y sus resultados analizados para realizar la mejora. Optimizar un proceso involucra hacer pruebas piloto de ideas y tecnologías innovadoras y cambiar procesos que no son efectivos en alcanzar sus objetivos definidos.

Para determinar la capacidad de los procesos se utilizan los atributos donde cada uno mide un aspecto particular del proceso. Los atributos se miden en una escala porcentual que brinda una visión detallada de los aspectos específicos de la capacidad del proceso requerida. En la tabla 2.1 se presentan los atributos por nivel de capacidad.

Nivel de capacidad	Atributo de proceso	Descripción
1	1.1 - Desempeño del proceso	Transforma productos de trabajo de entrada en productos de salida identificables
2	2.1 - Gestión del desempeño del proceso	Es gestionado para generar productos de trabajo que cumplen objetivos definidos, en tiempo y con los recursos necesarios
2	2.2 - Gestión de los productos de trabajo del proceso	Es gestionado para generar productos de trabajo que son documentados, controlados y verificados, y cumplen con sus requerimientos definidos
3	3.1 - Definición del proceso	Utiliza un proceso estándar definido, se registran experiencias de uso y desempeño del proceso
3	3.2 - Despliegue del proceso	Utiliza eficientemente los recursos humanos y materiales asignados para despliegue del proceso
4	4.1 - Medición del proceso	Utiliza objetivos del producto y del proceso y mediciones para asegurar que el desempeño del proceso cumple los objetivos definidos contribuyendo a los objetivos del negocio
4	4.2 - Control del proceso	Es controlado mediante la recolección, análisis y uso de mediciones del producto y del proceso para corregir el desempeño del proceso y cumplir con objetivos definidos del producto y proceso
5	5.1 - Innovación del proceso	Los cambios en la definición, gestión, y desempeño del proceso son controlados para alcanzar los objetivos de negocio relevantes
5	5.2 - Optimización del proceso	Los cambios al proceso son identificados e implementados asegurando la mejora continua en el cumplimiento de los objetivos de negocio relevantes

Cuadro 2.1: Definición de atributos por nivel de capacidad en ISO/IEC 15504

Como cada nivel de capacidad se corresponde con determinados atributos, según sea el cumplimiento de éstos se obtiene el nivel de capacidad para el proceso. Para determinar el cumplimiento de los atributos definidos por nivel de capacidad, se define la escala porcentual para la asignación de cumplimiento de los atributos. La escala para valoración de atributos se establece mediante franjas de cumplimiento del atributo evaluado según sea: No alcanzado (N) del 0 al 15 %, Parcialmente alcanzado (P) entre el 16 y 50 %, Ampliamente alcanzado (L por Largely) entre 51 y 85 % y Totalmente alcanzado (F de Fully) entre 86 y 100 %.

Luego de realizar la valoración de atributos de los procesos según esta escala, se debe determinar el nivel de capacidad a asignar al proceso evaluado. Para esto, se define para cada nivel de capacidad, la correspondencia con la escala de valoración para los atributos de evaluación, entonces por ejemplo, para que el proceso esté en el nivel de capacidad 3, debe tener valoración F para todos los atributos en niveles inferiores: desempeño del proceso, gestión del desempeño del proceso, gestión de los productos de trabajo del proceso, y L o F para los atributos en el nivel evaluado: definición y despliegue del proceso.

Entonces, para alcanzar el nivel de capacidad al que corresponde el atributo, éste puede ser valorado en los dos últimos valores de la misma (Largelly, Fully), pero todos los atributos correspondientes a niveles de capacidad inferiores al que se está valorando, deben ser valorados solo en el último valor de la escala (Fully). Esto muestra como la evolución en los niveles de capacidad implica ir alcanzando y estableciendo los atributos del proceso en niveles inferiores para construir la mejora.

2.4.2. Capability Maturity Model Integration (CMMI)

El Capability Maturity Model Integration (CMMI) como se establece en [CMMI06] "es un modelo de mejora de la madurez de los procesos para el desarrollo de productos y servicios.

Consiste de las mejores prácticas para las actividades de desarrollo y mantenimiento que cubren el ciclo de vida de un producto desde su concepción hasta su liberación y mantenimiento." Establece un marco común para realizar evaluación de procesos de la Organización, tanto sea de desarrollo como para la contratación de empresas.

Provee una guía para seleccionar estrategias de mejora de los procesos adecuadas a la Organización, determinando el nivel de capacidad de los procesos o madurez de la Organización y definiendo un camino evolutivo de mejoras recomendado para incrementarlo. En cada nivel de capacidad o madurez, los procesos cumplen determinados objetivos específicos y/o genéricos, que proveen las bases para la mejora de dichos procesos. Es un modelo descriptivo ya que describe los atributos claves necesarios para la realización de los procesos, pero no indica de que forma se deben obtener, permitiendo que cada Organización implemente sus procesos en la forma que le resulte más adecuada.

CMMI es uno de los productos del Software Engineering Institute (SEI) [SEI] que desde su concepción está en constante evolución. CMMI para Desarrollo (CMMI-DEV), Versión 1.2, fue liberado en agosto de 2006, y es el resultado de la evolución del Capability Maturity Model Integration (CMMI-SW), Versión 1.1, 2002, compuesto por dos documentos con las representaciones continua y escalonada para evaluación de procesos, y la integración de otros modelos de evaluación y mejora. Este a su vez, evolucionó del Capability Maturity Model for Software (CMM), Versión 1.1, liberado en 1993, que solo contenía representación escalonada, teniendo en cuenta también el trabajo que se estaba realizando en las normas ISO/IEC 15504 e ISO/IEC 12207, como se indica en [KITS00]. Todas las versiones del CMMI se pueden ver en [SEI].

CMMI versión 1.2, incorpora el concepto de "constelaciones" cada una de las cuales es un conjunto de componentes del CMMI para un enfoque específico. CMMI para Desarrollo (CMMI-DEV) es una de las constelaciones definidas, otras dos que se encuentran en etapa de trabajo son CMMI para Servicios (CMMI-SVC) y CMMI para Adquisiciones (CMMI-ACQ). Se incluyen también "adiciones" que extienden una constelación para un enfoque específico, por ejemplo en CMMI-DEV para tratar el desarrollo integrado de productos y procesos, Integrated Product and Process Development (IPPD).

En la primer versión del proceso base [ADBP00] en ese momento adaptación del Unified Process(UP) [USDP99], definida en el marco del programa de construcción y prueba de procesos del [GRIS06] en el año 2000, se tuvieron en cuenta varios elementos del CMM v.1.1 para complementar dicha adaptación, por ejemplo para la definición de las Disciplinas de Gestión de Calidad y Gestión de Configuración. A continuación se describen los componentes del CMMI en la sección 2.4.2.1, y los niveles del CMMI en la sección 2.4.2.2 para evaluación de la capacidad de los procesos y madurez de la Organización en las representaciones continua y escalonada respectivamente.

2.4.2.1. Componentes del CMMI

El CMMI se compone de veintidós áreas de procesos, que se definen en [CMMI06] como "agrupación de prácticas relacionadas en un área que, cuando son implementadas colectivamente, satisfacen un conjunto de objetivos considerados importantes para la mejora en dicha área." Para cada área de procesos se describe su propósito, se proveen notas introductorias y áreas de procesos relacionadas, y se definen objetivos específicos que son alcanzados mediante la realización de prácticas específicas. A su vez, se establecen objetivos genéricos que aplican a múltiples áreas de procesos y que son alcanzados mediante la realización de prácticas genéricas. Para las prácticas específicas se definen también subprácticas y productos típicos de trabajo a obtener, para las prácticas genéricas se definen también subprácticas y elaboraciones genéricas de las prácticas que son guías de como realizarlas para un área de procesos específica. La presencia de estos elementos en el modelo se define en tres niveles: requerido, esperado, informativo.

Las veintidós áreas de procesos que componen el CMMI se agrupan en forma distinta según se utilice la representación continua o la representación escalonada. En la primera se definen

cuatro categorías que son: Gestión de Procesos, Gestión de Proyectos, Ingeniería y Soporte, para mostrar la relación entre las mismas, en la segunda se agrupan directamente en los niveles de madurez. Además en la representación continua se definen en cada categoría áreas de procesos básicas y avanzadas, donde las básicas deben realizarse antes que las avanzadas. En la tabla 2.2 se presentan las áreas de procesos del CMMI indicando categoría y nivel de madurez asociado.

Area de procesos	Categoría	Nivel de Madurez
Innovación y despliegue organizacional	Gestión de procesos	5
Análisis de causas y resolución	Soporte	5
Desempeño de procesos organizacionales	Gestión de procesos	4
Gestión cuantitativa de proyectos	Gestión de proyectos	4
Definición de procesos organizacionales(+IPPD)	Gestión de procesos	3
Foco en los procesos organizacionales	Gestión de procesos	3
Formación (entrenamiento) organizacional	Gestión de procesos	3
Gestión de riesgos	Gestión de proyectos	3
Gestión de proyectos integrada (+ IPPD)	Gestión de proyectos	3
Desarrollo de requerimientos	Ingeniería	3
Integración de productos	Ingeniería	3
Solución técnica (diseño, implementación)	Ingeniería	3
Validación	Ingeniería	3
Verificación	Ingeniería	3
Análisis de decisiones y resolución	Soporte	3
Control y monitoreo de proyectos	Gestión de proyectos	2
Planificación de proyectos	Gestión de proyectos	2
Gestión de acuerdos con proveedores	Gestión de proyectos	2
Gestión de requerimientos	Ingeniería	2
Aseguramiento de calidad del proceso y del producto	Soporte	2
Gestión de Configuración	Soporte	2
Análisis y mediciones	Soporte	2

Cuadro 2.2: Áreas de procesos del CMMI indicando categoría y nivel de madurez adaptado de [CMMI06]

2.4.2.2. Niveles del CMMI

En CMMI los niveles se utilizan para describir un camino de evolución recomendado para una Organización que quiere mejorar los procesos que utiliza para desarrollar y mantener sus productos y servicios, y son también la salida de las actividades de puntuación de las evaluaciones. CMMI presenta dos caminos de mejora: en uno se permite la mejora incremental de procesos correspondientes a un área de procesos o más seleccionada por la Organización; en el otro se permite la mejora de un conjunto de procesos relacionados mediante el tratamiento incremental de sucesivos conjuntos de áreas de procesos.

El primero corresponde a la representación continua y se utiliza el término nivel de capacidad para el proceso individual, el segundo corresponde a la representación escalonada y se utiliza el término nivel de madurez de la Organización. Sin importar la representación utilizada el concepto de niveles es el mismo: caracterizan la mejora desde un estado definido de partida hacia un estado que utiliza información cuantitativa para determinar y gestionar las mejoras que son necesarias en la Organización para alcanzar sus objetivos de negocio. Para alcanzar un nivel particular, una Organización debe satisfacer todos los objetivos apropiados para el área de procesos o conjunto de áreas de procesos que constituyen el objetivo de la mejora, sin importar si es un nivel de capacidad o de madurez.

Los niveles de capacidad que corresponden a la representación continua son una forma para la mejora incremental de los procesos correspondientes a un área de procesos determinada, y son seis numerados desde el 0 hasta el 5. Los niveles de madurez que corresponden a la representación escalonada, que aplican a múltiples áreas de procesos, son una forma de predecir la salida general del próximo proyecto que se realice, y son cinco numerados desde el 1 hasta el 5. Los niveles de capacidad y madurez numerados del 2 al 5 comparten el nombre entre sí

y con los términos manejados en los objetivos genéricos, que son los componentes del modelo que tratan con la institucionalización de los procesos y por lo tanto, los niveles de valoración están sumamente relacionados con la obtención de dichos objetivos. A continuación se presenta la definición y significado de los niveles para cada una de las representaciones.

Niveles de capacidad de la representación continua:

- Nivel 0 - Incompleto: un proceso incompleto es aquel que no es realizado o es parcialmente realizado. Uno o más objetivos específicos del proceso no se satisfacen, y no existen objetivos genéricos para este nivel ya que no hay razón de institucionalizar un proceso realizado en forma parcial.
- Nivel 1 - Realizado: un proceso realizado es aquel que satisface los objetivos específicos del área de procesos, soporta y permite el trabajo necesario para generar los productos de trabajo. Aunque este nivel resulta en mejoras importantes, estas mejoras pueden ser perdidas en el tiempo si no son institucionalizadas.
- Nivel 2 - Gestionado: un proceso gestionado es un proceso realizado que tiene la infraestructura básica necesaria para soportar el proceso. Es planificado y ejecutado de acuerdo a las políticas definidas, con personas capacitadas que tienen los recursos adecuados para producir salidas controladas, involucra a los participantes relevantes, es monitoreado, controlado y revisado, y es evaluado por adherencia a la descripción del proceso. La disciplina reflejada en este nivel ayuda a asegurar que las prácticas se mantienen bajo estrés.
- Nivel 3 - Definido: un proceso definido es un proceso gestionado que es adaptado a partir del conjunto de procesos estándar de la Organización, de acuerdo a guías de adaptación de la Organización, y contribuye con información de productos de trabajo, mediciones y otra información de mejora del proceso al conjunto de procesos organizacionales. Un proceso definido establece claramente el propósito, entradas, criterios de entrada, actividades, roles, mediciones, pasos de verificación, salidas y criterios de salida.
- Nivel 4 - Gestionado cuantitativamente: un proceso gestionado cuantitativamente es un proceso definido que es controlado utilizando técnicas estadísticas u otras técnicas cuantitativas. Se establecen objetivos cuantitativos para la calidad y desempeño del proceso que son utilizados como criterios para gestionar el proceso. La calidad y desempeño del proceso es comprendida en términos estadísticos y es gestionada a través de la vida del proceso.
- Nivel 5 - Optimizado: un proceso optimizado es un proceso gestionado cuantitativamente que es mejorado en base a la comprensión de las causas comunes de variación inherentes al proceso. El foco de un proceso optimizado es la mejora continua del rango de desempeño del proceso a través de mejoras incrementales e innovativas.

Niveles de madurez de la representación escalonada:

- Nivel 1 - Inicial: los procesos en este nivel son usualmente ad hoc y caóticos, y la Organización en general no provee un entorno estable para soportar los procesos. El éxito en estas organizaciones depende de la competencia y esfuerzos de las personas y no en el uso de procesos probados. A pesar de esto, estas organizaciones pueden producir productos y servicios que funcionan, pero generalmente exceden su presupuesto, no cumplen la agenda prevista y no pueden repetir el éxito en otros proyectos.
- Nivel 2 - Gestionado: los procesos son planificados y ejecutados de acuerdo a las políticas definidas, con personas y recursos adecuados para producir salidas controladas, involucra a los participantes relevantes, es monitoreado, controlado y revisado, y es evaluado por adherencia a la descripción del proceso. El estado de los productos de trabajo y liberación de servicios es visible a la gestión en puntos definidos, por ejemplo hitos, y cumplen las descripciones de los procesos especificados, estándares y procedimientos.

- Nivel 3 - Definido: los procesos están bien caracterizados y comprendidos, descritos en estándares, procedimientos, herramientas y métodos. Los proyectos adaptan los procesos definidos según guías de adaptación existentes, y tienen claramente establecidos el propósito, criterios de entrada, actividades, roles, mediciones, pasos de verificación, salidas y criterios de salida. La gestión de procesos se realiza más proactivamente utilizando la comprensión de las interrelaciones entre las actividades del proceso y medidas detalladas del proceso, sus productos de trabajo y servicios.
- Nivel 4 - Gestionado cuantitativamente: la Organización y los proyectos establecen objetivos cuantitativos para la calidad y desempeño del proceso que son utilizados como criterios para gestionar el proceso. Estos objetivos están basados en las necesidades del cliente, usuarios finales, Organización e implementadores del proceso. La calidad y desempeño del proceso es comprendida en términos estadísticos y es gestionada a través de la vida del proceso. El desempeño de los procesos es controlado utilizando técnicas estadísticas u otras técnicas cuantitativas, y es cuantitativamente predecible además de cualitativamente predecible como en el nivel 3.
- Nivel 5 - Optimizado: la Organización mejora continuamente sus procesos en base a la comprensión de las causas comunes de variación inherentes al proceso. Se enfoca en la mejora continua del desempeño de los procesos a través de mejoras incrementales e innovativas de los procesos y tecnologías. Se establecen objetivos de mejora de procesos cuantitativas, que se revisan continuamente para reflejar los cambios en los objetivos del negocio, y que son utilizados como criterio para la gestión de la mejora de los procesos. Los efectos de las mejoras introducidas a los procesos son medidos y evaluados contra los objetivos definidos.

Dada la existencia de las dos representaciones continua y escalonada del modelo, y la correspondiente medida de las áreas de procesos en niveles de capacidad o niveles de madurez de la Organización, CMMI provee una equivalencia entre ambos enfoques, de forma de poder realizar comparaciones entre ambas medidas. Para evaluar organizaciones con CMMI se utiliza el método Standard CMMI Assessment Method for Process Improvement (SCAMPI), uno de cuyos resultados es una puntuación, que en la representación continua es un perfil de niveles de capacidad y en la representación escalonada es un nivel de madurez. Un perfil de niveles de capacidad es una lista de áreas de procesos y el nivel de capacidad alcanzado por cada uno, este perfil puede ser de capacidades alcanzadas o de capacidades objetivo (a alcanzar). La equivalencia entre representaciones se define a través de una secuencia de perfiles objetivo en la representación continua, cada uno de los cuales es equivalente a un nivel de madurez en la representación escalonada, y puede verse en detalle en [CMMI06].

2.4.3. COMPETISOFT

COMPETISOFT tiene como objetivo según [COMP06] "desarrollar un Marco Metodológico común ajustado a la realidad socio-económica de las PYMES iberoamericanas, orientado a la mejora continua de sus procesos", dirigido a empresas o áreas internas dedicadas al desarrollo y/o mantenimiento de software. El Marco Metodológico está compuesto por un Modelo de Procesos, un Modelo de Evaluación de Procesos que incluye un Modelo de Capacidades y un Método de Evaluación, y un Modelo de Mejora de Procesos. Plantea servir de apoyo a las organizaciones para la estandarización de sus prácticas, la evaluación de la efectividad de sus procesos y la integración de la mejora continua, y constituir la base para alcanzar evaluaciones exitosas con otros modelos o normas, tales como ISO 9000:2000 [ISO00] o CMMI v.1.1.

El modelo de procesos sirve como modelo de referencia que especifica los elementos de los procesos definidos, de forma que éstos puedan ser evaluados utilizando el modelo de evaluación definido. En éste se definen niveles de capacidad de los procesos que se evalúan mediante atributos a conseguir que serán valorados según una escala definida, siguiendo la modalidad de

la norma ISO/IEC 15504. El proceso de mejora de procesos provee una guía para la mejora continua de los procesos en la Organización.

El Marco Metodológico toma aportes de varias iniciativas realizadas en forma previa en países de iberoamérica como México, Colombia y España, cuyas referencias pueden verse en [COMP06]. Se basa también en modelos y normas existentes, tales como ISO 9000:2000, niveles 2 y 3 del CMMI v.1.1, y la norma ISO/IEC 15504-2. El proyecto COMPETISOFT es financiado por el Programa Iberoamericano de Ciencia y Tecnología para el Desarrollo (CYTED) [CYTED] y en su desarrollo han participado más de 100 investigadores de 24 entidades y 13 países de ambos lados del Atlántico, entre ellas la Universidad de la República a través del Gris [GRIS06]. Este proyecto está aún en desarrollo, iniciando en setiembre de 2007 las pruebas controladas del Marco Metodológico definido en empresas de los países iberoamericanos integrantes del proyecto, donde cabe destacar la participación del Centro de Ensayos de Software (CES) [CES] en la puesta en práctica del enfoque por Uruguay.

A continuación se presentan los componentes del Marco Metodológico definido por COMPETISOFT, en 2.4.3.1 se presenta el Modelo de procesos, en 2.4.3.2 el Modelo de Evaluación de procesos y en 2.4.3.3 el Modelo de mejora de procesos.

2.4.3.1. Modelo de procesos de COMPETISOFT

El modelo de procesos que plantea COMPETISOFT está basado en el modelo mexicano Moprosoft [MOPR04] y define diez procesos agrupados en tres categorías que reflejan la estructura de una Organización: Alta Dirección, Gerencia y Operación. A continuación se presentan las categorías y los procesos definidos en cada una, según [COMP06].

- Alta Dirección: aborda las prácticas de Alta Dirección relacionadas con la gestión del negocio. Proporciona los lineamientos a los procesos de la Categoría de Gerencia y se retroalimenta con la información generada por ellos. Contiene el proceso de Gestión de Negocio.
- Gerencia: aborda las prácticas de gestión de procesos, proyectos y recursos en función de los lineamientos establecidos en la Categoría de Alta Dirección. Proporciona los elementos para el funcionamiento de los procesos de la Categoría de Operación, recibe y evalúa la información generada por éstos y comunica los resultados a la Categoría de Alta Dirección. Contiene los procesos de Gestión de procesos, Gestión de proyectos y Gestión de recursos. A su vez, el proceso de Gestión de recursos contiene los subprocesos de Gestión de Recursos humanos, Gestión de bienes, servicios e infraestructura y Gestión del conocimiento.
- Operación: aborda las prácticas de los proyectos de desarrollo y de mantenimiento de software. Esta categoría realiza las actividades de acuerdo a los elementos proporcionados por la Categoría de Gerencia y entrega a ésta la información y productos generados. Contiene los procesos de Administración de un proyecto específico, Desarrollo de Software y Mantenimiento de Software.

Para la documentación de los procesos en COMPETISOFT se utiliza el patrón de procesos que se define en [COMP06] como "un esquema de elementos que servirá para la documentación de los procesos. Está constituido por tres partes: Definición general del proceso, Prácticas y Guías de ajuste." La definición general del proceso identifica aspectos del mismo como nombre, categoría a la que pertenece, propósito, descripción general de actividades, objetivos, indicadores, entradas, salidas y productos intermedios, entre otros. En las prácticas se identifican los roles involucrados, se describen en detalle las actividades incluyendo aspectos definidos para las mismas como recursos de infraestructura necesarios, entre otros. En las guías de ajuste se sugieren modificaciones al proceso que no deben afectar sus objetivos.

Los productos de entrada, salida, internos y toda la sección de prácticas del modelo de procesos se colorea según el nivel de capacidad al que corresponde cada elemento, de esta forma se indica

en cada proceso lo que es necesario realizar para cada nivel de capacidad. El nivel 1 - Realizado corresponde al color amarillo, el nivel 2 - Gestionado al azul, el nivel 3 - Establecido al verde, el nivel 4 - Predecible al rosa y el nivel 5 - Optimizado no se colorea. La interpretación para la coloración de elementos se indica en [COMP06] y consiste en realizar primero los elementos marcados en amarillo, una vez alcanzados se agregan los marcados en azul y así sucesivamente hasta llegar al nivel 5. Para los elementos que están marcados en más de un color, significa que algunas partes se requieren en niveles distintos, por lo que, según el nivel se tomará el color correspondiente, o más de uno si es un nivel más alto de capacidad.

2.4.3.2. Modelo de Evaluación de procesos de COMPETISOFT

El modelo de evaluación de procesos de COMPETISOFT está basado en la norma mexicana EvalProSoft [EVAL05] que sigue la norma ISO/IEC 15504-2, tanto para la definición del modelo de capacidad de procesos como para los requisitos para la realización de la evaluación, y como guía la ISO/IEC TR 15504-4. Como en dicha norma se definen seis niveles de capacidad numerados del 0 al 5 a los que se asocian atributos de procesos que se valoran según una escala porcentual. Ya que la definición de niveles de capacidad sigue la presentada en [ISO98], se muestra en la tabla 2.3 un resumen de los mismos y sus correspondientes atributos de procesos.

Nivel de capacidad	Atributos de proceso y descripción
0 – Incompleto	El proceso no está implantado o falla en alcanzar el propósito del proceso.
1 – Realizado	El proceso implantado logra su propósito
	AP 1.1 - Realización del proceso
2 – Administrado	El proceso Realizado se implanta de manera administrada y sus productos de trabajo están apropiadamente establecidos, controlados y mantenidos
	AP 2.1 - Administración de la realización
	AP 2.2 - Administracion del producto de trabajo
3 – Establecido	El proceso Administrado es implantado mediante el proceso definido, el cual es capaz de lograr los resultados del proceso.
	AP 3.1 - Definición del proceso
	AP 3.2 – Implantación del proceso
4 – Predecible	El proceso Establecido opera dentro de limites para lograr sus resultados.
	AP 4.1 - Medición del proceso
	AP 4.2 - Control del proceso
5 - Optimizante	El proceso Predecible es continuamente mejorado para lograr las metas de negocio actuales y futuras relevantes.
	AP 5.1 - Innovación del proceso
	AP 5.2 - Optimización del proceso

Cuadro 2.3: Resumen de niveles de capacidad y atributos de procesos en COMPETISOFT

Como en la norma ISO/IEC 15504-2, se define una escala porcentual para la evaluación de atributos, y la correspondencia de dicha valoración con los niveles de capacidad definidos, las que son equivalentes, definiendo los mismos rangos de valoración de atributos como sigue: No alcanzado (N), Parcialmente alcanzado (P), Ampliamente alcanzado (A) y Completamente alcanzado (C), y de la misma forma, la correspondencia de cumplimiento de los atributos para establecer el nivel de capacidad.

2.4.3.3. Modelo de Mejora de procesos de COMPETISOFT

El modelo de mejora de procesos de COMPETISOFT se nombra PMCompetisoft - Proceso de Mejora de COMPETISOFT y está basado en el framework colombiano Agile SPI (Softwa-

re Process Improvement) [ASPI04]. Como se establece en [COMP06] "PMCompetiSoft es un proceso de mejora de procesos de software que guía la ejecución de un programa de mejora de procesos de software en pequeñas y medianas empresas (PYMES). Se caracteriza por ser liviano para su aplicabilidad en las PYMES de software."

Es un proceso, iterativo e incremental que se organiza en mini-proyectos de mejora para cubrir casos de mejora en el marco de un programa de mejora global, que se compone de cinco fases: Instalación, Diagnóstico, Formulación, Mejora y Revisión del Programa. Se define también utilizando el patrón de procesos que se utiliza para describir los procesos del modelo de procesos presentando en 2.4.3.1, y se ubica en el área de Gestión de procesos en la categoría de Gerencia. Utiliza la notación SPEM presentada en 2.2.3.1 para modelar sus elementos. En la figura 2.7 se muestra el modelado de las fases en PMCompetisoft.

Figura 2.7: Fases del proceso de mejora de COMPETISOFT de [COMP06]

En la Fase de instalación se crea una propuesta de mejora basada en las necesidades del negocio y se definen los objetivos de la mejora, la que debe ser aprobada por la Gerencia para asegurar la asignación de recursos necesaria. En la Fase de Diagnóstico se realiza una valoración de los procesos de la Organización, que puede ser realizada utilizando cualquiera de los modelos de mejora existentes, tales como [CMMI06] en cualquiera de sus representaciones, [ISO98], [COMP06], que permitirá establecer la prioridad de los casos de mejora y definir el plan general de mejora, que es uno de los productos principales del proceso.

En la Fase de Formulación se toman los casos de mejora con mayor prioridad y se planifica una primer iteración de mejora. En la Fase de Mejora se ejecuta y gestiona la planificación de mejora definida, registrando dicha ejecución, la evaluación de la mejora realizada, y si el resultado es satisfactorio se deben crear planes de aceptación e institucionalización de los nuevos procesos en la Organización. En la Fase de Revisión se evalúa el trabajo realizado, donde las lecciones aprendidas y métricas desarrolladas servirán de base para el siguiente ciclo de mejora.

Capítulo 3

Diseño y Arquitectura de Software

3.1. Introducción

En este capítulo se presenta el estado actual del conocimiento en Diseño y Arquitectura de Software, que constituye la base para el trabajo de tesis realizado. Si bien la metodología propuesta está enfocada a desarrollos basados en un solo estilo de Arquitectura de Software orientado a servicios o Service Oriented Architecture (SOA), el trabajo realizado y el propio enfoque surge a partir de los conceptos, estándares y procesos de Diseño y Arquitectura de Software que se han venido desarrollando desde los inicios de la Ingeniería de Software, y que hoy en día son ampliamente aceptados. Por la importancia que tiene para este trabajo de tesis, la presentación y desarrollo del tema Service Oriented Architecture (SOA) se realiza aparte en el Capítulo 4.

Se presentan en primer lugar conceptos y definiciones de Diseño y Arquitectura de Software que serán utilizados a lo largo de todo el trabajo en la sección 3.2. Luego en la sección 3.3 se presentan enfoques que definen actividades macro que se realizan en un proceso genérico de Diseño y Arquitectura de Software. En la sección 3.4 se presentan enfoques y metodologías para la especificación y modelado de Arquitecturas de Software. En la sección 3.5 se presenta la evaluación de Arquitecturas de Software mediante un enfoque centrado en la especificación y evaluación de atributos de calidad del software.

Parte de la sección 3.5.1 que describe el método de evaluación de Arquitecturas Architecture Tradeoff Analysis Method (ATAM), fue incluida como contexto teórico en el artículo que presenta un caso de estudio de la aplicación dicha metodología, publicado y presentado en las "VI Jornadas de Ingeniería de Software e Ingeniería del Conocimiento (JIISIC'07)" realizadas en Lima, Perú, en Febrero de 2007 [ATAM07].

3.2. Conceptos y definiciones

En esta sección se presentan conceptos y definiciones de Diseño y Arquitectura de Software, que serán utilizados a lo largo de todo el trabajo de tesis. La importancia de presentarlos radica en que el estado actual en el área está basado en el conocimiento que se ha ido generando y probando desde hace más de cuatro décadas, lo que ha permitido continuos avances en la forma de construir software.

Se presentan conceptos y definiciones que constituyen la base para el estado actual de la Disciplina Diseño y Arquitectura de Software en el área de la Ingeniería de Software. En la subsección 3.2.1 se presentan conceptos de Diseño de Software, en la subsección 3.2.2 se presentan conceptos de Arquitectura de Software, la importancia de la Arquitectura de Software en el Diseño se

presenta en la subsección 3.2.3, y finalmente en la subsección 3.2.4 se mencionan estilos/patrones de Arquitectura de Software y la importancia de su utilización.

3.2.1. Diseño de Software

El diseño de software se define en el estándar 610.12 de la IEEE [IEEE90] de 1990 como "el proceso de definición de la arquitectura, componentes, interfaces y otras características de un sistema o componente" así como "el resultado de ese proceso".

El SWEBOK [SWEB04] establece que "visto como un proceso, el diseño de software es la actividad del ciclo de vida de la Ingeniería de Software en la cual los requerimientos del software son analizados para producir una descripción de la estructura interna del software que sirve como base para su construcción. Más precisamente, un diseño de software (el resultado) debe describir la Arquitectura de Software, esto es, como el software se descompone y organiza en componentes e interfaces entre estos componentes. También debe describir los componentes a un nivel de detalle que permitan su construcción. El diseño de software juega un rol importante en el desarrollo de software: permite la producción de varios modelos que forman una especie de plano de la solución a ser implementada. Se pueden analizar y evaluar estos modelos para determinar si permitirán cumplir los requerimientos. Se pueden también examinar y evaluar varias soluciones alternativas y concesiones a realizar. Finalmente, se pueden utilizar los modelos resultantes para planificar las actividades subsecuentes de desarrollo, además de utilizarlos como entrada y punto de partida para la construcción y verificación".

En [SOMM02] Sommerville define "Un diseño de software es una descripción de la estructura del software que se va a implementar, los datos que son parte del sistema, las interfaces entre los componentes del sistema y, algunas veces, los algoritmos detallados. Los diseñadores no obtienen inmediatamente un diseño detallado, sino que lo desarrollan de manera iterativa a través de diversas versiones diferentes. El proceso de diseño incluye agregar formalidad y detalle durante el desarrollo del diseño, y regresar a los diseños anteriores para corregirlos. ... incluye el desarrollo de varios modelos del sistema con diferentes niveles de abstracción. En cuanto se descompone el sistema, se descubren los errores y omisiones de las etapas previas. Esta retroalimentación permite mejorar los modelos de los diseños previos. .... Durante todo el proceso de diseño se detalla cada vez más la especificación. El resultado final del proceso son especificaciones precisas de los algoritmos y estructuras de datos a implementarse."

De las definiciones anteriores, podemos observar que tanto el diseño como solución a los requerimientos planteados para un sistema, como el proceso de diseño para la obtención de dicha solución, son elementos complejos donde deben interactuar distintos involucrados con distintos intereses, desde el cliente, los analistas de requerimientos, los propios diseñadores hasta los implementadores y verificadores del software. El proceso de diseño además no se puede realizar en forma lineal y abarcando todos los aspectos planteados de una única vez, sino que debe ser atacado en forma progresiva para ir ganando conocimiento del problema y así evaluar distintas opciones que provean la mejor solución para el mismo, modificando y mejorando los modelos del sistema realizados.

Una parte crucial de este proceso y resultado de gran importancia del mismo, es la definición y documentación de la Arquitectura de Software del sistema, como se establece en [SWEB04], siendo dicha definición de diseño de software la que servirá de base en este trabajo. En la subsección 3.2.2 se presentan definiciones y conceptos sobre la Arquitectura de Software como parte integral del Diseño de Software, en la sección 3.3 se profundiza en el proceso de Diseño y Arquitectura de Software.

3.2.2. Arquitectura de Software

No existe una única definición de Arquitectura de Software, si bien el concepto es compartido en términos generales por la mayoría de las definiciones. En [GARL03] se plantea que hace

ya más de tres décadas, en 1974 Parnas observó que el software consiste de varias estructuras que definió como descripciones parciales que muestran el sistema como una colección de partes y relaciones entre éstas, identificando la estructura de módulos como unidades de trabajo, la estructura "uses" entre módulos como dependencia de su correctitud y la estructura de procesos como proveedores de trabajo computacional. Ya en la década de los noventa, en 1992 Perry y Wolf reconocen que "Análogamente a la arquitectura de edificios, se propone el siguiente modelo de Arquitectura de Software = {Elementos, Formas, Razonamientos/Restricciones}, esto es, una Arquitectura de Software es un conjunto de elementos arquitectónicos (o si se prefiere, de diseño) que tienen una forma particular."

La definición de Arquitectura de Software dada en [SHGA96] por Garlan y Shaw proviene de un artículo publicado previamente en 1993 y establece que "La Arquitectura de Software va más allá de los algoritmos y las estructuras de datos de computación, diseñar y especificar la estructura global del sistema emerge como un nuevo tipo de problema. Aspectos estructurales incluyen organización y estructura de control global, protocolos de comunicación, sincronización, y acceso a datos; asignación de funcionalidad a elementos de diseño; distribución física; composición de elementos de diseño; escalabilidad y performance; y selección entre alternativas de diseño."

En [BASS03] Bass, Clements y Kazman retoman la definición dada en la primer edición del libro en 1998 "La Arquitectura de Software de un programa o sistema de computación es la estructura o estructuras del sistema, que comprende elementos de software, la propiedades visibles externamente de esos elementos, y las relaciones entre estos. Por "propiedades visibles externamente" nos referimos a las asunciones que otros elementos pueden hacer de un elemento, como ser los servicios que provee, características de performance, manejo de faltas, uso de recursos compartidos, entre otros.", sustituyendo lo que antes se mencionaba como"componente" por "elemento", lo que permite mayor amplitud en la definición.

La definición dada en [USDP99] en 1999 por Jacobson, Booch y Rumbaugh establece que "Una Arquitectura es el conjunto de decisiones significativas sobre la organización de un sistema de software, la selección de elementos estructurales y sus interfaces de los cuales se compone el sistema, conjuntamente con su comportamiento el cual se especifica en las colaboraciones entre estos elementos, la composición de estos elementos estructurales y de comportamiento en subsistemas de progresivamente mayor tamaño, y el estilo de arquitectura que guía esta organización, esos elementos y sus interfaces, sus colaboraciones y su composición."

El estándar 1471 de la IEEE [IEEE00] en su edición 2000 establece que "La Arquitectura es la organización fundamental de un sistema constituida por sus componentes, las relaciones entre cada uno, y con el ambiente, y los principios que guían su diseño y evolución", aclarando que se intenta permitir una variedad de usos del término arquitectura reconociendo sus elementos comunes, principalmente la necesidad de comprender y controlar los elementos del diseño que capturan la utilidad, costo y riesgo del sistema. En algunos casos estos elementos son componentes físicos y sus relaciones, en otros son componentes lógicos, promoviendo definiciones más rigurosas sobre que constituye la organización fundamental de un sistema en cada dominio en particular.

De las definiciones anteriores se puede extraer como factor común el hecho de que la Arquitectura de Software representa un primer nivel de descomposición que muestra como se organiza el sistema en términos de elementos de software, sus interfaces y las relaciones entre éstos. Este hecho hace que la definición de la Arquitectura de Software de una aplicación revista gran importancia como parte fundamental del Diseño de Software, se profundiza en este aspecto en la subsección 3.2.3. La organización del software definida presenta "una forma" que en general está dada por la elección entre alternativas de diseño basadas en estilo/s conocidos con características definidas; en la subsección 3.2.4 se presentan algunos de los estilos clásicos en la literatura.

3.2.3. Importancia de la Arquitectura de Software en el Diseño

En [GARL03] se plantea que "sin una arquitectura apropiada para el problema que se está resolviendo y también en el caso en que se tenga definida la arquitectura pero ésta no sea bien

comprendida ni comunicada, en otras palabras bien documentada, el proyecto puede fracasar. La arquitectura presenta un plano del sistema que abarca tanto el sistema en construcción como el proyecto de desarrollo, definiendo las unidades de trabajo que deben realizar los equipos tanto de diseño como de implementación. La documentación de la arquitectura con suficiente detalle, sin ambiguedades, y organizada de forma que otros puedan encontrar rápidamente la información necesaria, es tan importante como definirla correctamente. .... Posiblemente el concepto más importante asociado a la documentación de la Arquitectura es el de "vistas", representación de un conjunto de elementos del sistema y las relaciones asociadas con ellos."

Por su parte en [BASS03] se responde específicamente a la pregunta "porqué es importante la Arquitectura de Software", "desde el punto de vista técnico, por tres razones fundamentales: comunicación entre los involucrados, decisiones tempranas de diseño y abstracción reusable y transferible del sistema". A continuación se presenta un resumen de los principales conceptos planteados en [BASS03] para cada una de las razones fundamentales identificadas:

- Como vehículo de comunicación entre los involucrados en el desarrollo, desde clientes, director de proyecto, implementadores, verificadores, entre otros, donde cada uno está interesado en distintas características del sistema afectadas por la arquitectura, y ésta provee entonces un lenguaje común donde estos intereses distintos pueden expresarse, negociarse y ser resueltos a un nivel que es intelectualmente manejable incluso para sistemas complejos y de gran tamaño.
- Representa el conjunto de decisiones tempranas de diseño, las más difíciles de tomar correctamente y las más difícil de cambiar luego en el proceso de desarrollo. Define restricciones en la implementación en cuanto a los elementos que se deben desarrollar, la forma en que deben interactuar y realizar las responsabilidades asignadas. Los atributos de calidad del sistema son permitidos o innhibidos por la Arquitectura de Software definida. Las estrategias para obtener atributos de calidad como performance, modificabilidad, seguridad, entre otras, se aplican principalmente a nivel arquitectónico. Es posible predecir este cumplimiento en base a las decisiones arquitectónicas tomadas utilizando métodos de evaluación de Arquitecturas.
- Como un modelo reusable y transferible, permite reutilizar decisiones arquitectónicas, transfiriendo por lo tanto las consecuencias de estas decisiones, lo que resulta importante por ejemplo, en el marco de líneas de productos de software, donde la Arquitectura define que aspectos están fijos para todos los miembros y cuales son variables. Permite realizar análisis "implementar, comprar, reutilizar" en base a las definiciones realizadas. La recolección de estilos/patrones arquitectónicos reconocidos permite que "menos sea más" ya que si bien los elementos de software pueden combinarse en forma ilimitada, restringir estas combinaciones a determinadas elecciones permite ganar en varios aspectos como minimizar la complejidad del diseño, promover la reutilización de software, entender y comunicar más fácilmente la Arquitectura definida, brindar mayor capacidad de análisis y acortar el tiempo de diseño.

De las reflexiones presentadas, se puede observar que la importancia de la Arquitectura de Software en el Diseño de Software va más allá de ser únicamente relativa a aspectos técnicos que si son de los más importantes. La Arquitectura de Software constituye la base de las definiciones que se realizan como parte del Diseño de Software, provee las bases para el desarrollo del sistema, identificando los elementos de software a ser construidos, sus características y relaciones, permite identificar oportunidades de reuso.

Sirve como forma de comunicación tanto en el equipo de desarrollo como con los involucrados en el mismo, incluyendo cliente y gerentes del proyecto, la utilización de estilos/patrones permite reutilizar decisiones de diseño y facilita la comprensión de la Arquitectura definida, así como su comunicación. Tener una Arquitectura de Software definida permite razonar sobre las decisiones de diseño tomadas y sus consecuencias, y evaluarlas contra los requerimientos especificados para

los atributos de calidad de forma de comprobar su cumplimiento identificando riesgos asociados a las decisiones y definiciones tomadas. En desarrollos de la complejidad y tamaño que presentan las aplicaciones actuales, contar con una Arquitectura de Software definida, documentada y comprendida por todos los involucrados constituye la base para el éxito del proyecto.

3.2.4. Estilos/patrones arquitectónicos

Un estilo arquitectónico se define en [GARL03] como "una especialización de elementos y relaciones, más un conjunto de restricciones sobre como pueden ser utilizados." Esto es definir los elementos que componen el estilo, cuales son las relaciones entre estos elementos, y que restricciones aplican al uso de elementos y relaciones entre los mismos. Adicionalmente, plantea que"ningún sistema es construido exclusivamente a partir de un único estilo, por el contrario, cada sistema puede ser visto como un amalgamiento de varios estilos diferentes".

Según [POSA96] "los patrones arquitectónicos expresan esquemas fundamentales de organización estructural para sistemas de software. Proveen un conjunto de subsistemas predefinidos, especifican sus responsabilidades, e incluyen reglas y guías para organizar las relaciones entre ellos". Definen entonces familias de sistemas en términos de patrones de organización estructural, donde los elementos, sus responsabilidades y relaciones están predefinidas. Además "están entrelazados entre sí, se pueden utilizar para refinar otros patrones de mayor alcance y se pueden combinar para resolver problemas más complejos".

El uso indistinto de los términos estilo o patrón arquitectónico para referirse al concepto presentando, está aceptado por la comunidad, como se indica en [GARL03] donde se plantea que a principios de los años 90, a la vez que en la comunidad de la Arquitectura de Software se observaban y documentaban los estilos arquitectónicos, en la comunidad de Orientación a Objetos se observaban y documentaban patrones de diseño como en "Gang of Four" [GOF95], con el mismo espíritu pero distinto alcance. Mientras los estilos arquitectónicos refieren a soluciones de diseño de mayor granularidad, los patrones de diseño refieren a soluciones de diseño específicas localizadas en uno o pocos elementos arquitectónicos de software. En [POSA96] se realiza la conjunción de ambas visiones, presentando un conjunto de estilos arquitectónicos bien conocidos documentados con la forma de patrones de diseño, esto es utilizando lenguaje de patrones.

No hay una clasificación única de estilos/patrones, distintos autores los clasifican según distintas categorías o puntos de vista, pero es posible identificar un conjunto base de estilos que se mencionan en la literatura del tema, a los que comúnmente se les refiere como estilos/patrones clásicos: Pipes&Filters, Capas, Blackboard, Invocación implícita, Cliente/Servidor y Peer-to-peer. Información detallada sobre dichos estilos/patrones y otros no mencionados, puede consultarse en [POSA96] y [GARL03]; en [POSA00] se pueden consultar también otros estilos/patrones arquitectónicos.

La importancia de utilizar estilos/patrones arquitectónicos en la definición de una Arquitectura de Software está dada principalmente por aspectos relativos a la reutilización del conocimiento, donde soluciones conocidas a problemas identificados, probadas exitosamente, se documentan para su aplicación en la resolución del problema asociado en otras instancias del mismo; otro aspecto importante es que facilita la comunicación entre los diseñadores, ya que al referirse al estilo/patrón por su nombre, todos los involucrados tienen en mente el problema y la solución que se está planteando, en cuanto a elementos de software, así como las ventajas y desventajas aparejadas a su utilización.

3.3. Procesos de Diseño y Arquitectura de Software

El proceso de definición, creación y evolución de un Diseño de Software y la Arquitectura de Software de ese diseño, involucra varias actividades y artefactos que se realizan generalmente bajo el área de diseño de software durante un proyecto. Existen estándares, guías, modelos de

proceso y de mejora de procesos, que definen actividades y entregables macro que se deben obtener en estos procesos, y enfoques más específicos asociados directamente a la obtención de la Arquitectura de Software dentro del proceso de Diseño de Software.

Es deseable que cualquiera sea el enfoque elegido se tengan en cuenta estas recomendaciones para su desarrollo, contar con buenas prácticas para el proceso de Diseño y Arquitectura de Software provee un marco de trabajo que asegura por lo menos tener en cuenta aspectos importantes en dicha tarea. En la sección 3.3.1 se presentan distintos enfoques para el proceso de Diseño de Software que definen actividades macro a realizar en el mismo, en la sección 3.3.2 se presentan enfoques para el proceso de Arquitectura de Software que definen actividades específicas para dicha tarea.

3.3.1. Procesos de Diseño de Software

Existen varios enfoques para llevar adelante el proceso de Diseño de Software en un proyecto de desarrollo, por la variedad y cantidad de enfoques existentes, se limitará la exposición a la presentación del proceso que se define en las siguientes referencias, por su importancia en el área: la norma ISO/IEC 12207 - Procesos del Ciclo de Vida del Software en la subsección 3.3.1.1, la guía IEEE Guide to the Software Engineering Body of Knowledge (SWEBOK) en la subsección 3.3.1.2 y el proceso de mejora Capability Maturity Model Integration (CMMI) en la subsección 3.3.1.3.

3.3.1.1. Norma ISO/IEC 12207 - Procesos del Ciclo de Vida del Software

La Norma ISO/IEC 12207 [ISO95] como se presentó en 2.2.2.1 establece un marco de referencia común para los procesos del ciclo de vida del software, con una terminología bien definida a la que puede hacer referencia la industria del software. Define procesos, actividades y tareas para aplicar durante todo el ciclo de vida del sistema. Las actividades de Diseño se encuentran en el proceso de Desarrollo, pero también hay de documentación y validación del Diseño en los procesos de Documentación y Validación respectivamente. En todas las actividades se especifica que hay que hacer pero no como hay que hacerlo, dejando libre la aplicación de distintos enfoques. Las actividades macro definidas en este estándar para el proceso de Diseño del Software son:

1. Diseño de la Arquitectura del Sistema: se realiza el diseño de la Arquitectura del Sistema entero teniendo en cuenta todas las partes involucradas en el mismo, tanto de software como de hardware, comunicaciones, entre otros.
2. Diseño de la Arquitectura de Software: se realiza el diseño específico de la Arquitectura del Software como primer nivel de descomposición del software en elementos y las relaciones entre estos.
3. Diseño detallado del Software: se realiza el diseño detallado de cada uno de los elementos definidos en la Arquitectura de Software y de los subsecuentes, de forma que sea posible realizar su implementación.

3.3.1.2. IEEE Software Engineering Body of Knowledge (SWEBOK)

La IEEE Guide to the Software Engineering Body of Knowledge (SWEBOK) [SWEB04] es una guía que provee una caracterización consensuada de la Ingeniería de Software y su "cuerpo de conocimiento" existente en la literatura y prácticas recomendadas para las distintas áreas de la Ingeniería de Software, caracterizando el contenido de la Disciplina como tal. Está subdividida en diez Áreas de Conocimiento, las que a su vez se subdividen en un conjunto de tópicos para los cuales se presenta información asociada y las referencias a la bibliografía donde consultar ese aspecto. Plantea que el proceso de Diseño de Software en normas y estándares del área, está

generalmente compuesto de dos actividades que encajan entre el análisis de requerimientos y la construcción del software:

1. Diseño arquitectónico de Software: algunas veces llamado diseño de alto nivel, describe como se descompone y organiza el software en componentes
2. Diseño detallado de Software: describe el comportamiento específico de cada uno de esos componentes, en forma suficiente para permitir su construcción

3.3.1.3. Capability Maturity Model Integration (CMMI)

El Capability Maturity Model Integration (CMMI) [CMMI06] como se vio en 2.4.2 es un modelo de mejora de procesos que tiene como objetivos dar soporte a la mejora de productos y procesos en una organización, brindando guías de mejores prácticas para cada proceso definido. De las Áreas de Procesos definidas, el área de diseño se encuentra en el proceso de Ingeniería dentro del Área de procesos Solución Técnica, y las prácticas específicas definidas para el proceso se pueden resumir en:

1. Seleccionar soluciones de productos componentes
2. Desarrollar el diseño
   a) Diseñar el producto o producto componente: donde el Diseño preliminar es la Arquitectura de Software y el diseño detallado es el diseño de elementos del software
   b) Establecer un paquete de datos técnicos: documentación asociada al diseño
   c) Diseñar interfaces utilizando criterios establecidos: para los elementos definidos establecer que interfaces son necesarias
   d) Realizar análisis de "implementar, comprar o reutilizar": para los componentes definidos realizar el análisis que permita identificar que soluciones de productos componentes es posible incluir en el sistema
3. Implementar el diseño del producto

De los procesos genéricos de Diseño de Software presentados, se puede observar que como pasos principales y comunes a todos se encuentran el Diseño de la Arquitectura de Software o de alto nivel, y el Diseño detallado o diseño de elementos de software definidos. Estos niveles de Diseño indican que el primer enfoque debe estar en la definición de la estructura general de la aplicación en la forma de elementos de software, sus interfaces y relaciones, como lo indican las definiciones de Arquitectura de Software presentadas en la subsección 3.2.2.

Luego el enfoque se mueve hacia el interior de los elementos definidos, para los cuales se realiza el diseño detallado de forma que estos puedan ser construidos a partir de dichas definiciones, poniendo énfasis en que se implemente el diseño definido en ambos niveles. El enfoque del CMMI presentado en 3.3.1.3 agrega tener en cuenta la posibilidad de reutilizar componentes de software existentes, dentro y fuera de la Organización de desarrollo.

3.3.2. Procesos de Arquitectura de Software

Existen también varios enfoques y metodologías para realizar el proceso y la definición de la Arquitectura de Software. Por su importancia en el área y sus distintos enfoques, se seleccionaron para presentar en esta sección: el enfoque del Rational Unified Process (RUP) basado en definición de la Arquitectura a partir de la identificación de Casos de Uso relevantes a la Arquitectura, en la subsección 3.3.2.1 y el enfoque del Architecture Business Cycle (ABC) basado en la definición de la Arquitectura a partir de los escenarios de atributos de calidad, en la subsección 3.3.2.2.

3.3.2.1. Rational Unified Process (RUP)

El Rational Unified Process (RUP) como se describió en 3.3.2.1 tiene como una de sus características principales que es centrado en la Arquitectura. Esto significa que en el proceso se hace especial énfasis en las actividades y entregables asociados a la definición y documentación de la Arquitectura de Software en el desarrollo, planteando como objetivo principal de la Fase de Elaboración la obtención del ejecutable de la Línea Base de la Arquitectura, que como su nombre lo indica, es un ejecutable que implementa las principales definiciones de diseño realizadas en la creación de la Arquitectura. Las actividades y entregables asociados a la Arquitectura de Software se encuentran definidos en la Disciplina de Análisis&Diseño y prescriben:

1. Realizar síntesis arquitectónica: en la Fase Inicial en forma opcional, se realiza el análisis arquitectónico a partir del Modelo de Casos de Uso identificando Casos de Uso claves o relevantes a la Arquitectura, requerimientos suplementarios, arquitecturas de referencia (estilos/patrones) y restricciones del proyecto, para definir una Arquitectura preliminar y construir y evaluar si aplica, una Prueba de Concepto de la Arquitectura (Architectural-proof-of-concept - APOC) que demuestre que el sistema es factible con la solución planteada

2. Definir una Arquitectura Candidata: en la Fase de Elaboración temprana, se realiza el análisis arquitectónico a partir de los Casos de Uso identificados como relevantes a la Arquitectura, los Modelos de Casos de Uso y de Diseño (si existe una primera definición), requerimientos suplementarios, arquitecturas de referencia (estilos/patrones) y restricciones del proyecto, para definir la Arquitectura de Software inicial, y realizar análisis de los Casos de Uso especificados

3. Analizar comportamiento: como actividades más importantes plantea el análisis de Casos de Uso y la identificación de elementos de diseño a partir del Documento de la Arquitectura (SAD), modelos de Casos de Uso y Diseño, requerimientos suplementarios y restricciones del diseño, para transformar las descripciones de comportamiento especificadas en los requerimientos en un conjunto de elementos en los cuales basar el Diseño. Incluye revisar el diseño realizado.

4. Diseñar componentes: plantea el diseño específico de Casos de Uso, subsistemas y clases de diseño, en base a lo diseñado por el Arquitecto de Software en las actividades anteriores, quien en esta actividad no participa ya que se puede ver como el diseño detallado de elementos definidos.

5. Diseñar la Base de Datos: plantea realizar en forma opcional, el diseño de la Base de Datos según las definiciones realizadas, realizando el mapeo entre el paradigma de Orientación a Objetos utilizado en el diseño y el paradigma Relacional para la persistencia, si aplica.

6. Refinar la Arquitectura: plantea identificar elementos y mecanismos de diseño, así como elementos de diseño existentes a incorporar, estructurando el modelo de Implementación y definiendo la Arquitectura de ejecución y deployment, para completar la Arquitectura para una iteración. Se incluye también la actividad Revisar la Arquitectura que tiene como entrada el Documento de la Arquitectura (SAD) generado, los requerimientos suplementarios, las restricciones del proyecto y una lista de riesgos.

La metodología propuesta se basa en la identificación de Casos de Uso o escenarios relevantes a la Arquitectura, que son aquellos definidos en [RUP] como los que "representan alguna funcionalidad significativa o central del sistema final, o tienen una gran cobertura arquitectónica - ejercitan muchos elementos arquitectónicos, o estresan o ilustran un punto específico o delicado de la Arquitectura".

Para la identificación de los Casos de Uso relevantes a la Arquitectura se plantea en [RUP] que se deben tener en cuenta los siguientes factores claves: "el beneficio del escenario para los

clientes: crítico, importante, útil; el impacto del escenario en la Arquitectura: ninguno, extiende, modifica; ... los riesgos a ser mitigados (performance, disponibilidad de un producto, adecuación de un componente); la cobertura de la arquitectura; ... otros objetivos tácticos o restricciones: demostración al usuario final, entre otros."

Los Casos de Uso relevantes a la Arquitectura, definen en mi interpretación, clases de equivalencia de comportamiento, impacto en la Arquitectura, riesgos que mitigan y/o importancia para el usuario, o sea, los aspectos por los cuales son seleccionados, ya que la elección de los mismos o de algún escenario que contienen, implica la identificación de alguno de estos aspectos, y por lo tanto la definición de la solución para el mismo.

Como se aclara en [RUP] "pueden haber dos escenarios que impactan en los mismos componentes y tratan riesgos similares. Si se implementa A primero, entonces B no es significativo arquitectónicamente. Si se implementa B primero, entonces A no es significativo arquitectónicamente.", por lo que se entiende que cada Caso de Uso o escenario elegido, será "representante" de los aspectos por los cuales fue elegido, y cualquier otro que presente aspectos ya tenidos en cuenta por alguno de los seleccionados, no agrega información y por lo tanto la solución al mismo estará incluida en la solución definida para el Caso de Uso o escenario identificado como relevante por esos aspectos.

El proceso de definición de la Arquitectura en [RUP] se realiza también en forma iterativa incremental, comenzando con la identificación de los Casos de Uso relevantes según lo mencionado. Estos guían la Arquitectura que influencia a su vez los Casos de Uso definidos. En la Fase inicial se identifican los Casos de Uso relevantes y se documentan en el Documento de la Arquitectura (SAD) en la vista de Casos de Uso, y a partir de estos se trabaja con la Arquitectura preliminar, definiendo las vistas Lógica y de Deployment, si aplica. Es posible construir un prototipo de la Arquitectura (Architectural proof-of-concept - APOC) que permita explorar riesgos y requerimientos claves.

Entre los objetivos a cumplir en la Fase inicial están comprender y delimitar el problema, y exhibir una arquitectura inicial para la solución, definiendo la implementación de los primeros Casos de Uso para la Fase de Elaboración. En la Fase de Elaboración temprana se realiza la definición de una Arquitectura candidata en la cual: se itera sobre la Arquitectura preliminar (si existe) evaluando lo realizado y modificando/agregando elementos en las vistas Lógica, de Procesos, de Deployment y de Implementación, revisando los Casos de Uso relevantes y sus realizaciones; se realiza el análisis de Casos de Uso relevantes para identificar las interacciones necesarias entre los elementos de software definidos para cumplir estos Casos de Uso.

El objetivo principal de la Fase de Elaboración es obtener el ejecutable de la Línea Base de la Arquitectura, que demuestra la factibilidad de la solución, mitiga los riesgos y desafíos más importantes y constituye la base del desarrollo en la Fase de Construcción. En la Fase de Construcción entonces se construye sobre el ejecutable de la Línea Base de la Arquitectura obtenido en Elaboración. Puede ser que se hayan tenido que incorporar elementos al ejecutable que no son arquitectónicos pero si necesarios para su ejecución. Para poder paralelizar al máximo el desarrollo la Arquitectura debe estar estable, esto es, pueden ocurrir cambios menores pero no cambios importantes. Si ocurren cambios importantes se demuestra la inestabilidad de la Arquitectura y que no fueron tenidos en cuenta todos los aspectos requeridos en su definición.

3.3.2.2. Architecture Business Cycle (ABC)

El Architecture Business Cycle (ABC) planteado por el Software Engineering Institute (SEI) se define en [BASS03] como "el ciclo de influencias desde el ambiente a la Arquitectura y desde la Arquitectura de vuelta hacia el ambiente", bajo la hipótesis de que "la Arquitectura de Software es el resultado de influencias técnicas, del negocio y sociales, que luego se ven afectados nuevamente por la Arquitectura definida y a su vez influencian nuevamente la definición de futuras Arquitecturas".

Entre las influencias hacia la Arquitectura se destacan los involucrados en el sistema, la Organización en que se realiza el desarrollo, la experiencia y conocimiento acumulado de los Arquitectos y el ambiente técnico. Entre las influencias desde la Arquitectura se plantea que afecta la estructura de la Organización en que se realiza el desarrollo, los objetivos de la Organización, los requerimientos de los clientes para el próximo sistema a desarrollar, la experiencia del Arquitecto y en algunos casos incluso influenciar y cambiar la cultura de la Ingeniería de Software y el ambiente técnico de desarrollo. El ABC prescribe las siguientes actividades para el proceso de la Arquitectura de Software:

1. Crear el caso del negocio para el sistema: si aplica, definir el mercado objetivo, los costos y precio de venta asociados, el time-to-market. Es importante que el Arquitecto participe en esta actividad conjuntamente con los involucrados en el área del negocio.
2. Comprender los requerimientos: se propone el uso de escenarios de atributos de calidad para capturar y comprender los requerimientos de atributos de calidad para el sistema, a partir de los cuales se definirá la Arquitectura.
3. Crear o seleccionar la Arquitectura: se propone una metodología el Atribute-Driven Design (ADD) basada en los escenarios de atributos de calidad definidos, para crear una Arquitectura que cumpla con los requerimientos de calidad y comportamiento especificados.
4. Comunicar la Arquitectura: debe comunicarse en forma clara y sin ambiguedades a todos los involucrados en el desarrollo, se propone la documentación basada en vistas a elección, con siete secciones principales a documentar para cada vista y otras secciones agregadas "más allá de las vistas" según [GARL03].
5. Analizar o evaluar la Arquitectura: para asegurar que el sistema que se construya a partir de la Arquitectura satisface las necesidades de los clientes. Se propone el método Architecture Tradeoff Analysis Method (ATAM) basado también en escenarios de atributos de calidad definidos para evaluar el cumplimiento o no de los mismos en la Arquitectura definida.
6. Implementar en base a la Arquitectura: asegurarse que los desarrolladores comprenden la Arquitectura definida y mantienen la implementación fiel a las definiciones realizadas, como forma de asegurar la construcción conforme a la misma.
7. Asegurar conformidad con la Arquitectura: en la etapa de mantenimiento se debe asegurar que la Arquitectura actual y su representación se mantienen fieles entre sí.

Para el paso 3 donde se realiza la definición de la Arquitectura, se propone la metodología Atribute-Driven Design (ADD) para crearla, que basa el proceso de descomposición del software en los atributos de calidad que debe cumplir. Un atributo de calidad refiere a requerimientos de disponibilidad, modificabilidad, performance, seguridad, verificabilidad y usabilidad, entre otros. La entrada del método son los Escenarios de Calidad que expresan las definiciones conductoras de la arquitectura (architectural drivers), usando la relación entre alcanzar los atributos de calidad y la arquitectura, para diseñarla.

Un escenario describe la interacción de un usuario con el sistema, existiendo tres tipos: de Casos de Uso (usos del sistema), de crecimiento (anticipación al cambio) y de exploración (cambios extremos en el sistema). El proceso de descomposición sigue un enfoque top-down, descomponiendo en forma recursiva el sistema donde en cada paso se utilizan tácticas y patrones arquitectónicos para satisfacer los escenarios de calidad, asignando funcionalidad en los módulos definidos. A continuación se presentan los pasos generales del método Atribute-Driven Design (ADD):

1. Elegir el módulo a descomponer: el primero al comenzar es el sistema entero, con información de requerimientos funcionales y atributos de calidad para el sistema y/o módulo
2. Refinar el módulo de acuerdo a los pasos siguientes
   a) Seleccionar las definiciones conductoras de la arquitectura (architectural drivers) del conjunto de requerimientos funcionales y escenarios de calidad. Este paso determina que es importante para esta descomposición.
   b) Elegir un patrón/estilo de arquitectura que satisfaga las definiciones conductoras de la arquitectura para esta descomposición, identificando los módulos necesarios para cumplir con dichas definiciones
   c) Instanciar módulos y asignar funcionalidad a partir de los requerimientos, representándolos con múltiples vistas
   d) Definir interfaces de los módulos teniendo en cuenta las restricciones de interacción entre éstos
   e) Verificar y refinar los escenarios de calidad tomándolos como restricciones para los módulos definidos
3. Repetir los pasos anteriores para cada módulo que debe descomponerse

Una vez que se tiene definido lo suficiente, se asignan los módulos a los equipos de desarrollo (o personas), y se implementa un "esqueleto" del sistema que muestre ejecución e interacción entre elementos arquitectónicos. Como se plantea en [BASS03] "el método Atribute-Driven Design (ADD) puede ser visto como una extensión para la mayoría de los otros métodos de desarrollo, como el Rational Unified Process (RUP). El RUP tiene varios pasos que resultan en el diseño de alto nivel de la Arquitectura pero luego prosigue al diseño detallado e implementación. Incorporar el ADD implica modificar los pasos asociados con el diseño de alto nivel de la Arquitectura y luego continuar con el proceso que prescribe RUP."

3.4. Documentación de Arquitecturas de Software

Según [GARL03] una especificación de arquitectura en general connota la utilización de un lenguaje formal. La representación de una arquitectura "connota un modelo, una abstracción, una representación de una cosa que es separada o diferente de la cosa en si misma", representación de la realidad como un modelo que la representa. La descripción de una arquitectura se puede realizar de forma más o menos formal utilizando: Architecture Description Languages (ADLs) enfoque más formal pero restringido a expertos en el tema, Unified Modeling Language (UML) enfoque menos formal que el anterior, pero ampliamente aceptado por la comunidad. La documentación de una arquitectura refiere a la "creación de un artefacto, llamese un documento ... documentar una Arquitectura de Software se vuelve una tarea concreta: producir un documento de Arquitectura de Software (SAD)".

Sobre la utilización de "vistas" para la documentación de la Arquitectura se plantea en [GARL03] "Posiblemente el concepto más importante asociado con la documentación de la Arquitectura de Software es el de *vista*. Una Arquitectura de Software es una entidad compleja que no puede ser descrita de forma simple y unidimensional. Nuestra analogía con el ala de un pájaro lo prueba iluminando el hecho. No hay una única forma de describir el ala de un pájaro. Por el contrario, hay muchas: vistas de las plumas, esqueleto, circulación, músculos y muchas otras. ¿Cual de estas vistas *es* la "arquitectura" del ala ? Ninguna de ellas. ¿Qué vistas *constituyen* la arquitectura? Todas."

En general está ampliamente aceptado en la comunidad el hecho de documentar la Arquitectura de Software utilizando diferentes vistas que muestran los elementos involucrados en la misma

según esa vista o punto de vista. Las definiciones de vista, punto de vista, y elementos asociados se presentará para cada enfoque en la subsección correspondiente.

Existen varios enfoques para describir una Arquitectura en base a vistas, de los que se presentarán tres seleccionados por su importancia como referencia en la comunidad: en la subsección 3.4.1 el marco general para descripción de Arquitecturas de Software definido en el estándar IEEE Std 1471 - IEEE Recommended Practice for Architectural Description of Software-Intensive Systems, en la subsección 3.4.2 el "Modelo de vistas "4+1" de la Arquitectura de Software" de Kruchten [KRUC95] incluido en el enfoque del Rational Unified Process (RUP), y en la subsección 3.4.3 el enfoque de documentación de vistas de la Arquitectura de Software definido por el Software Engineering Institute (SEI).

Como otras propuestas existentes que no se presentan se pueden mencionar el "Siemens Four View model for Architecture" [SIEM95], definido a partir de la observación en la práctica en la industria, y el estándar "ISO/IEC 10746-3, ITU-T X.903 - Arquitectura - Reference Model for Open Distributed Processing (RM-ODP) [RMOD96], modelo de referencia de procesamiento abierto y distribuido para sistemas de integración empresarial.

3.4.1. IEEE Std 1471 - Práctica recomendada para Descripción de la Arquitectura de Sistemas de Software intensivo

El IEEE Std 1471 - IEEE Recommended Practice for Architectural Description of Software-Intensive System [IEEE00], como recomendación de prácticas "trata la descripción de Arquitecturas en sistemas de software intensivo. Un sistema de software-intensivo es cualquier sistema donde el software contribuye con influencias esenciales para el diseño, construcción, deployment y evolución del sistema como un todo. ... El propósito de estas prácticas recomendadas es facilitar la expresión y comunicación de Arquitecturas y establecer las bases para ganancias en calidad y costos a través de la estandarización de elementos y prácticas para la descripción de Arquitecturas."

Presenta un "marco de referencia, para descripción de Arquitecturas. El framework establece términos y conceptos asociados con el contenido y uso de descripciones arquitectónicas." El enfoque propuesto para la descripción de la Arquitectura es conforme con el estándar IEEE/EIA Std 12207.0-1996 - IEEE/EIA Standard - Industry Implementation of ISO/IEC 12207: 1995, Information Technology - Software life cycle processes y está basado entre otros en [BASS03] en su edición 1998, [UML] en su versión 1.1 de 1997 y [SHGA93].

El framework conceptual del estándar define que "una descripción arquitectónica está organizada en uno o más elementos llamados vistas (arquitectónicas). Cada vista trata uno o más de los intereses de los involucrados en el sistema. El término vista se utiliza para referir a la expresión de la arquitectura de un sistema con respecto a un punto de vista particular. ... Un punto de vista establece las convenciones por las cuales una vista es creada, descrita y analizada.

De esta forma, una vista es conforme a un punto de vista. El punto de vista determina los lenguajes (incluyendo notaciones, modelos o tipos de productos) a ser utilizados para describir la vista, y cualquier método de modelado asociado o técnicas de análisis a ser aplicadas a estas representaciones de la vista. Estos lenguajes y técnicas son utilizados para alcanzar resultados relevantes a los intereses tratados por el punto de vista." En la figura 3.1 se muestran los elementos de modelado del framework.

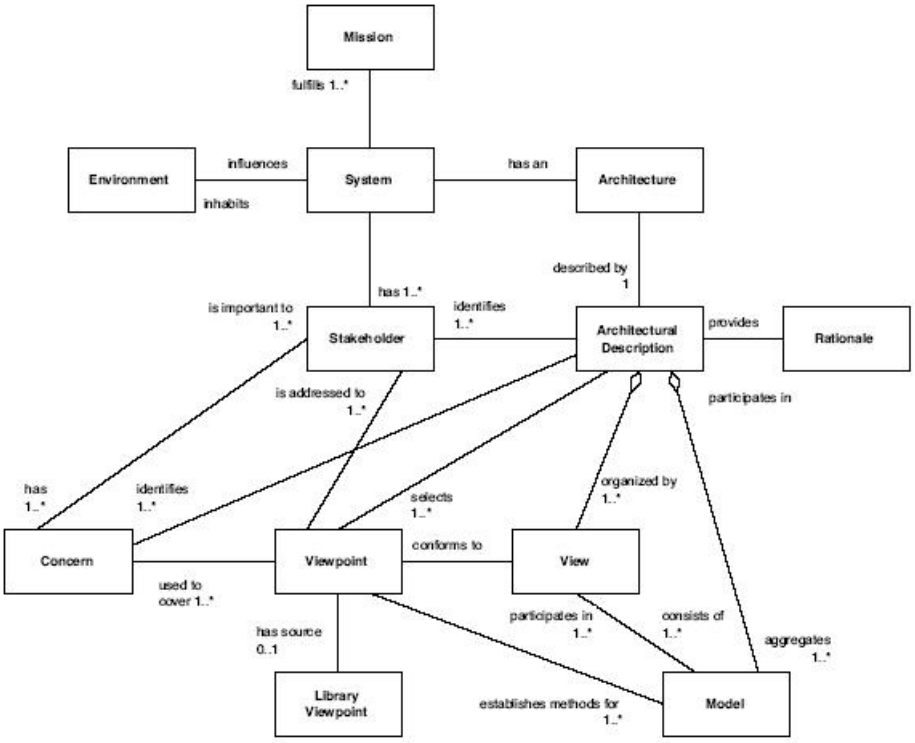


Figura 3.1: Elementos de modelado del framework conceptual en el IEEE Std. 1471 para descripción de Arquitecturas de [IEEE00]

El framework conceptual se presenta en cuatro secciones que contienen la siguiente información: la descripción arquitectónica en contexto, los involucrados y sus roles, las actividades arquitectónicas en el ciclo de vida (en cuanto a la descripción arquitectónica) y los usos de la descripción arquitectónica. Luego de la presentación del framework se presenta la sección de prácticas para la descripción de la Arquitectura, que contiene seis secciones en las que se indica que información se debe incluir en dicha descripción:

1. Documentación de la Arquitectura: incluye información básica de documentación sobre contexto, alcance, referencias, entre otras.
2. Identificación de involucrados y sus roles: se identifican los usuarios e intereses relevantes a la Arquitectura, por ejemplo el equipo de desarrollo y el cliente, entre otros, y sus intereses como por ejemplo el comportamiento de entrada/salida del sistema, el propósito del sistema y su factibilidad, entre otros.
3. Selección de puntos de vista arquitectónicos: requiere que conjunto de puntos de vista sea seleccionado, definido y analizada su cobertura, indicando el razonamiento para su elección. Para cada punto de vista seleccionado se incluye el nombre, involucrados a los que se dirige, los intereses que trata, lenguaje, técnicas de modelado o métodos analíticos para construir una vista basada en ese punto de vista, la fuente del punto de vista incluyendo autor, referencia a la documentación, entre otros.
4. Vistas arquitectónicas: requiere la representación de la Arquitectura resultante mediante la selección de una o más vistas. Cada una debe corresponder con un punto de vista de los seleccionados antes y ser conforme con lo especificado para el mismo. Se incluye un identificador e información introductoria de la vista, representación del sistema construida con lenguajes, métodos y técnicas de modelado o análisis especificadas para el punto de vista asociado, información de configuración.

5. Consistencia entre vistas arquitectónicas: presenta análisis de consistencia para todas las vistas incluidas en la descripción, y el registro de inconsistencias entre vistas si las hubiera.

6. Razonamiento arquitectónico: incluye el razonamiento arquitectónico para los conceptos seleccionados, así como evidencia de la consideración de conceptos arquitectónicos alternativos y el razonamiento para las elecciones realizadas.

Es importante notar que en el estándar no se prescribe el uso de determinadas vistas para la descripción de la Arquitectura, sino que se indica que información se debe incluir en dicha descripción en cuanto a la selección de puntos de vista y vistas conforme a los mismos, y que información incluir en cada sección. Otro aspecto importante del estándar es la inclusión de una sección de razonamiento arquitectónico, en la cual registrar las decisiones más importantes tomadas en la definición de la Arquitectura, proveyendo evidencia de la consideración de conceptos arquitectónicos alternativos y el razonamiento para las elecciones tomadas. Este es un registro importante para que otras personas puedan entender el porqué de las decisiones de diseño tomadas.

3.4.2. Modelo de vistas "4+1" de la Arquitectura de Software del RUP

El "Modelo de vistas "4+1" de la Arquitectura de Software" de Kruchten [KRUC95] fue publicado y tiene entre sus referencias [SHGA93]. Como se establece en [RUP] la "Arquitectura es representada por un número de distintas vistas arquitectónicas, las que en su esencia son extractos ilustrando los elementos "significativos arquitectónicamente" de los modelos. En el RUP se comienza con un conjunto típico de vistas, llamado el "Modelo de vistas "4+1" de la Arquitectura de Software" de Kruchten [KRUC95], que se compone de:

1. Vista de Casos de Uso: contiene los Casos de Uso y escenarios que contienen comportamiento significativo arquitectónicamente, clases, o riesgos técnicos. Es un subconjunto del Modelo de Casos de Uso.

2. Vista Lógica: contiene las clases de diseño más importantes y su organización en paquetes y subsistemas, y la organización de estos paquetes y subsistemas en capas. Contiene también algunas realizaciones de Casos de Uso. Es un subconjunto del Modelo de Diseño.

3. Vista de Procesos: contiene la descripción de tareas (procesos e hilos) involucrados, sus interacciones y configuraciones, y la asignación de objetos de diseño y clases a tareas. Esta vista solo es necesaria si el sistema tiene un grado significativo de concurrencia. Es un subconjunto del Modelo de Diseño.

4. Vista de Implementación: contiene una visión global del modelo de Implementación y su organización en términos de módulos en paquetes y capas. La asignación de paquetes y clases (de la vista lógica) en paquetes y módulos de la vista de Implementación también se describe. Es un subconjunto del Modelo de Implementación.

5. Vista de Deployment: contiene la descripción de los varios nodos físicos para las configuraciones más típicas de plataformas, y la asignación de tareas (de la Vista de Procesos) a los nodos físicos. Solo es necesario utilizar esta vista si el sistema es distribuido. Es un subconjunto del Modelo de Deployment.

Las vistas arquitectónicas se documentan en el Documento de la Arquitectura de Software (SAD). Se pueden agregar vistas adicionales para expresar distintos intereses especiales: vista de interface de usuario, vista de seguridad, vista de datos, y otras. Para sistemas simples, se pueden omitir algunas de las vistas contenidas en el "Modelo de Vistas "4+1" de la Arquitectura de Software" presentado. ... Por lo tanto, en esencia, las vistas arquitectónicas pueden ser definidas

como abstracciones o simplificaciones de los modelos construidos, en las que se hacen visibles las características más importantes dejando los detalles de lado.“ En la figura 3.2 se muestra graficamente la definición y relación entre las vistas definidas en el modelo.

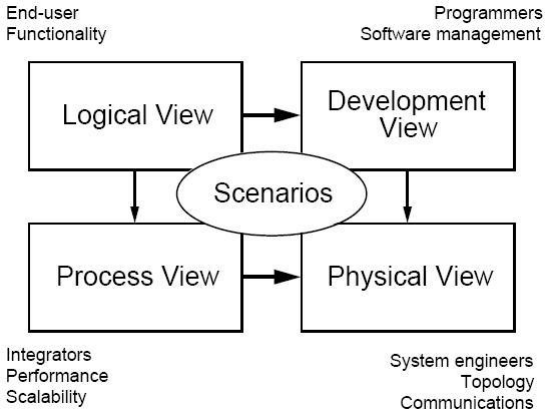


Figura 3.2: Vistas definidas y su relación en el “Modelo de vistas “4+1” de la Arquitectura de Software” de [KRUC95]

En la plantilla de Documento de la Arquitectura (SAD) que propone el RUP [RUP] se agregan además las siguientes vistas:

- Vista de Datos: Presenta la definición de tablas, vistas, índices, triggers y procedimientos almacenados necesarios para proveer la persistencia del sistema, indicando el mapeo de clases en la vista lógica a tablas.
- Tamaño y Performance: Características del sistema asociadas al volumen y tiempos de respuesta que impactan en la Arquitectura: cantidad de usuarios concurrentes, espacio en memoria y disco requeridos, entre otros.
- Calidad: Atributos de calidad requeridos para el sistema, como portabilidad, extensibilidad, modificabilidad, seguridad, tiempo medio entre caidas, entre otros.

La documentación de las distintas vistas en el Documento de la Arquitectura (SAD) se realiza siguiendo la metodología de definición de la Arquitectura establecida por el RUP [RUP] que fue presentada en la subsección 3.3.2.1. El conjunto de cinco vistas básico en general es obligatorio para todos los desarrollos que se realicen siguiendo el Rational Unified Process (RUP) a menos de las excepciones mencionadas (deployment si no hay distribución por ejemplo), esto es, si en un proyecto se define que se seguirá como proceso de desarrollo el RUP, entonces el Documento de la Arquitectura (SAD) debe tener las vistas indicadas con la información mencionada, obtenida siguiendo la metodología planteada.

Un aspecto importante que se considera faltante en la plantilla que presenta el RUP para el SAD, es una sección en la cual presentar información sobre el razonamiento realizado en la definición de la Arquitectura, documentando las alternativas evaluadas y los porqué de las decisiones tomadas y las opciones rechazadas también, como se indica en el estándar [IEEE00]. Si bien como se indica, es posible agregar secciones en el SAD, sería importante contar con esta sección como parte obligatoria de la documentación de la Arquitectura en el RUP.

3.4.3. Documentación de vistas de la Arquitectura de Software del SEI

El SEI (Software Engineering Institute) provee una plantilla para el Documento de la Arquitectura (SAD) basado en las recomendaciones del IEEE Std 1471-2000 [IEEE00] y en el trabajo del

Software Engineering Institute (SEI) en [GARL03], [BASS03], [ATAM04] en su edición 2001 y [QAW03]. Incorpora la identificación de atributos de calidad para el sistema y escenarios que los involucran, que serán luego utilizados para evaluar la Arquitectura en la metodología del Software Engineering Institute (SEI) Architecture Tradeoff Analysis Method (ATAM) [ATAM04]que se presenta en la sección 3.5.

En [GARL03] una vista se define como "una representación de un conjunto de elementos del sistema y las relaciones asociadas con ellos", un punto de vista "define los tipos de elementos y relaciones utilizados para describir la Arquitectura de un sistema de Software desde una perspectiva particular". Se define también un paquete de una vista como "el conjunto más pequeño cohesivo de documentación que se le puede entregar a un involucrado en el desarrollo, como ser el equipo de desarrollo o un subcontratista", luego "una vista consiste en un conjunto de paquetes de la vista que están relacionados mediante relaciones hermano y padre/hijo". Se presta especial atención a la documentación de interfaces, las que se definen como "una frontera a través de la cual dos entidades independientes se encuentran e interactúan o comunican una con otra".

Como parte principal de la plantilla se definen las secciones para la documentación de vistas mediante paquetes de vistas e interfaces. En la plantilla del SAD del SEI [SADS04], en [BASS03] y en [GARL03] se presenta la información que se debe incluir para cada sección. En la sección 1 se incluyen los puntos de vista seleccionados, con la información que se prescribe en [IEEE00], luego en la sección 2 se presenta el contexto de la Arquitectura incluyendo el contexto del problema y el contexto de la solución. En la sección 3 se presentan las vistas seleccionadas, donde para cada vista se incluye información para identificar la vista, descripción, contexto de la Arquitectura para esa vista particular, entre otros, y finalmente la información sobre los paquetes de vista que la integran, donde según [GARL03] para cada paquete de vista se deben incluir las siguientes secciones:

1. presentación primaria: se muestran los elementos y relaciones entre éstos que se incluyen en la porción de la vista mostrada en ese paquete, utilizando lenguajes y notaciones adecuadas.
2. catálogo de elementos: se detallan los elementos presentados brindando información adicional para completar la descripción:
   a) elementos y sus propiedades: se describe cada elemento identificando sus responsabilidades y valores para las propiedades relevantes según lo indicado en el punto de vista que se conforma.
   b) relaciones y sus propiedades: se describen las relaciones entre elementos, sus especializaciones o restricciones.
   c) interfaces de los elementos: se especifican las interfaces de los elementos definidos que son visibles a otros elementos.
   d) comportamiento de los elementos: se describe el comportamiento de los elementos o grupos de elementos relacionados.
3. diagrama de contexto: se presenta un diagrama de contexto de la parte del sistema que se describe en este paquete de vista, mostrando el alcance y las interacciones con entidades externas.
4. guía de variabilidad: se presentan variabilidades disponibles en la parte del sistema que se describe en este paquete de vista,
5. contexto de la Arquitectura: se presenta el razonamiento para las decisiones tomadas para la parte del sistema que describe este paquete de vista, incluyendo:
   a) razonamiento: se describe el razonamiento realizado para las decisiones de diseño tomadas, mostrando la lista de alternativas evaluadas y razones para rechazarlas.

   b) resultados del análisis: se describe el resultado de análisis realizados, por ejemplo de performance o seguridad.
   c) suposiciones: se describen las suposiciones tomadas, por ejemplo relativas al entorno o a las necesidades planteadas.

6. otra información: se incluye cualquier otra información que no esté incluida en las secciones anteriores.

7. paquetes de vista relacionados: se presentan los paquetes de vista hermanos, padres o hijos relacionados con este.

Es importante destacar que en la sección 2 del documento general en la cual se presenta el contexto de la Arquitectura para todo el sistema, en la descripción del contexto del problema se incluyen las líneas conductoras de la Arquitectura (architectural drivers) identificadas, donde según [SADS04] se "describen los requerimientos de comportamiento y de atributos de calidad que le dan forma a la arquitectura. Se incluyen escenarios que expresan objetivos de atributos de calidad y guías de comportamiento, como se indica en [QAW03] y [ATAM04]" y en la descripción del contexto de la solución se incluyen los enfoques arquitectónicos utilizados.

En la descripción del contexto de la solución, según [SADS04] se "provee el razonamiento para las principales decisiones de diseño contenidas en la Arquitectura de Software. Describe los enfoques de diseño aplicados a la Arquitectura de Software, incluyendo el uso de estilos arquitectónicos o patrones de diseño, cuando el alcance de dichos enfoques trasciende a una única vista. Esta sección provee también el razonamiento para la selección de esos enfoques. También describe cualquier alternativa significativa que fue seriamente considerada y las razones por las que fue finalmente descartada. Esta sección describe cualquier aspecto relacionado con el uso de COTS, incluyendo estudios asociados."

Sin embargo, la selección de puntos de vista en primer lugar, luego la selección de vistas que conforman con los puntos de vista seleccionados, y finalmente los paquetes de vista que componen cada vista definida, pueden hacer que la documentación sea compleja de realizar y de comprender, al existir tantos niveles de indirección entre las distintas secciones. Es clara sin embargo la información que se debe incluir en cada sección y subsección, y es posible adaptar la plantilla a las necesidades de cada Organización. Incluso en [BASS03] se hace una simplificación de los niveles definidos, presentando la información que corresponde a los paquetes de vista directamente como información de las vistas, esto es, asumiendo para cada vista la existencia de un único paquete de vista que es la vista misma. Esto hace bastante más comprensible, a mi entender, el uso de la plantilla y simplifica la información a incluir en la misma. Se observa también que esta plantilla de SAD se destaca por ser la más formal en su definición y en los conceptos e información manejados, así como en su presentación.

3.5. Métodos de evaluación de Arquitecturas de Software

Los métodos de evaluación de arquitecturas son bastante recientes, surgen de la mano de la popularización de la Arquitectura de Software como área de investigación y práctica dentro de la Ingeniería de Software, como se presentó en la sección 3.2. La evaluación de arquitecturas permite analizar las decisiones tomadas y ver si cumplen con los requerimientos definidos para el sistema, y si es esperable que la Arquitectura definida se comporte como es esperado con respecto principalmente a los atributos de calidad identificados.

El Software Engineering Institute (SEI) [SEI] propone distintos enfoques siendo el más importante el método de evaluación de Arquitecturas Architecture Tradeoff Analysis Method (ATAM) que se presenta en la subsección 3.5.1. El Quality Atributte Workshop (QAW) [QAW03] se utiliza generalmente en forma complementaria al ATAM para apoyar la especificación de escenarios de atributos de calidad en forma previa a la utilización de ATAM, pero no será presentado en este trabajo.

3.5.1. Architecture Tradeoff Analysis Method (ATAM)

En esta sección se describen los principales conceptos y la metodología del Architecture Tradeoff Analysis Method (ATAM) según [ATAM04]. Un caso de estudio de la aplicación de la metodología de ATAM se puede ver en [ATAM07].

Según [ATAM04] el Architecture Tradeoff Analysis Method (ATAM) es una metodología para evaluar Arquitecturas de Software que principalmente evalúa la adecuación de la Arquitectura de Software definida con respecto a los atributos de calidad especificados para el sistema. Surge confluyendo ideas y técnicas de tres áreas: la noción de estilos o patrones de arquitectura, el análisis de atributos de calidad y el método Software Architecture Analysis Method (SAAM) [SAAM] que es el predecesor del ATAM en el Software Engineering Institute (SEI). Obtiene su nombre del concepto de que a veces no será posible cumplir con todos los atributos de calidad definidos y se deberán realizar concesiones mutuas (o tradeoffs) entre estos, para obtener el balance adecuado.

En el marco de ATAM se toma como definición de arquitectura la realizada en [BASS03], considerando que la Arquitectura de Software condiciona las características del producto final en cuanto a cualidades o atributos de calidad como performance y mantenibilidad, resulta importante evaluar el cumplimiento de los mismos en forma temprana para corregir errores antes de pasar a la codificación del sistema, donde es más costoso. Sin embargo, evaluaciones a posteriori resultan útiles como aprendizaje y estudio de posibilidades de mejora, por ejemplo para desarrollar una nueva versión o entender un sistema legado. ATAM se basa en la especificación de requerimientos de atributos de calidad para el sistema como escenarios de calidad, y propone una metodología con Fases, actividades y entregables para su realización, así como especificación de roles y responsabilidades asociadas.

Conceptos del método

Para evaluar un diseño arquitectónico contra los requerimientos de atributos de calidad es necesario contar con una caracterización de los atributos de calidad. Por ejemplo para entender una arquitectura desde el punto de vista de la modificabilidad requiere entender como observar y medir este atributo, así como entender de que forma impactan en este atributo los distintos tipos de decisiones arquitectónicas.

ATAM brinda una caracterización para varios atributos creada a partir del conocimiento existente en las comunidades de atributos de calidad, principalmente para atributos como performance, modificabilidad, disponibilidad, usabilidad y seguridad. Cada caracterización de atributos de calidad se divide en tres categorías: estímulo externo, decisiones arquitectónicas y respuestas, donde los primeros son eventos que hacen que la arquitectura responda o cambie, las decisiones arquitectónicas son los elementos de la arquitectura (componentes, conectores y sus propiedades) que tienen impacto directo en la obtención de respuestas a los atributos, y las respuestas se caracterizan por atributos medibles.

El mecanismo utilizado para obtener esta información es el Escenario. Un escenario es una especificación corta que describe la interacción entre un stakeholder y el sistema, siendo la definición de stakeholder cualquier involucrado en el desarrollo, tanto cliente, como desarrolladores como gerentes, entre otros. Los escenarios proveen una forma de concretizar cualidades a tener en cuenta en tiempo de desarrollo como la modificabilidad, ayudan a entender cualidades de tiempo de ejecución como performance o disponibilidad, entre otros. Se definen tres tipos de escenarios: de Casos de Uso similares a los definidos en la metodología de Casos de Uso que utiliza el RUP [RUP], de crecimiento que cubren el principio de anticipación al cambio y de exploración que cubren cambios extremos que se espera estresen al sistema.

La información de los escenarios es obtenida y priorizada en ATAM utilizando dos mecanismos diferentes en distintos momentos, y con la participación de diferentes involucrados: los árboles de utilidad y la lluvia de ideas estructurada. El árbol de utilidad se arma partiendo de un nodo raíz que representa la cualidad global en estudio, a partir del cual se van detallando los requerimientos de atributos de calidad hasta llegar a expresiones concretas en las hojas del árbol,

que definen los escenarios para la evaluación. Estos escenarios se priorizan en dos dimensiones: por la importancia que tienen para el éxito del sistema, y por la dificultad que poseen para ser realizados, según estima el Arquitecto.

En ambas dimensiones se utiliza habitualmente la escala Alta, Media y Baja (High, Medium y Low). La salida de la generación del árbol de utilidad es una lista priorizada de los requerimientos de los atributos de calidad especificados como escenarios. La lluvia de ideas de escenarios se realiza con todos los involucrados en el sistema, ya que el objetivo principal es encontrar todos los escenarios posibles según la clasificación vista anteriormente. La lista obtenida por este medio se compara con el árbol de utilidad y se agregan a éste los nuevos escenarios. En la figura 3.3 se presenta un ejemplo simplificado de árbol de utilidad:

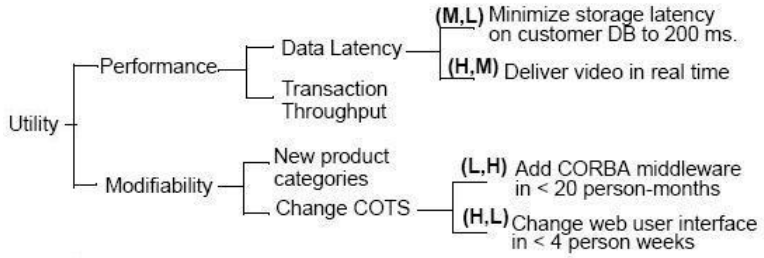


Figura 3.3: Ejemplo de árbol de utilidad con escenarios de calidad definidos en ATAM de [ATAM04]

A partir del árbol de utilidad y la priorización de escenarios obtenida, se evalúa la adecuación de las propuestas arquitectónicas presentes en la Arquitectura de Software en evaluación, respecto de los atributos de calidad especificados, identificando riesgos, no riesgos, puntos de sensibilidad y puntos de concesión. Los riesgos son decisiones arquitectónicas importantes que no se han tomado en el proyecto, o que si se han tomado pero cuyas consecuencias no son comprendidas totalmente. Los no riesgos son las fortalezas de la Arquitectura, aquellas decisiones de diseño arquitectónico tomadas que son adecuadas, es decir, cumplen con los requerimientos de calidad del sistema.

Los puntos de sensibilidad son parámetros en la arquitectura con los cuales se relaciona fuertemente la respuesta a algún atributo de calidad medible, por ejemplo el rendimiento o throughput afectado por la velocidad de un canal de comunicación. Un punto de concesión (tradeoff point) se encuentra en la arquitectura cuando un parámetro arquitectónico refiere a más de un punto de sensibilidad donde los atributos de calidad medibles son afectados en forma distinta según los cambios en dicho parámetro. Por ejemplo si se aumenta la velocidad del canal de comunicación se mejora el rendimiento o throughput pero se reduce la confiabilidad, entonces es un punto de concesión.

Enfoque del método

El corazón de ATAM consiste en la ejecución de nueve pasos que se dividen en cuatro grupos que su vez, se realizan en el tiempo en cuatro Fases diferenciadas. Estos cuatro grupos no se corresponden uno a uno con las Fases, sino que se realizan en las Fases 1 y 2, y se agrega una Fase 0 previa de preparación y una Fase 3 posterior de finalización del proyecto. Si bien la numeración de pasos sugiere linearidad, la ejecución de los mismos no necesariamente es un proceso en cascada estricto, ya que se podrá volver a pasos anteriores o saltar hacia adelante a pasos posteriores, o inclusive iterar entre pasos según sea necesario.

Los cuatro grupos en que se dividen los nueve pasos definidos consisten en un primer grupo de presentación, donde se intercambia información del sistema, un segundo grupo de investigación y análisis, donde se valoran los atributos de calidad claves uno a uno con las propuestas arquitectónicas, un tercer grupo de pruebas donde se revisan los resultados obtenidos contra las

necesidades relevantes de los stakeholders, y un cuarto y último grupo donde se presentan los resultados del ATAM.

En la Fase 0 se acuerdan tiempo, fechas, costos, esfuerzo y se forma el equipo de evaluación, en las Fases 1 y 2 se realiza la evaluación con ATAM siguiendo los nueve pasos especificados, y finalmente en la Fase 3 se realiza el informe final de la evaluación realizada, se recoge información para la medida y mejora del proceso y se actualizan los repositorios de productos generados. Se definen también involucrados del proyecto y roles para los mismos, incluyendo al Arquitecto de Software, Gerentes de Proyecto, Desarrolladores, entre otros, más el equipo que realiza la evaluación ATAM, identificando la participación y responsabilidades de cada uno en los pasos definidos por el método, así como en la generación de los entregables asociados a cada una de las actividades que se realizan. A continuación se describen las Fases 1 y 2 que constituyen la base de ATAM.

Fase 1: En la Fase 1 se realizan las actividades en los grupos de presentación e investigación y análisis.

El grupo de presentación se compone de tres pasos: 1 - Presentar el ATAM, 2 - Presentar las pautas del negocio y 3 - Presentar la Arquitectura. En el primer paso el equipo de ATAM presenta el método a los stakeholders explicando el proceso a seguir y el involucramiento y responsabilidad de cada uno en el proyecto. Se detallan los pasos a seguir, las técnicas a utilizar y los resultados a obtener. En el segundo paso un director de proyecto o gerente presenta el sistema desde el punto de vista del negocio, al equipo de ATAM y a los stakeholders, detallando principales funcionalidades, restricciones y metas definidas para el sistema. En el tercer paso el Arquitecto presenta la Arquitectura de Software definida incluyendo por lo menos los estilos utilizados, otros sistemas con que se debe interactuar, restricciones técnicas de software como por ejemplo uso de sistema operativo. Si se cuenta con un documento de Software Architecture Description (SAD) el Arquitecto debería basar su explicación en las distintas vistas contenidas en el mismo.

El grupo de investigación y análisis se compone también de tres pasos: 4 - Identificar las propuestas arquitectónicas, 5 - Generar el árbol de utilidad de los atributos de calidad y 6 - Analizar las propuestas arquitectónicas. En el cuarto paso el equipo de ATAM le pide al Arquitecto que identifique las propuestas arquitectónicas o estilos de arquitectura utilizados, ya que éstos definirán las estructuras importantes definidas para el sistema y las características implicadas. En el quinto paso se genera el árbol de utilidad donde principalmente el Arquitecto pero también los principales stakeholders, identifican, priorizan y refinan los requerimientos de atributos de calidad más importantes del sistema, según se mencionó previamente, identificando los escenarios en el árbol y su importancia en las dos dimensiones definidas. En el sexto paso se analizan las propuestas arquitectónicas según el árbol de utilidad generado, esto es, que tan adecuados son el uno para el otro, evaluando como cada propuesta arquitectónica influye en la obtención o no del atributo de calidad requerido, e identificando los riesgos, no riesgos, puntos de sensibilidad y concesión asociados a dicha evaluación.

Fase 2: En la Fase 2 se realizan las actividades en los grupos de pruebas e informes.

El grupo de pruebas se compone de dos pasos: 7 - Lluvia de ideas y 8 - Analizar las propuestas arquitectónicas y el grupo de Informes se compone de un solo paso: 9 - Presentar los resultados. En el séptimo paso se confirman e identifican nuevos escenarios según varios stakeholders involucrados, los que también se priorizan y se comparan con los identificados en el árbol de utilidad. Pueden suceder tres casos: el escenario ya está en el árbol de utilidad, no está pero encaja en alguna rama y se convierte en una nueva hoja, no está y no encaja en ninguna rama, lo que significa que no había sido considerado previamente. En el octavo paso se realiza lo mismo que en el sexto paso para el nuevo árbol de utilidad con todos los escenarios incluidos.

Finalmente en el noveno paso el equipo de ATAM presenta los resultados a los stakeholders y entrega la documentación correspondiente a las salidas del ATAM: documento de propuestas arquitectónicas, conjunto de escenarios priorizados, conjunto de preguntas basadas en los atributos, árbol de utilidad, los riesgos descubiertos, los no riesgos documentados, los puntos

de sensibilidad y de concesión encontrados, más la relación entre los riesgos encontrados y su impacto en las pautas del negocio definidas.

Beneficios de utilizar ATAM

Como resultado de la aplicación del método se obtiene una lista de escenarios priorizados, y las diferentes propuestas arquitectónicas que los intentan satisfacer, a partir de las cuales es posible descubrir puntos de sensibilidad, de concesión, y riesgos asociados. Estas propuestas arquitectónicas son una excelente guía para tomar futuras decisiones arquitectónicas, ya que se analiza su impacto en la Arquitectura, y se atacan directamente las expectativas de calidad que el cliente tiene sobre el sistema.

El árbol de utilidad donde se identifican, priorizan y refinan los requerimientos de atributos de calidad más importantes del sistema, aportan a que los involucrados, principalmente el Arquitecto pero también los principales stakeholders, reflexionen sobre los requerimientos y las distintas posibilidades en el diseño de la Arquitectura para cumplir con estos. El análisis de las propuestas arquitectónicas según el árbol de utilidad generado, brinda la visión de que tan adecuados son el uno para el otro, evaluando como cada propuesta arquitectónica influye en la obtención o no del atributo de calidad requerido, e identificando los riesgos, no riesgos, puntos de sensibilidad y concesión asociados a dicha evaluación.

Además la documentación generada por ATAM: documento de propuestas arquitectónicas, conjunto de escenarios priorizados, conjunto de preguntas basadas en los atributos, árbol de utilidad, riesgos descubiertos, no riesgos documentados, puntos de sensibilidad y de concesión encontrados, más la relación entre los riesgos encontrados y su impacto en las pautas del negocio definidas, aportan a la clarificación de la Arquitectura para todos los involucrados, constituyendo así material valioso de consulta y referencia en el proyecto y en proyectos futuros.

La evaluación con ATAM permite tomar un enfoque amplio del sistema, en lugar de uno estrecho o con miras a corto plazo. Gracias a este tipo de enfoque que plantea el método de evaluación es posible descubrir riesgos ocultos y errores que permitan identificar posibles problemas en la construcción del producto. Además la metodología de ATAM puede ser adaptada a las necesidades particulares de cada negocio, como se hizo en el caso de estudio presentado en [ATAM07].

Capítulo 4

Service Oriented Architecture (SOA)

4.1. Introducción

En este capítulo se presenta el estado actual del conocimiento en el enfoque Service Oriented Architecture (SOA), que constituye la base para el trabajo de tesis realizado en la metodología SOA propuesta para desarrollos con dicho enfoque. En primer lugar en la sección 4.2 se presenta el enfoque SOA, brindando el contexto en que se inscribe, presentando definiciones, conceptos, elementos que lo componen, enfoques de desarrollo con SOA y aspectos técnicos del mismo. En segundo lugar en la sección 4.3 se presenta el enfoque de Business Process Management (BPM) que constituye una parte importante del desarrollo orientado al negocio que es una de las bases de SOA y cuya utilización conjunta está tomando cada vez más importancia.

Finalmente, en la sección 4.4 se presenta el enfoque Model Driven Architecture (MDA), que tiene importancia en si mismo como enfoque de desarrollo, pero también por su relación actual con el enfoque SOA, donde la última tendencia es conjuntar ambos enfoques para realizar "desarrollo basado en modelos de aplicaciones orientadas a servicios". Los enfoques SOA y MDA, así como el estado actual de lo que proponen, se apoyan en los conceptos sobre Diseño y Arquitectura de Software presentados en el capítulo 3.

Parte de la sección 4.2 fue incluida como contexto teórico de los artículos publicados y presentados en las "V Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento (JIISIC'06)" realizadas en Puebla, México, en febrero de 2006 [ESOA06], y en la "XXXII Conferencia Latinoamericana de Informática (CLEI'06)" realizada en Santiago de Chile, Chile, en Agosto de 2006 [TMAD06]. Parte de la sección 4.4 corresponde a un artículo sobre desarrollo con enfoque MDA publicado y presentado en las "VI Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento (JIISIC'07)" realizadas en Lima, Perú, en febrero de 2007 [EMDA07].

4.2. Service Oriented Architecture (SOA)

En esta sección se describe el enfoque Service Oriented Architecture (SOA), primero en la subsección 4.2.1 se presenta el contexto en el cual se inscribe el enfoque SOA, en la subsección 4.2.2 se presenta el enfoque SOA y su importancia, en la subsección 4.2.3 se presentan los principales conceptos involucrados en SOA, en la subsección 4.3 se presenta el enfoque Business Process Management (BPM) y su relación con SOA, finalmente en la subsección 4.2.4 se presentan enfoques para el desarrollo SOA en los cuales se basa la propuesta de metodología SOA definida que se presenta en el capítulo 5.

4.2.1. Contexto de SOA

El área de Tecnología Informática (TI) en las Organizaciones actuales se puede caracterizar por tener diversidad de sistemas que tienen entre sí dependencias complejas, que han ido creciendo en forma separada y heterogénea a lo largo de los años. Un desafío que se plantea es poder integrarlos para reaccionar ágilmente a los cambios en los requerimientos del negocio, principalmente en dos aspectos: los procesos de la Organización y las tecnologías disponibles. La definición y disponibilidad de estos servicios para toda la Organización es la base del enfoque SOA [KRAF05].

El concepto de SOA está enfocado en la definición de una infraestructura del Negocio, con el término servicio lo que se está denotando es un servicio del Negocio como puede ser realizar una reservación aérea o acceder la base de datos de la compañía de un cliente. El servicio provee operaciones del Negocio como hacer una reservación, cancelar una reservación o recuperar el perfil de un cliente. Sin embargo también hay servicios técnicos de infraestructura que proveen operaciones como comienzo de transacción o modificación de datos. Una SOA debe desacoplar aplicaciones del Negocio de servicios técnicos y hacer que la Organización sea independiente de una implementación técnica específica o infraestructura. [KRAF05]

En [ERLT05] se establece que "La lógica colectiva que define y guía una empresa es una entidad siempre en evolución que cambia constantemente en respuesta a influencias externas e internas. Desde la perspectiva de TI, esta lógica empresarial puede ser dividida en dos mitades importantes: lógica del negocio y lógica de la aplicación. Cada una existe en un mundo por si mismo, y cada una representa una parte necesaria de la estructura organizacional contemporánea. La lógica del negocio es una implementación documentada de los requerimientos del negocio que se origina en el área del negocio de la empresa."

Además "La lógica del negocio está generalmente estructurada en procesos que expresan esos requerimientos, así como cualquier restricción asociada, dependencia e influencias externas. La lógica de la aplicación es una implementación automatizada de la lógica del negocio organizada en varias soluciones tecnológicas. La lógica de la aplicación expresa flujos de los procesos del negocio mediante sistemas adquiridos o desarrollados especificamente por la infraestructura de TI de la Organización, con restricciones de seguridad, capacidad técnica y dependencias con vendedores." [ERLT05] En la figura 4.1 se muestran estos conceptos.

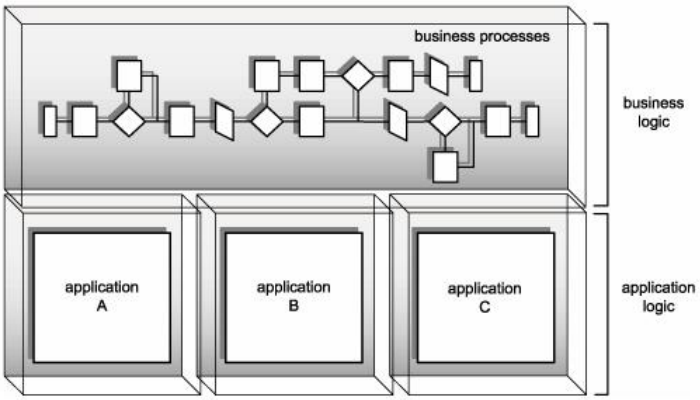


Figura 4.1: Dos aspectos claves de las Organizaciones: procesos del Negocio y tecnologías de [ERLT05]

Como se establece en [ERLT05] "Los conceptos introducidos por la orientación a servicios son realizados mediante la introducción de servicios. Los servicios establecen una forma de abstracción de alto nivel entre las capas tradicionales del negocio y la aplicación. Al posicionarse entre medio, los servicios pueden encapsular la lógica de la aplicación física así como la lógica

de los procesos del negocio. Los servicios modularizan la empresa, formando unidades de lógica independientes que existen mediante una capa común de conectividad. Los servicios pueden ser estratificados de forma que servicios padre puedan encapsular servicios hijos. Esto permite que la capa de servicios a su vez consista de múltiples capas de abstracción. ... En un nivel físico, los servicios son desarrollados y distribuidos en ambientes propietarios, mientras son individualmente responsables por el encapsulamiento de la lógica de la aplicación específica." En la figura 4.2 se muestra como los servicios en la capa de interface de servicios representan lógica de la aplicación realizada en distintas plataformas.

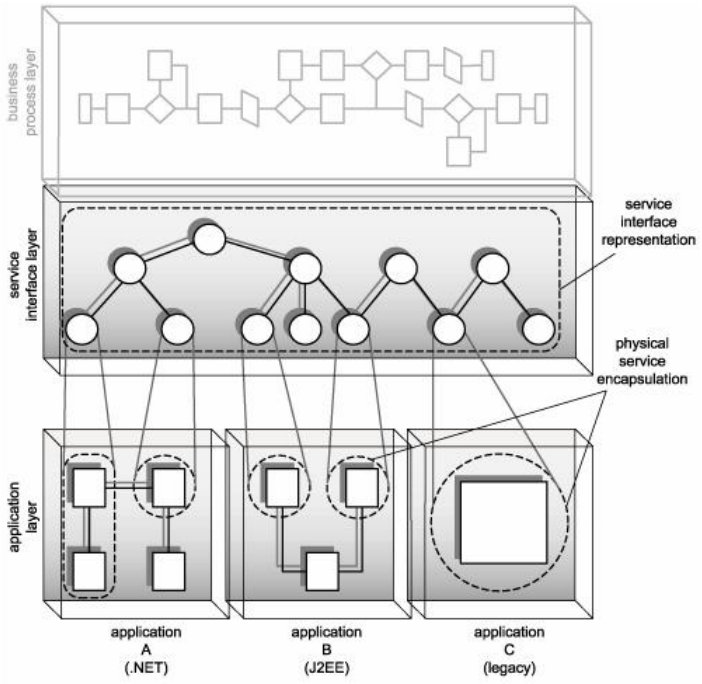


Figura 4.2: Capa de servicios entre los procesos del Negocio y las tecnologías de [ERLT05]

En el mismo sentido, en [ENDR04] se menciona que los problemas de heterogeneidad, interoperabilidad y requerimientos cambiantes, son solucionados por una SOA que provee una plataforma para la construcción de aplicaciones basadas en servicios con las características de bajo acoplamiento, ubicación transparente de servicios e independencia de protocolos. Un consumidor de servicios no debe preocuparse por un servicio particular con el que comunicarse debido a que la infraestructura por debajo, el bus de servicios, puede hacer la elección en representación del consumidor. La infraestructura esconde la mayor cantidad de detalles técnicos posible al consumidor, y diferentes tecnologías de implementación como J2EE [J2EE] o .NET [MNET] no afectan a los usuarios de una SOA. También debe ser posible sustituir una implementación de un servicio por una mejor si la hay disponible, con mejores características de calidad de servicio, según la metadata que describe al servicio.

4.2.2. Presentación de SOA

En las últimas décadas se han experimentado cambios importantes en las tecnologías y paradigmas de desarrollo y uso de aplicaciones, que han impactado fuertemente en la forma en que las Organizaciones realizan e implementan su negocio. Por un lado el acceso cada vez mayor a Internet pone a disposición información y software, permite comunicación entre distintas Organizaciones, sus proveedores y clientes, ampliando tanto los requerimientos del negocio como sus necesidades y realización. Las mejoras en la tecnología continúan en forma acelerada, incrementando a su vez el ritmo de los cambios en los requerimientos. El negocio debe adaptarse

rápidamente para sobrevivir en el ambiente dinámico actual, y la infraestructura de TI debe permitir la habilidad de adaptación del negocio [ENDR04]. Esta evolución desde las tecnologías al negocio y viceversa, presenta una parte importante en la evolución de las Arquitecturas de Software desde los primeros sistemas monolíticos hasta la visión actual de orientación a servicios, que se muestra en la figura 4.3.

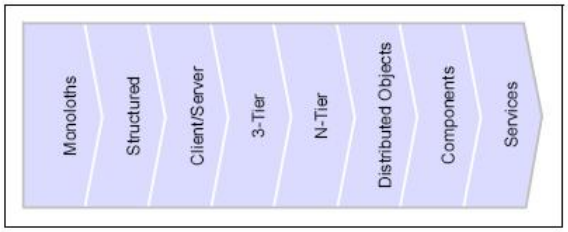


Figura 4.3: Evolución de Arquitecturas de Software hasta SOA de [ENDR04]

Así como no hay una única definición para Arquitectura de Software, como se presentó en la sección 3.2.2, tampoco existe una única definición para Service Oriented Architecture (SOA), las que sin embargo comparten algunos conceptos base. A continuación se presentan las definiciones de las Organizaciones de mayor importancia y esfuerzos para la estandarización de este enfoque: OMG [OMG], OASIS [OASIS] y W3C [W3C]. Estas Organizaciones tienen entre sus estándares definidos, algunos tan importantes y ampliamente utilizados como UML [UML] y XML [XML].

Para el Object Management Group (OMG) [OMG] Service Oriented Architecture (SOA) es un estilo de arquitectura para que una comunidad de proveedores y consumidores de servicios alcancen valor mutuo. Permite que los participantes en las comunidades trabajen juntos con una mínima co-dependencia o dependencia tecnológica, mediante la especificación de contratos a los que las Organizaciones, personas y tecnologías deben adherir para participar en la comunidad. Provee la realización por parte de la comunidad, de procesos del negocio y permite que una variedad de tecnologías sean utilizadas para facilitar las interacciones en la comunidad.

Según la Organization for the Advancement of Structured Information Standards (OASIS) [OASIS] Service Oriented Architecture (SOA) es un paradigma para organizar y utilizar capacidades distribuidas que pueden estar bajo el control de distintas Organizaciones. Éstas crean distintas soluciones a los problemas que enfrentan en el curso de sus negocios, que constituyen las capacidades de la Organización para cubrir sus necesidades. La correlación entre necesidades y capacidades no es uno a uno, por lo que una necesidad puede requerir más de una capacidad para ser cubierta, y una capacidad puede cubrir más de una necesidad. El valor percibido de SOA es que provee un framework fundamental para asociar necesidades y capacidades y para combinar capacidades para cubrir esas necesidades.

En World Wide Web Consortium (W3C) [W3C] Service Oriented Architecture (SOA) se define como una forma de arquitectura de sistemas distribuidos que se caracteriza por presentar servicios como una vista lógica abstracta de programas, bases de datos, procesos de negocio, cuya semántica se documenta en su descripción, los que tienden a usar un número pequeño de operaciones con mensajes relativamente grandes y complejos, lo que define su granularidad, y tienden a ser orientados a uso sobre la red, aunque esto no es un requerimiento absoluto. Los mensajes son enviados en formato estandarizado neutral a plataformas a través de las interfaces, donde XML es el formato más obvio que cumple estas restricciones.

Se pueden encontrar también otras definiciones en la literatura del área, que son más elaboradas y que hacen énfasis además en otros aspectos que componen el enfoque. Entre la gran variedad de definiciones y enfoques existentes, se seleccionaron los presentados en los libros [ENDR04], [ERLT05] y [KRAF05], que constituyen los enfoques de referencia para la realización de este trabajo.

En [ENDR04] Service Oriented Architecture (SOA) se presenta como un enfoque para la construcción de sistemas distribuidos que libera funcionalidad de aplicación como servicios tanto a usuarios finales de aplicaciones como a otros servicios, que se compone de elementos categorizados en funcionales y de calidad de servicio. Los aspectos funcionales incluyen mecanismo de transporte y protocolos para intercambio de mensajes, descripción de servicios con esquema acordado que indica que es el servicio, como debe ser invocado, y que datos se requieren para invocarlo, y finalmente el servicio en si mismo disponible para ser usado. Los aspectos de calidad incluyen políticas como conjuntos de condiciones o reglas bajo las que los proveedores de servicios los hacen disponibles a los consumidores, por ejemplo políticas de seguridad para la identificación, autorización, y control de acceso a los consumidores de servicios, transaccionalidad para liberar un resultado consistente por parte de varios servicios trabajando juntos, entre otros.

Agrega también que los procesos de Negocio son una colección de servicios invocados en una secuencia particular con un conjunto particular de reglas, para cumplir un requerimiento del negocio, que puede ser considerado como un servicio en si mismo, por lo que los procesos de negocio pueden componerse de servicios de diferente granularidad. El registro de servicios es un repositorio de servicios y descripciones de datos que puede ser usado por los proveedores para publicar sus servicios, y por los consumidores para descubrirlos.

En [ERLT05] se establece "El témino Service Oriented Architecture (SOA) representa un modelo en el cual la lógica de automatización se descompone en distintas unidades de menor tamaño. Colectivamente estas unidades componen una pieza de mayor tamaño de lógica de automatización del negocio. Individualmente, estas unidades pueden ser distribuidas. ... SOA fomenta que las unidades individuales de lógica existan en forma autónoma pero no aisladas unas de otras. Se requiere que las unidades de lógica conformen un conjunto de principios que les permitan evolucionar en forma independiente, pero a la vez manteniendo una cantidad suficiente de aspectos comunes y estandarización. En SOA, estas unidades de lógica se conocen como servicios."

Agrega que los principios más importantes de la orientación a servicios están relacionados con el bajo acoplamiento entre servicios, autonomía de cada servicio, descubrimiento de servicios brindados por proveedores por parte de los consumidores, composición de servicios, reutilización de servicios, contratos de servicio, abstracción y servicios sin estado.

Según [KRAF05] "una Service Oriented Architecture (SOA) es una Arquitectura de Software que está basada en los conceptos claves de application frontend, servicio, repositorio de servicios y bus de servicios. Un servicio consiste en un contrato, una o más interfaces, y una implementación." SOA se basa en la definición de servicios reutilizables, con interfaces públicas bien definidas, donde los proveedores y consumidores de servicios interactúan en forma desacoplada para realizar los procesos de negocio. Un servicio consiste en una implementación que provee lógica de negocio y datos, un contrato de servicio, las restricciones para el consumidor, y una interfaz que expone físicamente la funcionalidad que provee.

Además las application frontend se pueden ver como consumidores de servicios que forman procesos de negocios, que pueden estar encapsulados en un único servicio que se compone a partir de otros de menor granularidad en una secuencia definida de pasos para la ejecución del proceso del Negocio modelado. Un repositorio de servicios almacena los contratos de servicios que brindan los proveedores para hacerlos disponibles a los consumidores y el bus de servicios interconecta las application frontend y los servicios, así como los servicios entre sí.

La definición de SOA que se toma para este trabajo es la dada en [KRAF05] en la cual se introducen la mayoría de los elementos mencionados como parte de SOA, además de los servicios y sus características en que se basa el enfoque, lo que la hace una de las más completas. La propuesta de [KRAF05] además provee otros elementos que se mencionan más adelante, que permite tratar el enfoque SOA tomando en cuenta diversos aspectos que hacen a su incorporación en el desarrollo.

4.2.3. Conceptos en SOA

En esta subsección se presentan los conceptos más importantes involucrados en el enfoque Service Oriented Architecture (SOA). En primer lugar se presentan los elementos principales que consituyen el enfoque en 4.2.3.1, luego se presentan clasificaciones de servicios en 4.2.3.2, distintos niveles de SOA se describen en 4.2.3.4, y finalmente aspectos técnicos relacionados con implementaciones de SOA se presentan en 4.2.3.5.

4.2.3.1. Elementos de SOA

Como se presentó en [KRAF05] SOA se compone de elementos claves que son: las application frontend, los servicios, el repositorio de servicios y el bus de servicios, donde los servicios constituyen la base del enfoque.

Las application frontend son los elementos activos de SOA, inician y controlan las actividades en los sistemas empresariales. Pueden tener interfaces gráficas para usuarios como ser las aplicaciones Web, o ser procesos batch que no interactúan directamente con los usuarios. Aunque mucha de la responsabilidad sobre los procesos del negocio se delegue en uno o más servicios, es siempre un application frontend quién inicia un proceso del negocio y recibe los resultados del mismo. [KRAF05]

El repositorio de servicios provee facilidades para descubrir servicios y adquirir la información para utilizarlos, particularmente si estos servicios deben ser descubiertos por fuera del alcance funcional y temporal del proyecto que los creo. Aunque mucha de la información requerida es parte del contrato de servicio, el repositorio de servicios provee información adicional como ubicación física, información sobre el proveedor, personas de contacto, tarifas de uso, restricciones técnicas, aspectos de seguridad y niveles de disponibilidad del servicio. Debe notarse que el enfoque está dado en repositorios dentro de los límites de una Organización, repositorios que son usados para integración de servicios cruzando fronteras entre Organizaciones, típicamente tienen otros requerimientos - en particular, los que se hacen públicos en Internet. [KRAF05]

El bus de servicios conecta a todos los participantes de SOA - servicios y application frontends - entre si. Si dos participantes necesitan comunicarse - por ejemplo, si una application frontend necesita invocar alguna funcionalidad de un servicio básico - el bus de servicios hace esto posible. En algún aspecto, el bus de servicios es similar en concepto al bus de software como está definido en el contexto de CORBA [CORB]. Sin embargo, existen diferencias significativas entre estos conceptos. La más importante, es que el bus de servicios no está compuesto necesariamente de una única tecnología, sino que preferiblemente se compone de una variedad de productos y conceptos. Como características principales debe proveer: conectividad entre los elementos de SOA, heterogeneidad de tecnología, heterogeneidad de conceptos de comunicación por ejemplo comunicación sincrónica y asincrónica, y servicios técnicos como seguridad, transformación de mensajes, auditoría, entre otros. [KRAF05]

Los servicios representan grupos lógicos de operaciones relacionadas con algún concepto del negocio. Un servicio consiste en una implementación que provee lógica de negocio y datos, un contrato de servicio, las restricciones para el consumidor, y una interfaz que expone físicamente la funcionalidad. La implementación del servicio provee la lógica del Negocio requerida y los datos correspondientes. Es la realización técnica que cumple con el contrato definido, en forma de componentes, programas, datos de configuración, bases de datos. El contrato de servicios provee la especificación del propósito, funcionalidad, restricciones y uso del servicio. La forma de esta especificación varía dependiendo del tipo de servicio. Las funcionalidades del servicio se exponen en la interface del servicio a clientes conectados en la red. Existen distintos tipos de servicios con determinadas características que tienen distinto significado en SOA, por ejemplo los procesos del negocio se realizan en servicios orientados a procesos que se componen de secuencias definidas de invocaciones a servicios. [KRAF05]

Según [ERLT05] SOA se compone de mensajes, operaciones, servicios y procesos (e instancias de procesos). Un mensaje representa los datos que son requeridos para completar alguna o

todas las partes de una unidad de trabajo. Los Message Exchange Patterns (MEPs) definen patrones de intercambios de mensajes comúnmente utilizados. Una operación representa la lógica requerida para procesar los mensajes de forma de completar una unidad de trabajo. Un servicio representa una agrupación lógica de un conjunto de operaciones capaz de realizar unidades de trabajo relacionadas.

Un proceso contiene las reglas del negocio que determinan que operaciones de servicio son utilizadas para completar una unidad de automatización, en otras palabras, un proceso representa una pieza más grande de trabajo que requiere que se completen unidades de trabajo más pequeñas. Adicionalmente los servicios adhieren a un acuerdo de comunicación, definido en forma colectiva por una o más descripciones de servicios y documentos asociados, llamados contratos de servicios. Los servicios tienen control sobre la lógica que encapsulan y minimizan la retención de información específica de una actividad, siendo generalmente, sin estado. Son diseñados para ser descriptivos de forma que puedan ser encontrados y evaluados vía mecanismos de descubrimiento disponibles, por ejemplo registro de servicios. [ERLT05]

Los servicios son unidades de lógica independientes. Para retener su independencia, los servicios encapsulan lógica dentro de un contexto distinto. Este contexto puede ser específico a una tarea del negocio, a una entidad del negocio, o alguna otra agrupación de lógica. El interés tratado por un servicio puede ser pequeño o grande. Por lo tanto, el tamaño y alcance de la lógica representada por el servicio puede variar. Lo que es más, la lógica de un servicio puede contener la lógica provista por otros servicios. En ese caso, uno o más servicios se componen en uno colectivo. [ERLT05]

Por ejemplo, las soluciones de automatización del negocio son tipicamente una implementación de un proceso del negocio. Este proceso se compone de lógica que dicta las acciones realizadas por la solución. La lógica se descompone en una serie de pasos que se ejecutan en una secuencia predefinida de acuerdo a reglas del negocio y condiciones de ejecución. Al constuir una solución automatizada que consiste en servicios, cada servicio puede encapsular una tarea realizada por un paso individual o un subproceso compuesto de un conjunto de pasos. [ERLT05] En la figura 4.4 se muestran estos conceptos.

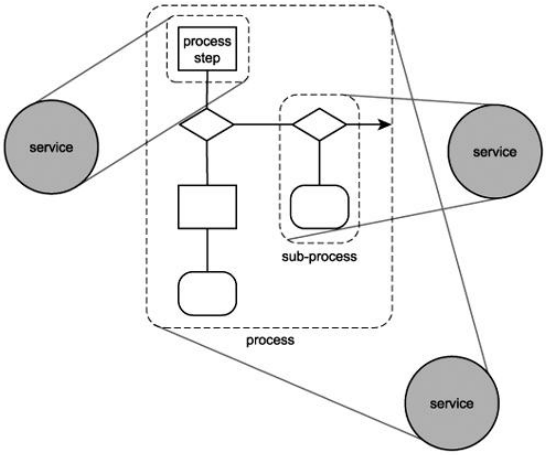


Figura 4.4: Granularidad de servicios en SOA de [ERLT05]

En [STEV04] los elementos de que se compone una SOA son: los consumidores y proveedores de servicios, el registro de servicios, el contrato de servicio, el proxy del servicio y el arrendamiento del servicio.

Un consumidor de servicios puede ser una aplicación, otro servicio, o cualquier elemento de software que requiera un servicio. Inicia la ubicación de un servicio en el registro, y una vez que

obtiene la ubicación física del mismo, lo ejecuta enviando una solicitud formateada de acuerdo al contrato del servicio. Un proveedor de servicios es el servicio, la entidad direccionable en la red que acepta y ejecuta las solicitudes de los consumidores. Puede ser cualquier tipo de componente de software. El proveedor de servicios publica su contrato en el registro para que sea accedido por los consumidores de servicios.

El contrato de servicio es una especificación de la forma que un consumidor de un servicio deberá interactuar con el proveedor del servicio. Especifica el formato de la solicitud y respuesta del servicio, precondiciones, postcondiciones, y podría especificar características de calidad de servicio (QoS), por ejemplo, tiempo de ejecución del servicio. El registro de servicios es un directorio en la red que contiene los servicios disponibles. Es una entidad que acepta y almacena contratos de servicios de los proveedores de servicios y provee dichos contratos a los consumidores de servicios interesados. El arrendamiento del servicio también es entregado por el registro al consumidor y especifica el tiempo en que el contrato entregado es válido.

Según [STEV04] entonces, SOA consiste en los elementos mencionados que configurados en conjunto proveen el soporte para el paradigma base de SOA "descubrir, ligar e invocar" (find, bind and invoke) donde un consumidor de un servicio obtiene la ubicación de otro servicio consultando el repositorio de servicios con determinado criterio. Si el servicio existe, el registro de servicios le provee al consumidor el contrato del servicio y la dirección física del servicio. En la figura 4.5 se muestra el paradigma y su funcionamiento.

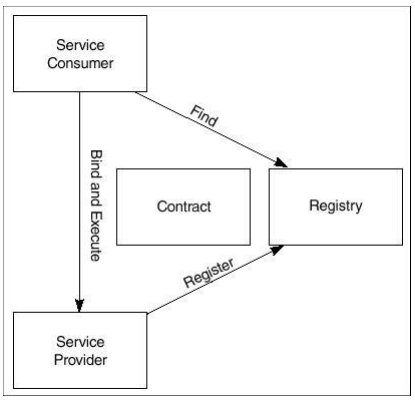


Figura 4.5: Paradigma "descubrir, ligar e invocar" para descubrimiento e invocación de servicios en un registro de servicios de [STEV04]

Según [ENDR04] para que este paradigma funcione se deben poder realizar las funciones: Publicar, la descripción de un servicio debe ser publicada para estar accesible, de forma de ser descubierto e invocado por un consumidor del servicio; Descubrir, un consumidor de un servicio lo localiza consultando al registro de servicios por un servicio que cumpla sus criterios; Ligar e invocar, luego de recuperar la descripción del servicio, el consumidor del servicio procede a invocarlo de acuerdo a la información provista en dicha descripción.

4.2.3.2. Clasificación y relaciones entre servicios

Existen varios enfoques para la clasificación de servicios. Cualquiera sea la clasificación o la granularidad de la misma, su importancia radica en el hecho de que sirven como guía al momento de diseñar los servicios. Sabiendo lo que se quiere obtener resulta más fácil poder definir en base al negocio, los distintos servicios necesarios para cumplir con los requerimientos planteados. Un aspecto común en las clasificaciones que se presentan refiere a la existencia de servicios orientados a procesos del Negocio o servicios que encapsulan procesos del Negocio, haciendo énfasis en el enfoque de desarrollo hacia el Negocio que se plantea en SOA.

La clasificación de servicios dada en [KRAF05] los separa en básicos, intermediarios, centrados en procesos y empresariales públicos.

- Los servicios básicos son la base de SOA, son servidores puros y no mantienen el estado conversacional de la sesión. Se dividen a su vez en servicios centrados en los datos y centrados en la lógica.
  - Los servicios centrados en los datos tienen como propósito manejar los datos persistentes, almacenamiento, recuperación, mecanismos de lockeo y gestión transaccional. Se comportan similar a la capa de acceso a datos de una aplicación tradicional. La diferencia es que la capa de acceso a datos maneja los datos de la aplicación entera, mientras que un servicio centrado en los datos trata con una sola entidad del negocio. Cualquier otro servicio que requiera acceso a estos datos debe usar la interface del servicio, por lo que una aplicación podría requerir entonces varios servicios centrados en los datos coordinados.
  - Los servicios centrados en la lógica proveen encapsulamiento para cálculos complejos o reglas del Negocio, tradicionalmente encapsulados en bibliotecas y frameworks del negocio. Por ejemplo un motor de productos de seguros que encapsula el conocimiento, términos y condiciones de los productos de una compañía de seguros.
- Los servicios intermediarios son servicios sin estado que hacen de puente entre las inconsistencias técnicas o discrepancias conceptuales en el diseño. Son tanto consumidores como proveedores, mediando entre los distintos elementos que deben funcionar juntos. Se dividen a su vez en technology gateways, adapters, facades y funcionality-adding.
  - Los servicios Technology gateways hacen de puente entre discrepancias tecnológicas, incorporando dos o más tecnologías para comunicación o codificación de datos. Hacen de proxys para sus servicios de Negocio y representan la funcionalidad de los servicios que están por debajo en un ambiente que es distinto tecnológicamente que el ambiente de ejecución del servicio de Negocio original.
  - Los servicios Adapters son un tipo especial de servicio intermediario que mapea las firmas y formatos de mensaje de un servicio a los requerimientos de un cliente.
  - Los servicios Facade tienen como propósito proveer una vista distinta (y posiblemente agregada) de uno o más servicios existentes, por lo que también pueden actuar como technology gateways y/o adapters. Una facade puede utilizarse para proveer una vista específica de un conjunto de servicios que se encuentran por debajo, que en general son servicios básicos.
  - Los servicios Funcionality-adding se usan cuando se quiere agregar funcionalidad a un servicio existente sin cambiar el servicio mismo. En este caso se crea un servicio que provea la misma funcionalidad del servicio original y agregue las nuevas características requeridas.
- Los servicios centrados en procesos encapsulan el conocimiento de los procesos de Negocio de la Organización y controlan y mantienen el estado del proceso en ejecución. Son tanto consumidores como proveedores y preservan el estado, desde el punto de vista técnico son la clase de servicios más sofisticada. Una de las principales ventajas que presentan estos servicios es que separan la lógica de los procesos, definiendo los procesos de negocio y su control, en base a una secuencia de invocación de servicios existentes. Los servicios centrados en procesos manejan procesos de larga vida, en particular si involucran interacción con usuarios o con servicios del backend en forma asincrónica. Los procesos del negocio se realizan en estos servicios centrados en procesos, que se componen de secuencias definidas de invocaciones a otros servicios, mediante lo que se conoce como orquestación de servicios.

- Los servicios empresariales públicos a diferencia de los anteriores son servicios que una empresa ofrece a socios y clientes externos. Estos servicios tienen requerimientos específicos de facturación de uso, interfaz, desacoplamiento y seguridad ya que las empresas deben acordar claramente como se realiza el uso de los mismos.

En la figura 4.6 se presenta la ubicación en capas de servicios y las relaciones entre servicios según la clasificación presentada en [KRAF05].

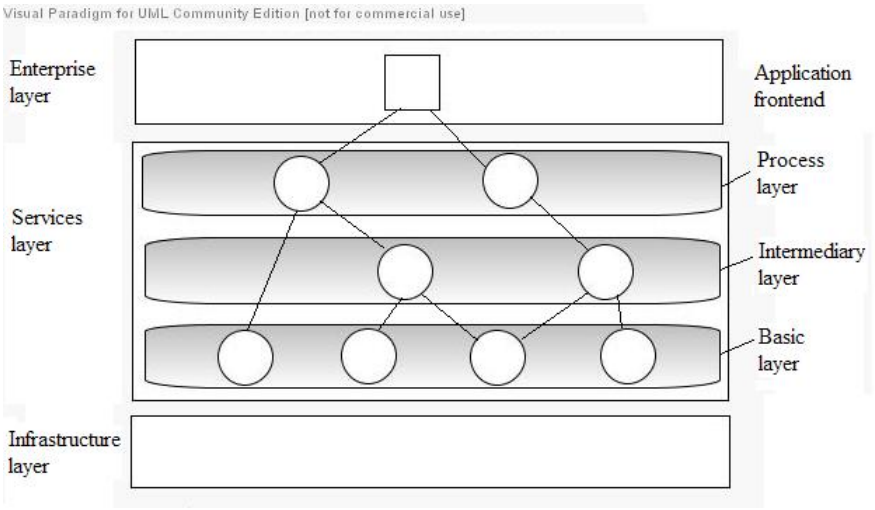


Figura 4.6: Capas de servicios según la clasificación de [KRAF05]

En [ERLT05] la clasificación de servicios define servicios de aplicación, servicios del negocio y servicios de procesos.

- Los servicios de aplicación exponen funcionalidad en un contexto específico de procesamiento, están sobre los recursos que proveen las plataformas, son genéricos y reusables, presentan distinta granularidad en su interface, pueden consistir en una mezcla de servicios desarrollados en forma particular y servicios de terceros comprados o arrendados. Un servicio de aplicación puede componer otros servicios de aplicación de menor granularidad en una unidad de mayor granularidad de lógica de la aplicación. Esta agregación es frecuente para soportar requerimientos de integración, dando lugar a que estos servicios se nombren también como *servicios de integración*, comunmente implementados como *controladores*. Los servicios de aplicación se dividen a su vez en servicios de utilidad y wrappers.

    - Los servicios de utilidad son servicios que ofrecen lógica reutilizable que es genérica y no específica a una aplicación. Cuando existe capa de servicios del negocio por encima se tiende a que los servicios de aplicación sean servicios de utilidad, proveyendo operaciones reusables que pueden componerse por los servicios del negocio para realizar requerimientos de procesamiento centrados en el negocio. Cuano no existe capa de servicios del negocio por encima, muchas veces se embebe lógica del negocio en los servicios de aplicación, dando lugar a los *servicios híbridos*, si bien no recomendados, son comunes.
    - Los servicios wrapper encapsulan y exponen lógica que reside en un sistema legado. Son generalmente utilizados para propósitos de integración. Consisten en servicios que encapsulan alguna o todas las partes de un ambiente legado para exponer funcionalidad legada a los servicios solicitantes. La forma más frecuente de servicio wrapper es un servicio adapter provisto por los vendedores del sistema legado.

- Los servicios del negocio representan lógica del negocio en la forma más pura posible. Sin embargo esto no previene que según las variaciones de lo que implementa, un servicio del negocio pueda ser clasificado también como un servicio controlador (de integración) o un servicio de utilidad (híbrido). De hecho, cuando la lógica de la aplicación se abstrae en una capa aparte de servicios de aplicación, los servicios del negocio pueden actuar como controladores para componer los servicios de aplicación disponibles para ejecutar su lógica del negocio. Los servicios del negocio se dividen a su vez en servicios del negocio centrados en tareas y servicios del negocio centrados en entidades.
  - Los servicios del negocio centrados en tareas encapsulan lógica del negocio específica a una tarea o proceso del negocio. Este tipo de servicios en general es requerido cuando la lógica del proceso del negocio no está centralizada en servicios de procesos en la capa por encima. Los servicios del negocio centrados en tareas tienen potencial de reuso limitado.
  - Los servicios del negocio centrados en entidades encapsulan una entidad del negocio específica. Son útiles para crear servicios del negocio altamente reusables independientes de los procesos del negocio, que luego puedan componerse en servicios de procesos o servicios del negocio centrados en tareas, o ambos.
- Los servicios de procesos componen servicios del negocio y de la aplicación de acuerdo a reglas del negocio y lógica del negocio embebida en la definición del proceso. Componen otros servicios que proveen conjuntos específicos de funciones independientes de las reglas del negocio, con lógica específica de un escenario requerida para ejecutar una instancia de un proceso. Los servicios de procesos son también servicios controladores ya que requieren componer otros servicios para ejecutar lógica de los procesos del negocio. Estos servicios son de los más complejos ya que involucran orquestación de servicios, y por lo tanto la introducción de nuevo middleware en la infraestructura de TI que incluye servidores de orquestación, que no son una adición trivial ni de poco costo.

En la figura 4.7 se presenta la ubicación en capas de servicios y las relaciones entre servicios según la clasificación presentada en [ERLT05].

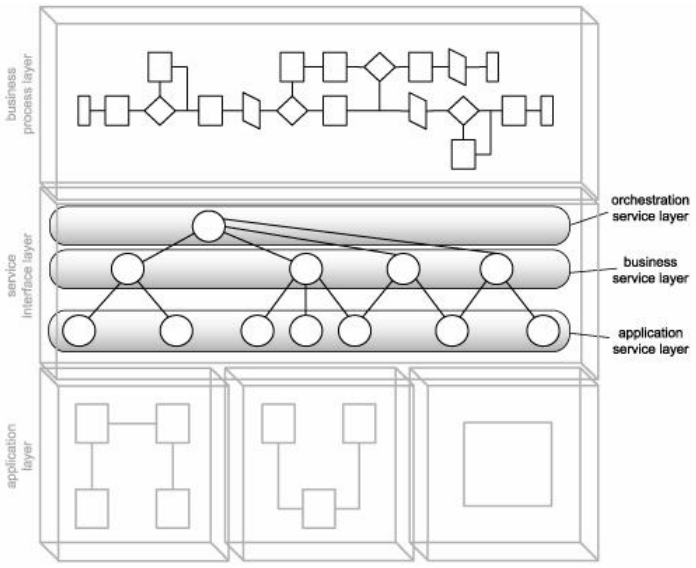


Figura 4.7: Capas de servicios según la clasificación de [ERLT05]

Se puede observar en las clasificaciones presentadas, que la categorización de servicios que se realiza, tiene en todos los casos enfoque del Negocio. Esto significa que cuando se está diseñando

con orientación a servicios, las piezas de software a obtener deben tener todas relación directa con conceptos, entidades, reglas y procesos del Negocio en la Organización, maximizando el desacoplamiento entre los elementos. Esto constituye un enfoque distinto de diseño incluso con el paradigma de desarrollo basado en componentes, donde el enfoque de diseño puede darse más en aspectos técnicos de la reutilización de software que en aspectos del Negocio en si mismo, y el desacoplamiento muchas veces puede no ser el deseado.

Se debe aclarar, que si bien en las clasificaciones presentadas no se mencionan los servicios de infraestructura informática para la construcción de sistemas, si se mencionan como parte de la infraestructura tecnológica que deben proveer las plataformas, por ejemplo para manejo de la seguridad, sesión, loggings, auditoría, entre otros, y que generalmente son provistos por el bus de servicios, el middleware que permite las comunicaciones entre los distintos componentes de SOA.

Existen también más clasificaciones de servicios que no se presentan en este trabajo. Si bien la definición de la metodología no establece la utilización de una clasificación en especial, si se indica que los servicios deben ser identificados y categorizados, brindandose como guía de clasificación la presentada en [KRAF05]. Esto se debe a que se consideró que la misma resulta más simple y fácil de utilizar para servir como guía en la categorización de servicios.

La clasificación presentada en [ERLT05] fue liberada sobre la fecha en que se probaría la metodología en el curso "Proyecto de Ingeniería de Software", esto es agosto de 2005, por lo que se consideró mejor no incluir una nueva opción de clasificación de servicios para no dificultar la prueba de la misma con elecciones de este tipo. Luego de los ajustes realizados, para una aplicación posterior de la metodología será posible brindar como guías ambas clasificaciones de forma de permitir la elección de una u otra clasificación, según el criterio de la Organización.

4.2.3.3. Procesos del Negocio, Orquestación y Coreografías de servicios

En las secciones anteriores se mencionó brevemente como los procesos del negocio pueden ser representados y expresados a través de servicios de procesos o centrados en procesos. Según [ERLT05] "particionar la lógica del negocio en servicios que pueden componerse tiene implicaciones significativas sobre como se pueden modelar los procesos del negocio. Los analistas pueden influenciar estas características incorporando una extensión de orientación a servicios a los procesos del negocio para su implementación mediante SOAs. Los servicios además pueden ser diseñados para expresar lógica del negocio. Modelos de Business Process Management (BPM), de entidad, y otras formas de inteligencia del negocio pueden ser representadas con precisión a través de la composición coordinada de servicios centrados en el negocio. Esta es un área de SOA que todavía no está ampliamente aceptada o comprendida, el paradigma de modelado del negocio con orientación a servicios."

Con la capa de servicios intermedia, que abstrae tanto los procesos del negocio como las tecnologías por debajo, ambas partes de la Organización pueden evolucionar en forma independiente, más rápidamente y con menor impacto de los cambios originados en una en la otra. Como indica [ERLT05] "es la mutua abstracción del negocio y la tecnología, lo que soporta el paradigma de modelado del negocio con orientación a servicios y establece el modelo empresarial de bajo acoplamiento. ... En una Organización donde los principios de orientación a servicios se aplican tanto en el diseño técnico como el modelado del negocio, el concepto de bajo acoplamiento se amplifica. Implementando capas de abstracción de servicios estandarizadas, la relación de bajo acoplamiento puede ser también alcanzada entre los dominios del negocio y de tecnología de la aplicación. Cada uno requiere solo consciencia del otro, permitiendo de esa forma que cada dominio pueda evolucionar de forma más independiente. El resultado es un ambiente que puede absorver mejor los cambios en el negocio y en la tecnología, una cualidad conocida como agilidad organizacional." En la figura 4.8 se muestran estos conceptos.

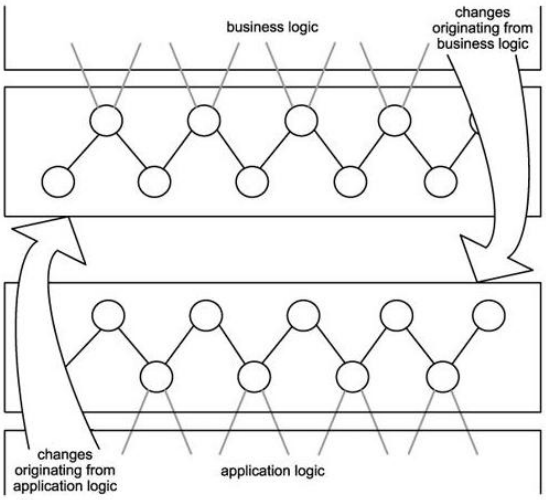


Figura 4.8: Agilidad organizacional con capa de servicios intermedia en SOA

La agilidad organizacional se ve incrementada con el uso de orquestaciones de servicios para modelar los procesos del Negocio con servicios centrados en procesos. Una orquestación es una forma de composición de servicios que permite definir procesos del Negocio específicos para un proyecto, mediante una secuencia definida de invocaciones a servicios existentes.

Según [ERLT05] "Una orquestación expresa el cuerpo de la lógica de un proceso del Negocio que es típicamente propiedad de una única Organización. Establece un protocolo del negocio que define formalmente la definición de un proceso del Negocio. La lógica del workflow en una orquestación es separada en una serie de actividades básicas y estructuradas que pueden ser organizadas en secuencias y flujos. ... El alcance de una única orquestación, entonces, puede ser clasificada como una actividad compleja, y posiblemente, de larga duración. ... La orquestación provee un modelo de automatización donde la lógica del proceso aunque centralizada, es aún extensible y componible. Mediante el uso de orquestaciones, los ambientes de soluciones orientadas a servicios se vuelven inherentemente extensibles y adaptables. Las orquestaciones en si mismas tipicamente establecen un punto común de integración para otras aplicaciones, lo que hace que una orquestación implementada sea un punto clave para permitir la integración."

Por otro lado existe también otro tipo de composición de servicios cuyo alcance va más allá de una única Organización, permitiendo la interacción entre diversas Organizaciones para realizar procesos del Negocio inter-organizacionales. En este caso el control de la secuencia de invocaciones a servicios no es propiedad de ninguna de las Organizaciones participantes en la coreografía.

Según [ERLT05] "El objetivo (de una coreografía) es establecer un tipo de colaboración organizada entre servicios que representan distintas entidades de servicios, solo que ninguna entidad (u organización) necesariamente controla la lógica de la colaboración. Las coreografías entonces proveen el potencial para establecer patrones de interoperabilidad universal para tareas del negocio comunes inter-organizacionales. ... En cualquier coreografía dada un servicio asume uno de varios roles predefinidos. Esto establece lo que el servicio hace y lo que el servicio puede hacer en el contexto de una tarea del negocio particular. Cada acción que es mapeada en una coreografía puede ser descompuesta en una serie de intercambios de mensajes entre dos servicios. Cada intercambio potencial entre dos roles en una coreografía es definido en forma individual como una relacion. Cada relación entonces consiste de exactamente dos roles. La naturaleza de las conversaciones están definidas por las características del intercambio de mensajes entre dos roles específicos por los canales de comunicación. ... Dos servicios que pertenecen a dos organizaciones distintas, cada uno exponiendo funcionalidad de la solución del negocio de la empresa entera, pueden interactuar vía una coreografía básica para completar una tarea compleja."

Tanto las orquestaciones como las coreografías son formas de composición de servicios que permiten realizar procesos del Negocio dentro de una Organización o entre varias Organizaciones. Según [ERLT05] "Mientras ambas representan patrones de intercambio de mensajes complejos, hay una distinción común que separa los términos orquestación y coreografía. Una orquestación expresa un flujo del negocio específico a una organización. Esto significa que la organización es dueña y controla la lógica detrás de la orquestación, incluso si esa lógica involucre interacciones con socios de negocio externos. Una coreografía, por otro lado, no necesariamente es propiedad de una única entidad. Actúa como un patrón de intercambio de la comunidad utilizado para propósitos colaborativos por servicios de distintas entidades proveedoras."

4.2.3.4. Niveles de realización de SOA

Para que una Organización actual como las mencionadas, que cuenta con varios sistemas separados en tecnologías heterogéneas, incorpore el enfoque SOA, se hace necesario realizar en forma previa una evaluación de la Organización y una planificación de como será la incorporación de los distintos elementos que componen SOA, en forma gradual, como forma de incorporar el cambio con riesgos controlados.

En [KRAF05] se propone un "architectural roadmap" o itinerario arquitectónico que sirve como base para tener en cuenta las distintas etapas por las que debería transitar una Organización para incorporar SOA incrementalmente en forma segura. Las etapas identificadas indican distintos niveles de madurez de una SOA, desde el primer nivel de *"fundamental SOA"*, pasando por el de *"networked SOA"* hasta llegar al de *"process-enabled SOA"* que incluye la orquestación de servicios para la implementación de los procesos del Negocio en la Organización.

- Fundamental SOA: este nivel consiste únicamente de dos capas de abstracción, la capa básica compuesta por servicios básicos y la capa empresarial donde se encuentran las application frontend. Esta distinción ayuda a que las aplicaciones definan una estructura de alto nivel adecuada, se identifican, definen e implementan servicios básicos del negocio permitiendo que dos o más aplicaciones compartan la lógica de negocio y los datos asociados. Aunque es un enfoque bastante simple provee las bases para ir avanzando hacia una SOA de mayor nivel.
- Networked SOA: en este nivel se agrega la capa de abstracción intermediaria que se compone de servicios intermediarios, que pueden incluir servicios del tipo facades, technology gateways, adapters y funcionality-adding. Como se mencionó, los servicios intermediarios encapsulan parte de la complejidad de los servicios básicos, y permiten la integración de conjuntos de software en forma independiente de la tecnología. Parte de la complejidad de las application frontend es delegada a los servicios intermediarios.
- Process-enabled SOA: el nivel de process-enabled SOA es el nivel más completo de SOA, donde los procesos del Negocio se realizan con servicios centrados en procesos que orquestan servicios de las capas por debajo. Agrega la capa de abstracción de procesos que se compone de servicios centrados en procesos que encapsulan la complejidad de los procesos del Negocio, comparten el estado entre múltiples clientes y manejan procesos de larga duración. Las application frontend pasan a ser más simples ya que la complejidad del Negocio es manejada por estos servicios. La complejidad de las aplicaciones en el backend es manejada por los servicios intermediarios y los servicios centrados en procesos. La lógica de los procesos del Negocio queda claramente separada de otro tipo de lógica y código específico que es manejada por los servicios por debajo.

En [ERLT05] no se proponen niveles de SOA, sino que se plantea como realizar las etapas del ciclo de vida de creación hasta puesta en producción de los servicios, según distintos enfoques. El objetivo principal que se plantea es "soportar la transición hacia una SOA completa y estandarizada, cumpliendo a la vez con requerimientos inmediatos específicos a distintos proyectos."

Para esto se plantea utilizar enfoque top-down, bottom-up y ágil o "intermedio", a pesar de su nombre este último no tiene que ver con enfoques de desarrollo ágil.

- Enfoque top-down: esta estrategia promueve la definición formal de modelos del negocio corporativos antes de modelar los servicios. Es un enfoque de "primero análisis" que requiere que no solo los procesos del Negocio se vuelvan orientados a servicios, sino que también promueve la creación (o realineamiento) del modelo del negocio entero de una Organización. Este proceso está muy relacionado o derivado desde la lógica del negocio existente en la Organización. Soporta la creación de las tres capas de servicios presentadas, y es común que resulte en la creación de numerosos servicios del negocio y la aplicación reusables. Puede resultar en el nivel más alto de calidad de SOA pero también impone un volumen significativo de trabajo de análisis previo.
- Enfoque bottom-up: esta estrategia se basa en la liberación de servicios de aplicación según sea necesario. Esencialmente fomenta la creación de servicios como forma de cumplir con requerimientos centrados en las aplicaciones. Los servicios son construidos según sea necesario y modelados para encapsular lógica de aplicación para servir mejor a los requerimientos inmediatos de la solución. La integración es el principal motivador de diseños bottom-up, donde la necesidad de sacar ventaja de algunas propiedades de SOA pueden ser alcanzadas simplemente introduciendo servicios wrappers sobre sistemas legados. Este enfoque es fácil de seguir pero no resulta en una SOA avanzada con todas sus propiedades.
- Enfoque ágil o intermedio: propone una combinación de los enfoques top-down y bottom-up. El desafío consiste en encontrar un balance aceptable entre la incorporación de principios de diseño orientado a servicios en los ambientes de ánalisis del negocio sin tener que esperar por aspectos tecnológicos asociados. Para muchas organizaciones puede ser útil contar con un enfoque intermedio entre los otros dos extremos que se proponen. Esto es posible definiendo un nuevo proceso que permite que el ánalisis a nivel del negocio se realice en forma concurrente con el diseño y desarrollo de servicios. Por lo tanto, soporta el análisis continuo a la vez que permite la liberación inmediata de servicios. A medida que el análisis progresa, los servicios existentes son revisados cuando se requiera.

4.2.3.5. Aspectos técnicos de la realización de SOA

Existen varios aspectos técnicos de SOA que deben ser tenidos en cuenta también al momento de incursionar en la puesta en práctica del enfoque. Principalmente asociados a la definición, implementación y distribución de servicios, comunicación entre estos (mensajería), incluyendo confiabilidad, forma y protocolos a utilizar, notificación, manejo de eventos, publicación y descubrimiento de servicios, integridad transaccional, compensaciones, seguridad, entre otros. Estos aspectos además tienen distintas características según sea la implementación de SOA que se elija. A continuación se mencionan brevemente aspectos de la modelización con Service Component Architecture (SCA) [SCAS] e implementación con Web Services (WS) [WSSP] y luego se describe brevemente el estado de herramientas asociadas.

Modelización con Service Component Architecture (SCA)

Como se define en [SCA05] "SCA es una especificación que describe un modelo para construir aplicaciones y sistemas utilizando Service Oriented Architecture (SOA) ... Mientras los sistemas basados en SOA pueden tener servicios individuales implementados utilizando tecnología orientada a objetos (entre otros enfoques), el diseño global del sistema es orientado a servicios ... SCA está basado en una especificación abierta, permitiendo que mútliples vendedores implementen el soporte para SCA en sus herramientas de desarrollo y ejecución. Esto es particularmente importante para el despliegue, administración y configuración de las aplicaciones basadas en SCA."

Entre las principales características del modelo que plantea SCA se encuentra que permite una variedad de implementaciones de componentes y tipos de interfaces. Por ejemplo, un componente SCA podría ser un proceso BPEL, y su interface estar definida en un documento WSDL, o podría ser una clase Java con su interface definida como una interface Java. La idea detrás del modelo es proveer al Negocio con la flexibilidad de incorporar una amplia gama de activos existentes y futuros en un sistema basado en SCA con poco o ningún código intermedio requerido. [SCA05]

SCA complementa la implementación con Web Services, proveyendo una forma para empaquetar servicios en un sistema del negocio, así como un modelo de construcción de servicios. Sin embargo, como se indica en [SCA05] "SCA reconoce que utilizar Web Services no es la única forma de implementar una SOA por lo que SCA soporta una amplia gama de tecnologías. Las interfaces de los servicios pueden ser definidas con WSDL e interfaces Java, y será expandida a otros lenguajes de interfaces."

Sobre la construcción de los componentes en SCA [SCA05] agrega "Los componentes SCA pueden ser construidos con una variedad de tecnologías como EJBs [EJB], Spring beans [SPRI] y componentes CORBA [CORB], y lenguajes de programación incluyendo Java [JAVA], PHP [PHP] y C++ [C++]. También están evolucionando enfoques centrados en XML para la construcción de componentes de servicios, incluyendo BPEL [WS-BP], XSLT [XSLT] y XQuery [XQUE]. Los componentes SCA pueden ser conectados también mediante una variedad de ligamientos como WSDL [WSDL]/SOAP [SOAP] para Web Services, Java Message Service (JMS) [JMS] para middleware con orientación a mensajes, y J2EE Connector Architecture (JCA) [JCA] para servicios de información empresarial."

SCA divide la implementación en dos pasos principales, primero la implementación de componentes que proveen servicios y consumen otros servicios, segundo el ensamblaje (assembly) de componentes para construir la aplicación mediante la conexión de referencias a servicios con los servicios. Provee también una forma de empaquetar y desplegar conjuntos de componentes relacionados, como una unidad. Una implementación provee un servicio que es la implementación de una interface, y puede utilizar otros servicios mediante referencias a servicios.

El ensamblaje de componentes se da en dos niveles: componentes relacionados en módulos, que pueden a su vez formar subsistemas, y el armado del sistema que puede componerse de varios subsistemas. Los subsistemas también pueden componerse únicamente de referencias a servicios, lo que permite cambiar también estas referencias. En la figura 4.9 se muestra la composición de un sistema con SCA.

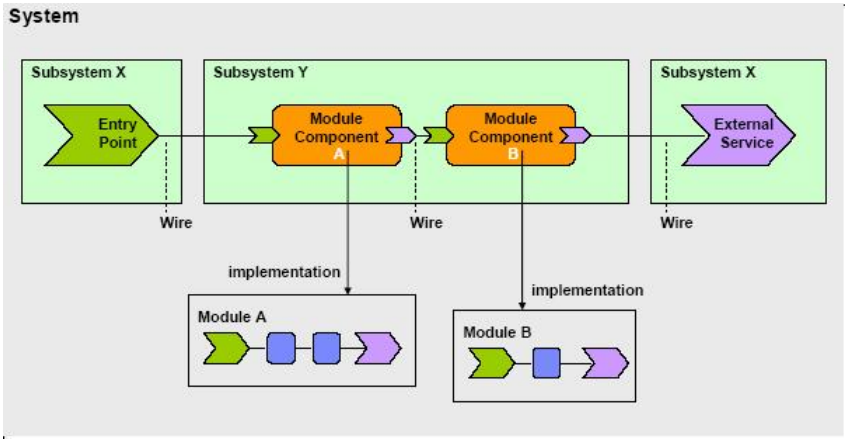


Figura 4.9: Modelo de implementación con Service Component Architecture (SCA) de [SCA05]

Como se puede ver en la figura 4.9 se definen puntos de entrada (entry points) que representan los servicios que el componente, módulo, subsistema o sistema ofrece para ser invocados, y

referencias a servicios que representan los servicios fuera del componente, módulo, subsistema o sistema que se utilizan. La definición de los elementos en SCA se realiza mediante documentos XML que tienen la estructura definida en la especificación de SCA, el modelo plantea también la utilización de Service Data Objects (SDO) [SDOS] para el intercambio de datos de manera uniforme entre servicios, encapsulando la forma de acceso a los mismos, que consiste también en documentos XML que definen los datos y la forma de obtenerlos.

Como aspectos principales, SCA soporta distintos tipos de ligamientos a servicios, que pueden ser Web Services, mensajería JMS, EJB de sesión sin estado, procedimientos almacenados de bases de datos, entre otros. Provee soporte para comunicación sincrónica, asincrónica, conversacional, también para la definición de políticas por ejemplo de seguridad, y conforma el estándar de definición de políticas WS-Policity [WS-PO]. Una descripción completa del modelo puede verse en [SCA05].

Implementación con Web Services (WS)

Actualmente una forma para publicar y acceder servicios ampliamente aceptada y utilizada por las Organizaciones, y soportada por la mayoría de los kits de herramientas de desarrollo, son los Web Services. La definición de Web Services provista en [W3C] establece que "Un Web Service es un sistema de software diseñado para soportar interacción interoperable de máquina a máquina sobre una red. Tiene una interface descrita en formato procesable por máquina (especificamente WSDL). Otros sistemas interactúan con el Web Service en una forma que se prescribe en su descripción utilizando mensajes SOAP, tipicamente utilizando HTTP con una serialización XML en conjunción con otros estándares Web relacionados."

Aunque no es la única forma para implementar servicios, es una forma que brinda mayor interoperabilidad debido a la estandarización de los protocolos para el diseño y la comunicación entre los Web Services, lo que resulta muy importante principalmente para las relaciones inter-organizacionales, ya que utilizando Web Services es posible comunicar software implementado en distintas plataformas.

La implementación con Web Services implica la definición de los elementos involucrados en la orientación a servicios para dicha implementación, para los cuales existen estándares definidos que permiten que las distintas organizaciones que proveean y consuman servicios utilizando dichos estándares puedan interoperar mediante servicios para realizar sus procesos de negocio. Como se establece en [SING04] dichos estándares cubren las siguientes áreas:

- Lenguaje de marcas común para comunicación: un lenguaje de marcas común facilita la comunicación entre proveedores y consumidores, ya que cada parte puede leer y comprender la información intercambiada basada en las marcas embebidas. Los Web Services utilizan el *eXtensible Markup Language (XML)* [XML] como lenguaje de marcas común y *XML Schema* [XMLS] para definición de la estructura de los documentos XML intercambiados.
- Formato de mensajes común para intercambio de información: para la comunicación las partes deben acordar el intercambio de mensajes basado en un formato acordado. Teniendo ese formato, las partes aunque sean desconocidas, pueden comunicarse en forma efectiva. El formato común para mensajes utilizado por los Web Services es el *Simple Object Access Protocol (SOAP)* [SOAP].
- Formato de especificación de servicios común: para describir los detalles de un servicio, como el tipo de servicio, como accederlo, y otros, se debe utilizar un mecanismo estándar para que los proveedores especifiquen los servicios y los consumidores puedan entenderlo y utilizarlos. El *Web Services Description Language (WSDL)* [WSDL] provee a los Web Services un formato de especificación común.
- Forma común para búsqueda de servicios: para descubrir y obtener los detalles de un servicio, los consumidores deben tener una forma común de hacerlo. Esto se logra teniendo ubicaciones comunes bien conocidas donde los proveedores registran sus servicios y los

consumidores los buscan. De esta forma, los servicios pueden ser accedidos universalmente por los proveedores y consumidores. La especificación *Universal Description, Discovery and Integration (UDDI)* [UDDI] define una forma común de búsqueda de Web Services.

En la figura 4.10 se muestra la relación entre los distintos elementos que componen el marco de referencia para implementación con Web Services:

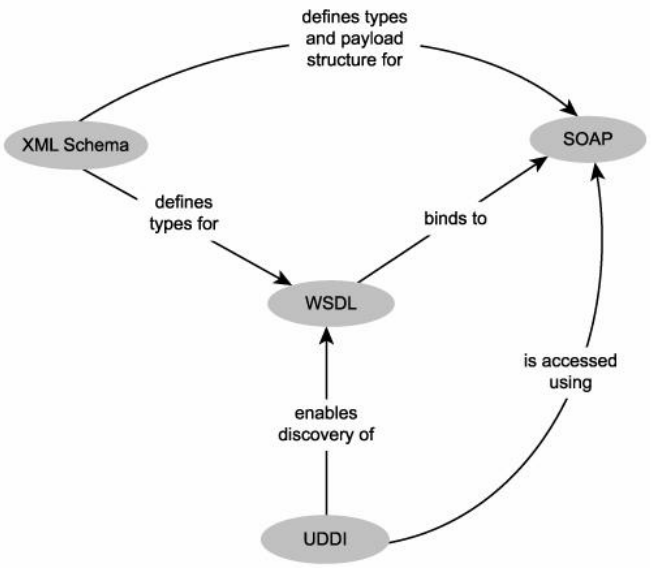


Figura 4.10: Relación entre los elementos del marco de referencia de implementación con Web Services de [ERLT05]

Existen otros esfuerzos de estandarización del uso de Web Services como el de Web Services Interoperability Organization (WS-I) [WS-I] que es un grupo formado por organizaciones que cruzan varias industrias creado para promover la interoperabilidad de Web Services a través de distintas plataformas, sistemas operativos y lenguajes de programación . WS-I ha publicado el WS-I Basic Profile [WS-IB] que agrupa un conjunto de los estándares mencionados para desarrollar soluciones con Web Services interoperables, estableciendo restricciones y clarificaciones sobre el uso de dichos estándares para evitar problemas de interoperabilidad por distintos usos o interpretaciones de los estándares, que fue extendido también para incluir archivos adjuntos en los mensajes SOAP en el WS-I Attachments Profile [WS-IA].

Aspectos importantes a tener en cuenta en la implementación con Web Services incluyen patrones de intercambio de mensajes (sincrónicos, asincrónicos), transacciones atómicas, direccionamiento, mensajería confiable, correlación de mensajes, definición de políticas, intercambio de metadata, seguridad, notificación y manejo de eventos, entre otros. Estos aspectos son tenidos en cuenta en otros estándares conocidos como de extensión de Web Services, cuya notación incluye en todos los casos el prefijo WS- por lo que son conocidos como los estándares WS-*. Estas son extensiones al framework básico de Web Services establecido por la primera generación de estándares representados por WSDL, SOAP y UDDI. Entre ellos se pueden destacar el WS-BPEL [WS-BP] para orquestaciones, WS-CDL [WS-CD] para coreografías, WS-Security [WS-SE] para seguridad de Web Services, entre otros. El detalle de estos estándares, sus especificación y utilización puede verse en las referencias mencionadas.

Herramientas asociadas a SOA

En general la mayoría de los grandes proveedores de herramientas de desarrollo y software de base, tanto comerciales como open source, han incorporado elementos relacionados con la inclusión de Web Services con capacidades de generación de los mismos, y definición y ejecución

de procesos del Negocio con modelado desde los IDE de desarrollo, generación de BPEL y ejecución en motores de procesos. Como principales herramientas se pueden mencionar las incluidas en IBM SOA Foundation [SOAF05], Oracle SOA Suite [SOAO06], Microsoft SOA Platform [SOAM06], Sun Java CAPS [SOAC06], Jboss SOA Platform [SOAJ06] y Eclipse SOA Tools Platform (STP) [**?**]. La evaluación de dichas herramientas excede el foco de este trabajo por lo que los detalles sobre las mismas pueden consultarse en la bibliografía mencionada.

4.2.4. Enfoques para desarrollo SOA

En esta sección se presentan enfoques existentes para desarrollo SOA, que fueron tomados como base para la definición de la metodología SOA propuesta que se presenta en el capítulo 5, así como para la mejora de la misma una vez realizada la prueba asociada que se presenta en el capítulo 6. En primer lugar en 4.2.4.1 se presenta el enfoque SOA planteado en [ENDR04], del cual se tomaron en cuenta algunos de los conceptos planteados para la definición de la metodología SOA propuesta; en segundo lugar en 4.2.4.2 se presenta el enfoque SOA planteado en [KRAF05] que constituyó la base de los conceptos utilizados en todo el trabajo de tesis, y finalmente en 4.2.4.3 se presenta el enfoque SOA planteado en [ERLT05] del cual se tomaron conceptos principalmente para el ajuste de la metodología SOA propuesta.

4.2.4.1. Enfoque SOA de [ENDR04]

En el Capítulo 4 de [ENDR04] se propone un enfoque SOA que "consiste en siete pasos, donde las actividades no son necesariamente lineales y secuenciales, sino más bien exploratorias e iterativas, incrementando la funcionalidad y comprensión del equipo a medida que el proyecto avanza." Este enfoque combina los patrones de negocio de IBM con los conceptos SOA, y puede ser utilizado con el proceso de desarrollo existente en la Organización, ya que provee agregados para el diseño de SOA, incluyendo artefactos a obtener. Propone un enfoque top-down para descubrimiento de servicios, y bottom-up para obtención de servicios de sistemas legados y aplicaciones empaquetadas. Utiliza como el RUP Casos de Uso del Negocio para modelar los procesos del Negocio. En la figura 4.11 se muestran los siete pasos que componen el enfoque propuesto.

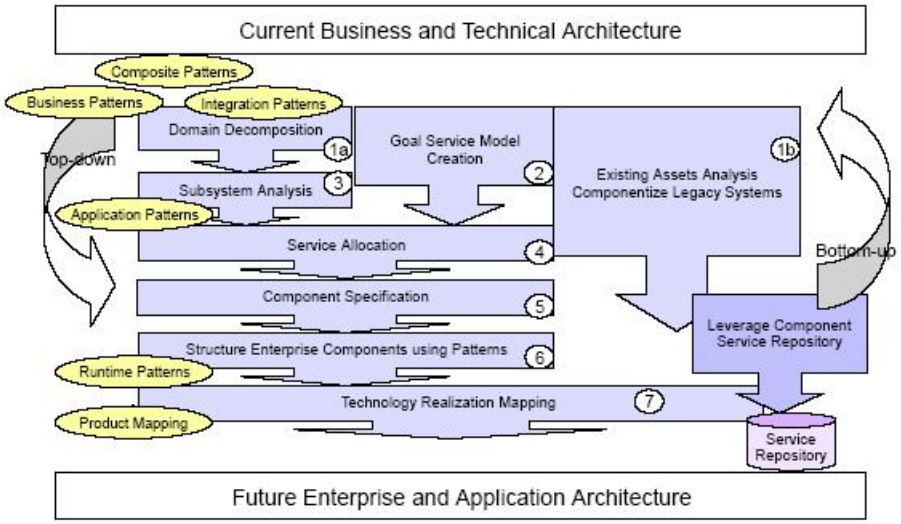


Figura 4.11: Pasos del enfoque SOA propuesto en [ENDR04]

1. Descomposición del dominio: en primer lugar se descompone el dominio en su cadena de valor de áreas funcionales, luego cada área funcional se descompone en procesos, subpro-

cesos y Casos de Uso del Negocio. Estos últimos se utilizan para refinar la descomposición del dominio realizada, ubicandolos en sus áreas funcionales. Luego cada área funcional es mapeada a uno o más subsistemas en la Arquitectura, y los Casos de Uso del Negocio son mapeados a Casos de Uso del Sistema. Luego se aplican patrones del Negocio y de Integración de aplicaciones, que clasifican y definen formas de interacción entre los elementos identificados.

2. Creación de un modelo objetivos-servicios: plantean que los Casos de Uso del Negocio son buenos candidatos a ser servicios, por lo que la identificación de servicios objetivos parte de éstos. Se identifican los objetivos y sub-objetivos que deben ser cumplidos para alcanzar objetivos de más alto nivel del negocio, asociando los sub-objetivos con servicios requeridos para obtenerlos.

3. Análisis de subsistemas: refina los Casos de Uso del Negocio en Casos de uso del Sistema que soportan un proceso del Negocio dado. Se identifican componentes del Negocio y técnicos analizando los flujos de los procesos entre subsistemas para descubrir/identificar componentes del negocio candidatos. Se utilizan los requerimientos no funcionales para encontrar componentes técnicos. Se identifica la funcionalidad requerida para cada componente del negocio, esto es los Casos de Uso del sistema que cada componente debe soportar. Luego se aplican patrones de aplicación para estructurar los servicios del negocio y técnicos necesarios.

4. Asignación de servicios: asegura que todos los servicios identificados tengan un componente de implementación asignado. Estos componentes pueden ser creados nuevos o pueden existir como funcionalidad implementada pero no en forma de componente.

5. Especificación de componentes: para cada componente identificado se especifican propiedades como reglas, servicios, atributos, otros componentes que usa, puntos de variación por ejemplo requerimientos de configuración.

6. Estructuración de componentes y servicios utilizando patrones: se aplican patrones de tiempo de ejecución del negocio, para establecer la estructura necesaria del middleware para soportar los servicios identificados y la asignación de cada servicio a un nodo lógico del middleware.

7. Mapeo de realización tecnológica: se definen los mecanismos de implementación adecuados para los servicios y componentes especificados. Esto refiere tanto a decisiones del tipo realizar el desarrollo desde cero o hacer outsourcing como solución llave en mano, o intermedio de análisis "implementar, reutilizar, comprar" ya visto, y alternativas que pueden combinar estas opciones, y también refiere a elecciones del tipo, implementar un wrapper de un sistema legado con un Web Service o con un servicio de cola de mensajes. Luego es posible realizar un análisis de ventajas y desventajas de las decisiones tomadas, y aplicar mapeos de productos para seleccionar software de base e infraestructura necesaria y cubrimiento de las necesidades por cada vendedor.

El enfoque presentado plantea siete pasos para realizar un desarrollo SOA enfocado en el diseño de servicios a partir de procesos del Negocio identificados en la Organización. Utiliza como el RUP Casos de Uso del Negocio para describir los procesos del Negocio que luego son mapeados a Casos de Uso del Sistema. Sin embargo no se provee una guía de como realizar este mapeo ni de la relación y diferencias entre los dos tipos de Casos de Uso. Los pasos que se proponen no se identifican como parte de Disciplinas de un proceso de desarrollo, ni se describen las entradas, salidas y roles asociados a cada uno, lo que es más, no se menciona en ninguno de los pasos que roles intervienen ni con que responsabilidades.

Plantea que los Casos de Uso del Negocio son buenos candidatos para servicios, pero esto sería más bien aplicable a servicios centrados en procesos únicamente, según la teoría presentada, ya que los servicios que luego se orquesten para realizarlos, debieran ser de menor granularidad o

alcance. No se propone una clasificación de servicios ni distinto enfoque ni granularidad para los mismos, lo que como se vió previamente, ayuda a la definición asociada, sino que los servicios se derivan directamente de las áreas funcionales y los procesos asociados. No se indica tampoco como se relaciona la definición de servicios con la definición de la Arquitectura de Software del sistema, aunque ésta se nombra en el análisis de subsistemas como parte de las decisiones de diseño. La aplicación de distintos patrones del negocio si bien es un enfoque interesante también puede agregar complejidad al desarrollo ya que se debe capacitar a los involucrados para su correcta aplicación.

En los artículos [ZIMM04] y [ARSA04] también de IBM, se presentan elementos del análisis y diseño orientado a servicios, así como descripción de los pasos del enfoque SOA planteado en [ENDR04], introduciendo los términos Service Oriented Analysis and Design (SOAD) y Service Oriented Modeling and Architecture (SOMA), que luego fue intregado en el plug-in para SOA que se agrega al RUP sobre mediados del año 2005 en su primera versión, cuya última actualización de noviembre de 2006 se comenta en el Capítulo 7. Uno de los elementos que agregan estos artículos es el hecho que se mencionaba como faltante en el enfoque de [ENDR04] sobre la clasificación o categorización de servicios.

Especificamente en [ARSA04] se plantea realizar esta actividad luego de identificar los servicios, estableciendo que "es importante comenzar la clasificación de servicios en una jerarquía de servicios, reflejando la naturaleza de composición de los servicios: éstos pueden y deben ser compuestos de servicios y componentes de granularidad más fina. La clasificación ayuda a determinar la composición y las capas de servicios, así como a la coordinación de la construcción de servicios interdependientes basada en la jerarquía." Se cambia el orden de los siete pasos presentados y se agrupan en tres pasos generales: Identificación, especificación y realización de servicios, componentes y flujos.

Se destacan como aportes del enfoque SOA y los aspectos presentados en [ENDR04], [ARSA04] y [ZIMM04] la conceptualización de los tres pasos generales de identificación, especificación y realización de servicios, la asignación de servicios que permite identificar los componentes que se deben implementar, y la utilización de Casos de Uso del Negocio para modelar los procesos del Negocio. Estas características fueron tenidas en cuenta para la propuesta de metodología SOA realizada, pero las actividades relativas al Modelado del Negocio fueron adaptadas de la Disciplina Modelado del Negocio original del RUP, donde están definidas en forma más completa y relacionadas con el resto de la propuesta de desarrollo que plantea el RUP.

4.2.4.2. Enfoque SOA de [KRAF05]

El enfoque presentado en [KRAF05] es una guía que no establece un proceso de desarrollo, sino aspectos a tener en cuenta en el agregado de la orientación a servicios en un desarrollo de software. El primer punto que plantea es referido a los procesos de desarrollo a los cuales es posible agregar el enfoque SOA, describiendo las similitudes y diferencias entre procesos "pesados" y "ágiles", como el RUP y XP presentados en 2.3.

La elección del proceso y metodología de desarrollo dependerá del contexto del proyecto, aspectos como si la Organización tiene o no instaurado un proceso de desarrollo, la complejidad, alcance y duración del proyecto, requerimientos de calidad, importancia estratégica, entre otros. Sin embargo, dada la naturaleza compleja y frecuencia de cambios en los requerimientos del Negocio en proyectos del tipo SOA, se recomienda que el enfoque elegido sea del tipo iterativo incremental, fundamental para tratar con la complejidad y con los requerimientos inestables. Además, dada la importancia estratégica de este tipo de desarrollos, es esperable que la metodología elegida tienda hacia el lado de los procesos "pesados", donde se puedan ir obteniendo varios entregables intermedios en los hitos establecidos.

Plantea incluir diseño de servicios en la definición del proyecto ya que son fundamentales tanto para la Arquitectura como para la planificación del proyecto. Un entregable clave de la fase de definición del proyecto debe ser una Arquitectura preliminar, en la que se hayan identificado

las application frontend, los servicios externos, y lo servicios básicos más importantes. Sobre estos servicios se debe distinguir los servicios que serán construidos desde cero, los servicios nuevos basados en aplicaciones existentes, extensiones y modificaciones a servicios existentes. El diseño de servicios intermediarios y centrados en procesos puede realizarse más adelante. Por lo que además de la identificación de servicios en el diseño se realiza también la clasificación o categorización de los mismos, como guía para la definición de los mismos.

Para el diseño e implementación se propone la aplicación del modelo "hilo fino" (thin thread) de desarrollo, que aplica el enfoque de desarrollo iterativo incremental de aplicaciones. Un hilo representa un corte vertical del sistema completamente funcional expandiendo todas las capas del sistema desde el frontend al backend. Los contratos de los servicios involucrados en un hilo fino definen el alcance y guían el desarrollo y el testing. Los hilos que se construyen con funcionalidad básica se van "engrosando" en fases siguientes hasta cumplir completamente con los requerimientos definidos para ese servicio, donde se van desarrollando distintos hilos en paralelo.

Para la gestión de proyecto propone basarse en los contratos de los servicios. Porque los servicios tienen el nivel ideal de granularidad y orientación a servicios para servir como base para guíar los aspectos más importantes de un proyecto, incluyendo: estimación de tiempos y costos, planificación de iteraciones y desarrollo, sincronización del desarrollo, planificación del testing e implantación. Para comenzar con la aplicación del enfoque SOA en una Organización es imprescindible contar con apoyo político de la gerencia. Es importante también instaurar un comité SOA o una institución similar para controlar y promover el uso de servicios como herramienta de gestión de proyectos.

Este comité SOA será responsable por la sincronización de servicios que son compartidos por varios proyectos y subproyectos. Mediante los contratos de servicio se establecen las interfaces entre las distintas capas de servicios y los datos, permitiendo el desarrollo en paralelo y la estabilidad de cada proyecto alrededor de las decisiones y compromisos tomados en la definición de estos contratos. En la figura 4.12 se muestra el enfoque SOA para una iteración.

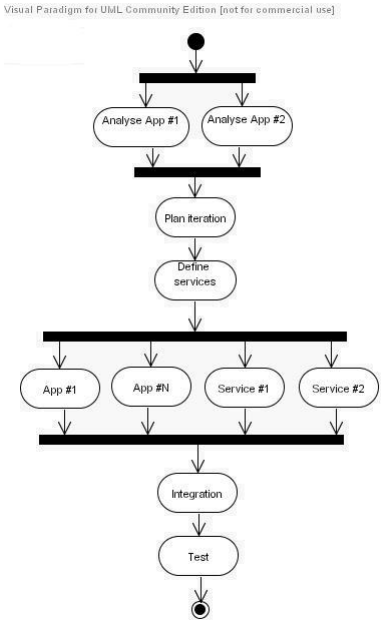


Figura 4.12: Enfoque SOA para una iteración de [KRAF05]

Para la gestión de configuración se recomienda mantener los servicios en diferentes proyectos dentro del repositorio de configuración, ya que éstos serán compartidos por varias aplicaciones,

por lo que es importante mantener su independencia de versiones también. Parece natural definir estos proyectos a partir de los servicios básicos que serán los más utilizados, a los que se pueden agregar algunos servicios intermedios que estén muy relacionados. Luego el resto de los servicios intermedios y los servicios centrados en procesos deberían estar también en proyectos separados, así como las application frontend específicas de cada proyecto. Las bibliotecas compartidas por varios proyectos se pueden "factorizar" y colocarlas agrupadas también en proyectos distintos.

Para el testing de servicios se propone que la unidad de testeo sea el servicio, realizando sobre éste tanto testeo funcional como de carga. En primer lugar para el testing funcional, testeo unitario sobre cada servicio para asegurar un primer cumplimiento de funcionalidades, luego testeo de integración sobre varios servicios que deban interactuar, y finalmente testeo del sistema de extremo a extremo, para lo cual es posible introducir herramientas que automaticen las pruebas funcionales, como el Rational Robot [RROB]. Para el testeo de carga se recomienda la introducción de herramientas como Jmeter [JMET] o Mercury [MERC], entre otras.

El enfoque presentado no plantea un proceso de desarrollo sino que brinda buenas prácticas para las actividades a realizar en las distintas áreas del mismo. Plantea, sin embargo, que el proceso más adecuado sobre el cual construir un desarrollo SOA es un proceso iterativo incremental con características del tipo de los procesos "pesados". Como guías interesantes propone realizar un diseño e implementación incremental de servicios, similar al enfoque del RUP, haciendo énfasis en que la definición de elementos de SOA debe estar incluida en la Arquitectura preliminar definida en las primeras Fases del desarrollo. Estos elementos deben ser clasificados según el enfoque de clasificación que se propone para los elementos de SOA y para las distintas clases de servicios.

Se destacan como aportes del enfoque SOA presentado en [KRAF05] las guías propuestas para la realización de las distintas actividades en el proceso de desarrollo, principalmente las que tienen que ver con los aspectos de diseño de servicios incluyendo la clasificación provista, fueron tenidos en cuenta para la metodología SOA propuesta. La clasificación de elementos de SOA provista también fue brindada como guía básica para la comprensión y apoyo a la realización de las actividades definidas en la metodología SOA propuesta.

Aspectos planteados para las Disciplinas de Gestión de la Configuración, Gestión del Proyecto y Testing fueron tenidos en cuenta para la mejora de la metodología realizada, para incluir también el enfoque a servicios en dichas disciplinas. Principalmente en cuanto al enfoque de la estructura para el repositorio de gestión de configuración basado en proyectos por servicios, la gestión del proyecto por iteraciones según los servicios a construir en cada una, y la utilización de servicios como unidad de testeo para el testing unitario, de integración y de sistema, así como la generación de tests de regresión y la puesta en configuración de todos los artefactos de testing asociados a cada versión.

4.2.4.3. Enfoque SOA de [ERLT05]

El enfoque que se presenta en [ERLT05] plantea que "proyectos de desarrollo para soluciones orientadas a servicios, en la superficie, son similares a proyectos de desarrollo para aplicaciones distribuidas. ... Cuando nos adentramos en las capas de la orientación a servicios, sin embargo, encontramos que para construir y posicionar apropiadamente los servicios como parte de SOA, los ciclos de proyectos tradicionales requieren algunos ajustes." Se propone un proceso de desarrollo que consta de seis fases que, sin embargo, recuerdan los procesos de desarrollo tradicionales "en cascada". En la figura 4.13 se presentan las fases propuestas.

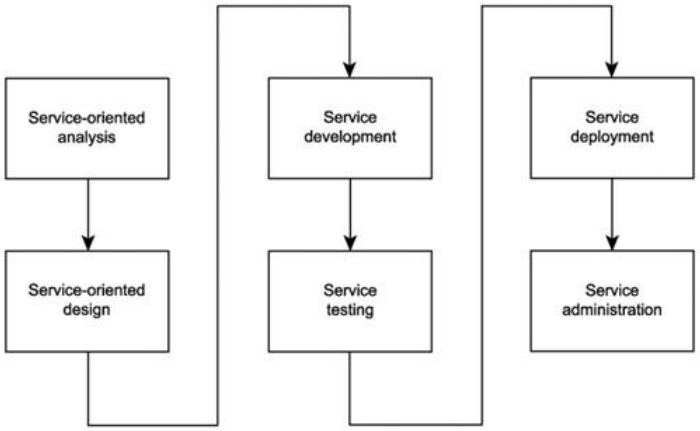


Figura 4.13: Fases del proceso de desarrollo propuesto en [ERLT05]

En la fase de análisis orientado a servicios los objetivos principales son establecer las capas de servicios, y modelar servicios individuales como servicios candidatos que componen una SOA preliminar. Como objetivos específicos se plantean la definición de un conjunto preliminar de operaciones de servicios candidatas, la agrupación de estas operaciones candidatas en contextos lógicos que representarán los servicios candidatos, la definición de los límites preliminares de los servicios para que no se superpongan con servicios existentes o planificados, la identificación de lógica encapsulada con potencial de reuso, la evaluación de que el contexto de la lógica encapsulada es apropiada para el uso definido, y la definición de modelos de composición preliminares conocidos. La fase de análisis se compone de tres pasos:

1. definir los requerimientos de automatización del negocio: los requerimientos del negocio deben permitir la definición de un proceso de automatización de alto nivel. La documentación de este proceso del negocio será usada como punto de entrada para el proceso de modelado de servicios del paso 3.

2. identificar sistemas de automatización existentes: la lógica de aplicación que ya se encuentra, de alguna manera, automatizando cualquiera de los requerimientos identificados en el paso 1 debe ser identificada. En general este paso tiende a soportar esfuerzos de modelado de soluciones orientadas a servicios de larga escala.

3. modelar servicios candidatos: en este paso se identifican operaciones candidatas que se agrupan en contextos lógicos que pueden dar lugar a servicios candidatos. La razón de distinguir que son candidatos es debido a que luego en el diseño son sujetos a las realidades de la arquitectura técnica en la que deben residir. Al incluir restricciones, requerimientos, limitaciones específicas al ambiente de implementación, el diseño final de un servicio puede ser bastante distinto del candidato original. Este es el paso más importante en la fase de análisis y se divide a su vez en doce subpasos que guían el proceso de análisis hacia sus objetivos.

En la Fase de Diseño se definen los siguientes objetivos: determinar el conjunto núcleo de extensiones arquitectónicas, establecer los límites de la arquitectura, identificar los estándares de diseño requeridos, definir el diseño de las interfaces abstractas de servicios, identificar las composiciones de servicios potenciales, evaluar el soporte a principios de orientación a servicios y explorar soporte para características tecnológicas de la implementación de SOA (especificamente con Web Services). La fase de diseño se compone de cinco pasos, los que a su vez se dividen en subpasos para alcanzar los objetivos planteados:

1. componer SOA: se eligen las capas de servicios que serán utilizadas, así como los estándares y tecnologías para la implementación de los servicios (especificamente soporte para XML y WS), y extensiones que se utilizarán (especificamente de las especificaciones WS-* para seguridad, encriptación, transaccionalidad, entre otros)

2. diseñar servicios del negocio centrados en entidades: estos servicios representan la capa de servicios menos influenciada por otras. Debe representar las entidades de datos definidas en los modelos del negocio. Se diseñan para ser reutilizados por cualquier aplicación que necesite acceder o manejar información asociada con una entidad en particular.

3. diseñar servicios de aplicación: estos servicios representan servicios responsables de realizar procesamientos requeridos por las capas de negocio y orquestación. No se requiere conocimiento del negocio para su diseño, ya que son una abstracción pura orientada a servicios del ambiente tecnológico de una organización. Están sujetos a cambios constantes según las actualizaciones o cambios de la tecnología y la lógica relacionada es construida o alterada.

4. diseñar servicios del negocio centrados en tareas: el proceso para diseñar servicios centrados en tareas usualmente requieren menos esfuerzo que los dos anteriores, ya que el reuso no es una consideración primaria. Solamente las operaciones candidatas de servicios identificadas en el análisis se tratan en estos servicios.

5. diseñar procesos del Negocio orientados a servicios: consiste en interpretar apropiadamente los requerimientos de los procesos del negocio que se han recolectado e implementarlos acertadamente. Hay que tener en cuenta las posibles variaciones de las actividades de los procesos, esto significa entender no sólo que puede ir mal, sino como el proceso debe responder a condiciones no esperadas o anormales. En este paso se diseñan los detalles del workflow, así como la interface del servicio de procesos.

En la Fase de Desarrollo de servicios se introducen las características específicas de las plataformas, sin importar el tipo de servicio. Especificamente, la elección del lenguaje de programación y del ambiente de desarrollo determinarán la forma física que toman los servicios y los procesos del negocio orquestados, de acuerdo al diseño definido. Se incluyen ejemplos para plataformas .NET [MNET] y J2EE [J2EE].

Para la Fase de Testing de servicios se indica que dada la naturaleza genérica y el potencial para ser reutilizados y compuestos en distintas situaciones, se requiere que los servicios pasen por un testing riguroso antes de su pasaje al ambiente de producción. Sin embargo, no se introducen subpasos ni actividades, se brinda una checklist con preguntas para guiar el testing, como por ejemplo, que tipo de consumidores de servicios podrían potencialmente acceder a un servicio, entre otras.

En la Fase de Deployment de servicios se deben instalar y configurar los componentes distribuidos, las interfaces de servicios y cualquier otro producto de middleware asociado, en los servidores de producción. Se brinda una checklist de guía que incluye preguntas como por ejemplo, como serán distribuidos los servicios, y se incluyen ejemplos en las plataformas .NET [MNET] y J2EE [J2EE].

La Fase de Administración de servicios ocurre luego de que se ha realizado el deployment de servicios. Son similares en naturaleza a la administración de aplicaciones distribuidas, basadas en componentes, excepto que se deben aplicar a los servicios como un todo, en oposición a los servicios en un ambiente de aplicación específico. No se incluyen subpasos ni actividades específicas, y se brinda una pequeña checklist a modo de ejemplo sobre los temas a considerar.

El enfoque presentado consta de seis Fases que se realizan en forma secuencial, al estilo de los procesos tradicionales "en cascada". Si bien no plantea vueltas de feedback a las Fases anteriores, podría ser aplicado en forma iterativa incremental con algunas adaptaciones. No se indican roles intervinientes ni sus responsabilidades, ni tampoco los artefactos de entrada

y salida de las actividades, aunque al estar presentado para realizar en forma secuencial, se puede asumir que se utilizarán para las actividades siguientes, los productos realizados en las anteriores. No se brindan tampoco plantillas o templates para la realización de los productos a obtener.

En la Fase de análisis se menciona la necesidad de identificar los procesos del Negocio y los requerimientos del Negocio, pero no se propone como realizar su modelado, esto es con diagramas BPMN, diagramas de actividad de UML, Casos de Uso del Negocio, u otras formas. En la Fase de Diseño se nombra superficialmente la Arquitectura de Software sin indicar la relación de los servicios con ésta, ni la importancia en el enfoque. Además, también en la Fase de Diseño, las actividades están definidas sobre la clasificación de servicios que se propone, lo que hace el proceso muy específico. En la metodología SOA propuesta, por el contrario, las actividades de identificar y categorizar servicios, especificar servicios e investigar servicios existentes, pueden realizarse utilizando la clasificación de servicios con la que se esté más cómodo, ya que solo se referencian a modo de guía para facilitar la definición de servicios.

También se observa en la descripción de varias de las actividades, que si bien están basadas en conceptos de orientación a servicios, se introducen constantemente aspectos de la implementación con Web Services que también las hacen específicas a dicho enfoque, al contrario de la metodología propuesta que es también independiente del tipo de implementación elegida.

De las seis Fases que se definen, solo se describen en profundidad las de Análisis y Diseño, para la de Desarrollo se brindan ejemplos, pero para las últimas tres de Testing, Deployment y Administración, solo se define su objetivo. Este hecho es común en propuestas de este estilo, como también lo es en la propuesta inicial de la metodología SOA puesta en práctica, ya que el énfasis del enfoque está en el área de Diseño, y en este caso al estar como Fase separada, también en el Análisis. Cabe agregar que en el RUP y en el proceso base adaptación, el Análisis y Diseño son una sola Disciplina nombrada Análisis&Diseño.

Como ya se comentó, este libro fue editado al momento de poner en práctica la metodología SOA definida, cuyo énfasis principal está dado en la Disciplina de Diseño que es donde se incluyen las principales actividades orientadas a servicios, además de incluir la Disciplina de Modelado del Negocio, que en esta propuesta no se incluye. Por lo tanto, si bien este enfoque no fue tenido en cuenta al momento de definir la metodología SOA propuesta, su posterior estudio permitió comparar aspectos del enfoque que también se plantean en este proceso, confirmando coincidencias e identificando diferencias, que permitieron tomar aportes para el ajuste y mejora de la metodología realizado, principalmente en lo que tiene que ver con el agregado de guías útiles para algunas actividades.

4.3. SOA y Business Process Modeling (BPM)

Como se vió previamente, uno de los aspectos principales de una SOA completa es que los procesos del Negocio se realicen como servicios centrados en procesos, que mediante orquestación de otros servicios de menor granularidad, ejecuten los flujos definidos para dichos procesos. [ERLT05] plantea que "Históricamente, los procesos del negocio eran diseñados por analistas utilizando herramientas de modelado que producían diagramas que se les entregaban a los arquitectos y desarrolladores para su implementación. Los diagramas de workflow y su documentación eran la única forma de comunicar como esta lógica debía ser realizada por la solución automatizada. Aunque un buen análisis y documentación, además de mentes abiertas y experticia técnica sobre el negocio, pueden llevar a una colaboración exitosa entre los miembros del equipo del negocio y técnicos, este enfoque deja lugar a errores."

Según [KRAF05] "Los procesos del Negocio en general son dinámicos en el sentido de que están propensos a cambios frecuentes y generalmente requieren una compleja coordinación con los participantes del proceso. ... tienen estado relacionado con el proceso en si mismo. Este estado del proceso incluye información sobre los participantes en el proceso (personas y servicios),

entradas de los participantes, la posición actual en el flujo del proceso, y reglas básicas. Los servicios centrados en procesos son altamente dependientes de otros servicios, en particular de aquellos servicios que representan la lógica núcleo del negocio." [KRAF05] agrega que "Aunque los servicios centrados en procesos pueden ser realizados de varias maneras distintas (y por lo tanto pueden tomar muchas formas distintas), Business Process Management (BPM) representa posiblemente el enfoque más consecuente para lograr una process-enabled SOA."

A continuación se presenta brevemente el enfoque Business Process Management (BPM), en 4.3.1 se introducen conceptos del enfoque, en 4.3.2 se presenta el modelado de procesos del negocio que propone, y finalmente en 4.3.3 se muestra como se incluye el enfoque en SOA. No es la intención en esta sección entrar en detalles del enfoque, sino presentar sus principales conceptos para poder establecer como y porqué se relaciona con SOA, y la importancia que tiene en SOA la inclusión de este enfoque.

4.3.1. Presentación de BPM

En [SHFP03] se introduce Business Process Management (BPM) planteando que "La visión de Business Process Management (BPM) no es nueva, sin embargo, las teorías y sistemas existentes no han podido lidiar con la realidad de los procesos del Negocio, hasta ahora. Al ubicarlos en el centro del escenario, las organizaciones pueden obtener las capacidades que necesitan para innovar, mejorar la performance y entregar el valor que el mercado actual demanda." BPM entonces "acompaña el descubrimiento, diseño y deployment de procesos del negocio, así como el control ejecutivo, administrativo y de supervisión sobre los mismos para asegurar que se mantienen conforme a los objetivos del negocio."

Sobre los antecedentes de BPM en las organizaciones [SHFP03] agrega que "Durante la ola de reingeniería de procesos de los años 90, diversos libros de gestión contaban experiencias de empresas para utilizar como guía de transformación del negocio. A pesar de que la teoría subyacente estaba basada en sentido común "de siempre" y teoría general de sistemas propuesta cincuenta años atrás, no se ofrecía un camino para su ejecución. En contraste, la organización que gestiona sus procesos, toma control de sus procesos internos y se comunica con un lenguaje de procesos universal que permite que los socios ejecuten una visión compartida, comprendiendo las operaciones de cada uno, conjuntando diseño de procesos y gestionando el ciclo de vida completo de sus iniciativas de mejora del negocio."

En [BPMI] se define BPM como el "conjunto de actividades que realizan las Organizaciones para optimizar o adaptar sus procesos de negocio a las nuevas necesidades organizacionales".

Para [KRAF05] "BPM introduce el concepto de "procesamiento de procesos" y enfatiza que este concepto no está limitado a la ejecución automática de modelos de procesos digitales" incluyendo la definición en [SHFP03].

Según [OWEN03] BPM tiene que ver con manejar el cambio para mejorar los procesos de negocio que por años se han gestionado con distintas técnicas y herramientas (ej. workflows), pero con falta de estándares y ciclo de vida completo para diseñarlos y ejecutarlos. El manejo del cambio requiere control y entendimiento de los procesos para eso son necesarios estándares de modelado y ejecución de procesos.

4.3.2. Modelado de Procesos del Negocio con BPM

Business Process Management Initiative (BPMI) [BPMI] promueve tres estándares: Business Process Modeling Notation (BPMN) para modelado de procesos, como estándar de notación para especificarlos; Business Process Modeling Language (BPML) para ejecución de procesos, como estándar de Business Process Execution Language (BPEL) y Business Process Query Language (BPQL) para distribución y ejecución de procesos de e-Business, como interface de gestión estándar. BPMN ha sido adoptado en febrero de 2006 como estándar del OMG [OMG].

"Una característica crucial y fundamental que distingue los estándares propuestos por BPMI es que han sido desarrollados con una base matemática sólida. La rama de Pi-Calculus del Process Calculi fue utilizada, que es un método formal de computación que provee la base de procesos dinámicos y móviles. Esto hace que dichos estándares sean análogos en su base matemática a la teoría relacional de conjuntos que está bajo los Manejadores de Bases de Datos. Esto significa que los procesos diseñados utilizando el estándar BPMN pueden ser manipulados directamente y creados con un lenguaje ejecutable y puestos a disposición para su ejecución inmediata. Nuevamente, esto es análogo a la funcionalidad de los modelos de datos relacionales y la generación de sentencias SQL/DDL. El lenguaje de modelado de procesos BPML es diseñado por BPMI para ser una descripción estándar basada en Pi-Calculus de los procesos del negocio" [OWEN03].

A continuación se presentan los aspectos principales para el modelado de procesos del Negocio con BPM, en 4.3.2.1 se presenta la notación BPMN para modelado gráfico de procesos, en 4.3.2.2 se presenta el lenguaje BPML estándar para lenguajes de ejecución de procesos BPEL, y en 4.3.2.3 se presentan los sistemas de BPM para ejecución de procesos definidos con BPMN y BPML.

4.3.2.1. Business Process Modeling Notation (BPMN)

Business Process Modeling Notation (BPMN) es un estándar para modelar visualmente flujos de procesos que tiene como objetivo proveer notación común a analistas del negocio que crean los flujos iniciales de los procesos y desarrolladores de software responsables por la tecnología e implementación de los procesos. Está basado entre otros en Diagrama de Actividad de UML y Diagrama de Flujo Actividad-Decisión, por lo que tiene similitud con ambos, pero su público objetivo son los analistas del negocio. BPMN especifica un único tipo de diagrama, el Business Process Diagram (BPD) con un conjunto de elementos núcleo y un conjunto de elementos completo que extiende dicho núcleo. Sin embargo, el conjunto núcleo serviría para modelar la mayoría de los procesos de negocio [BPMN06].

Según [OWEN03] BPMN provee un número de ventajas para modelar procesos del negocio sobre UML. Primero, ofrece una técnica de modelado del flujo de los procesos que es más propicia a la forma de modelado de los analistas del negocio. Segundo, su base matemática sólida está expresamente diseñada para mapear a Business Process Execution Languages (BPEL), mientras UML no. BPMN puede mapear a UML, y proveer un frontend de modelado del negocio sólido para sistemas diseñados con UML. En [KRAF05] se hace notar que "BPMN fue especialmente diseñado con motores de BPM en mente. Consecuentemente, incluye el mapeo de sus objetos gráficos a BPML". En la figura 4.14 se muestra un ejemplo básico de BPD para ilustrar el tipo de notación.

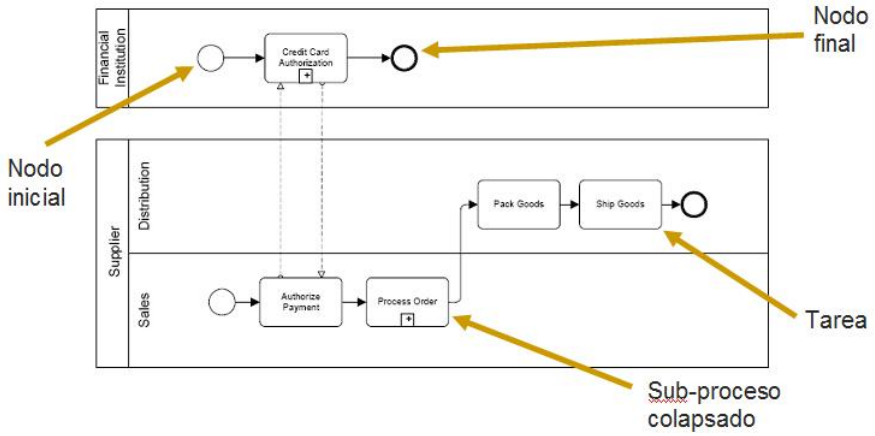


Figura 4.14: Ejemplo básico de Business Process Diagram (BPD) en BPMN de [BPMN06]

Es importante mencionar que BPMN incluye tres tipos básicos de submodelos en un modelo completo en BPMN: procesos del Negocio Privados (internos), Procesos Abstractos (públicos) y Procesos de Colaboración (globales). Los procesos del Negocio privados son internos a una Organización específica y son el tipo de procesos que generalmente han sido llamados workflows o procesos BPM. Los procesos abstractos representan las interacciones entre procesos del Negocio privados y otro proceso o participante (por ejemplo humano). Un proceso de colaboración muestra las interacciones entre una o más entidades de negocio. Estas interacciones se definen como una secuencia de actividades que representan los patrones de intercambio de mensajes entre las entidades involucradas [BPMN06]. Los primeros corresponderán a orquestaciones y los últimos a coreografías, los procesos abstractos dependerá de quién es el otro participante el alcance del mismo.

4.3.2.2. Business Process Modeling Language (BPML)

Business Process Modeling Language (BPML) es un estándar para lenguajes de ejecución de procesos (BPEL) basado en XML. Establece un formato estándar para expresión e intercambio de procesos independiente de la implementación. BPMN se mapea directamente a Business Process Modeling Language (BPML) y también a Business Process Execution Language for Web Services (BPEL4WS).

Según [SHFP03] "BPML es una especificación tanto para modelar procesos del Negocio como para construir sistemas de gestión de procesos. Provee el modelo abstracto para todos los procesos, junto con sintaxis y XML-Schema basados en estandares para expresar y gestionar procesos de Negocio. ... Un aspecto clave de BPML es que es directamente ejecutable en una infraestructura de TI. ... BPML es ejecutado por una "máquina virtual de procesos" en el BPMS. Esto es comparable a la forma en que un programa Java es ejecutado en una "máquina virtual java" provista por el sistema operativo. ... hay una correspondencia uno a uno entre BPML y BPMN. El diagrama representa el código y el código representa el diagrama. No hay pérdida de información al moverse de uno al otro." [SHFP03] En la figura 4.15 se muestra esta correspondencia.

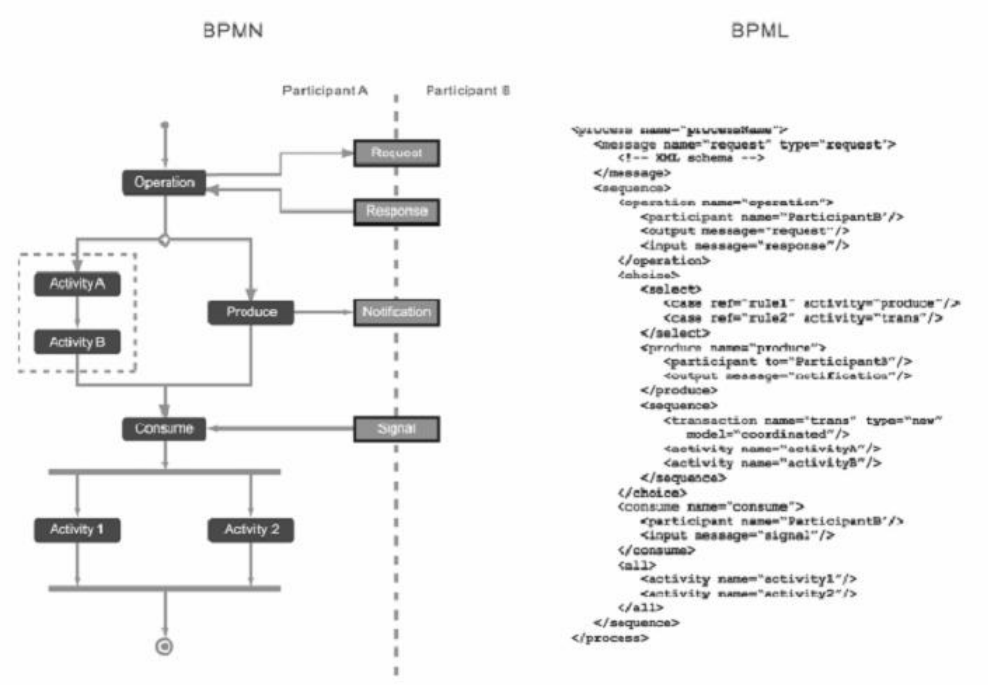


Figura 4.15: Correspondencia uno a uno entre BPMN y BPML de [SHFP03]

[ERLT05] plantea que la brecha entre el modelado de los procesos del negocio por los analistas y la implementación de esos procesos por los arquitectos y desarrolladores "está siendo tratada mediante lenguajes de modelado del negocio operacionales, como WS-BPEL. Existen herramientas de modelado, que permiten que analistas y arquitectos técnicos creen en forma gráfica los diagramas de los procesos del negocio que representan los requerimientos de lógica de workflow, y autogeneran sintaxis de WS-BPEL por debajo. El resultado es un diagrama en el frontend que expresa la visión que tienen los analistas del proceso, y una definición del proceso ejecutable en computadoras en el backend que puede ser manejada por arquitectos y desarrolladores para implementación inmediata (no abierta a interpretaciones)."

4.3.2.3. Business Process Management Systems (BPMS)

Los Business Process Management Systems (BPMS) o sistemas de BPM, entran entonces en juego para permitir el modelado y ejecución de los procesos del negocio. En [KRAF05] se plantea que "un producto BPMS debe permitir a los analistas del negocio, desarrolladores de software y administradores de sistemas, modelar y distribuir (deploy) procesos del negocio (en tiempo de desarrollo) e interactuar, monitorear y analizar instancias de procesos (en tiempo de ejecución)".

Según [PYKE05] "una buena parte de la tecnología que provee la base para los conceptos de automatización de procesos se encuentra en los esfuerzos iniciales de la comunidad de workflow, de la cuales la mayoría en esos momentos ofrecían ruteo de documentos y herramientas de integración. Desde ese momento hasta ahora las herramientas han evolucionado a productos que incorporan todas las características y funciones necesarias para diseñar, ejecutar y monitorear una amplia gama de procesos. Por lo que, la tecnología base de procesos no es algo nuevo, sino que lo que ha cambiado es darse cuenta que los gerentes de negocio necesitan comprender y mejorar esos procesos."

Según [SHFP03] un "BPMS puede ser visto de dos formas distintas: como una nueva plataforma sobre la cual será construida la nueva generación de aplicaciones de negocios, o como una nueva capacidad embebida en las categorias actuales de sistemas de negocios. En cada caso la analogía es entre los RDBMS existentes y los nuevos BPMS, entre los datos relacionales y los procesos, entre el ciclo de vida de gestión de los datos y el ciclo de vida de gestión de los procesos." Agrega que "Los sistemas legados existentes, sin embargo, permanecen valiosos tanto para el desarrollo interno o externo basado en procesos, porque su funcionalidad, actualmente embebida, puede ser encapsulada por el BPMS como componentes de software, que contribuyen a diseños nuevos o mejorados de procesos del Negocio."

Para la utilización de BPMS en [SHFP03] se indica que "los BPMS permiten una metodología de tres pasos hacia la integración de los procesos de Negocio, en la cuya aplicación se verán involucrados distintos roles, principalmente los analistas del negocio durante todo el proceso, pero también desarrolladores de software y administradores de sistemas. El primer paso involucra el modelado del proceso del Negocio mediante una interface gráfica (GUI) y los patrones de diseño de procesos subyacentes son almacenados en un repositorio de procesos, accesible a varios usuarios en la red. En segundo lugar los procesos almacenados en el repositorio son instalados en el servidor de procesos mediante herramientas automáticas, para lo cual el servidor de procesos no tiene porqué ser interrumpido, permitiendo entonces agregado y modificación de procesos del negocio en forma dinámica. Mediante herramientas provistas también se puede consultar el estado de cualquier instancia de procesos y del servidor de procesos. En tercer lugar los analistas del negocio y administradores de sistemas pueden gestionar los procesos que se están ejecutando, utilizando lenguajes de consulta de procesos estándares." En [OMG] se puede encontrar una extensa lista de herramientas con enfoque BPM.

4.3.3. Como se incluye BPM en SOA

En [KRAF05] se plantea que "la visión de BPM es una visión fuerte, en vez de incluir directamente en el código información y reglas sobre procesos del Negocio importantes, esta información

es retirada de los sistemas de aplicación y puesta bajo el control de un BPMS. BPM facilita la modificación, reconfiguración, y optimización de definiciones de procesos con herramientas gráficas que pueden ser utilizadas por analistas del negocio con menor orientación tecnológica. ... SOA provee la funcionalidad del backend que es requerida por un BPMS para implementar la funcionalidad de sus procesos. Todos los conceptos discutidos previamente respecto a SOA, incluyendo el desarrollo evolutivo de capas báica, intermediaria y de procesos, tienen en sentido en este contexto. SOA se convierte en la infraestructura que permite la Organización orientada a procesos." En la figura 4.16 se muestra como encajan juntos los enfoques BPM y SOA.

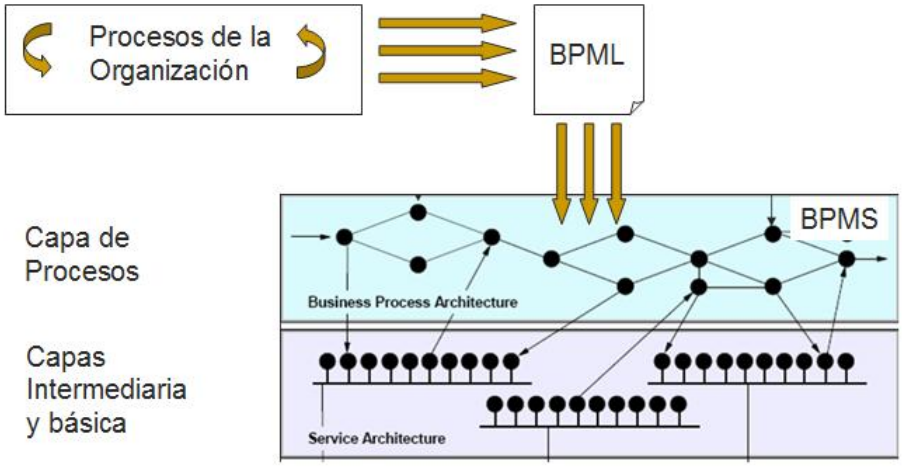


Figura 4.16: SOA provee la infraestructura para BPM adaptada de [KRAF05] [ENDR04]

El modelo presentado se puede ver como el paso final en la adopción de SOA, donde las Organizaciones, luego de atravesar los niveles de SOA planteados, podrán contar con una base de servicios definidos e implementados, que permitan la introducción de BPM incluyendo BPMS que permitan el modelado gráfico de los procesos del Negocio con BPMN, que serán traducidos en forma automática a BPEL que podrá ser ejecutado directamente en los motores de procesos que estos sistemas proveen. Esto permitirá cerrar la brecha entre los analistas del negocio y los desarrolladores de software, donde cada uno podrá enfocarse en su área de conocimiento manteniendo a la vez un objetivo común: la Organización centrada en procesos del Negocio que reacciona ágilmente a los cambios originados en ambas partes.

Como se presentó en la subsección 4.2.3.3 la capa de orquestación de servicios, que sería entonces realizada en el motor de procesos de un BPMS, permite absorber rápidamente los cambios originados tanto en los procesos del Negocio como en las tecnologías subyacentes, facilitando la reconversión de procesos existentes y la introducción de nuevos procesos por parte de los analistas del negocio, a la vez que las actualizaciones de la implementación de los servicios subyacentes (software implementado) originadas por cambios de tecnología o requerimientos de actualización de plataformas o nuevas posibilidades que se incorporan a las plataformas existentes.

Queda claro que las ideas anteriores son atractivas a las Organizaciones y a los involucrados en el área de TI en las mismas, así como a los analistas del Negocio. Sin embargo, no debe subestimarse el esfuerzo que implica para una Organización, sobre todo si es de gran tamaño, el compromiso de incorporar el enfoque SOA con BPM, y el software requerido. Otra línea abierta de trabajo en la conjunción de estos enfoques, es el proceso de desarrollo a seguir para poder partir desde el área del Negocio para el modelado de los procesos del Negocio de la Organización, definiendo la interface del área del Negocio con el área de TI, y el trabajo en el área de TI para la definición, modelado, implementación, despliegue y mantenimiento de los servicios que realicen los procesos del Negocio modelados.

4.4. SOA y Model Driven Architecture (MDA)

El enfoque de desarrollo Model Driven Architecture (MDA) según la especificación de OMG en [MDA03] se basa en la utilización de modelos en el desarrollo de software, y en la idea de separar la especificación de un sistema, de los detalles de la forma en que dicho sistema utiliza las capacidades de sus plataformas. Está guiado por modelos en tanto provee sentido a la utilización de modelos como base para dirigir el curso de la comprensión, diseño, construcción, deployment, operación, mantenimiento y modificación de los sistemas.

A partir de la separación de intereses planteada en el enfoque, se buscan obtener tres metas principales para los sistemas desarrollados: portabilidad, interoperabilidad y reusabilidad. Esta separación de intereses se ve reflejada en la definición de las tres vistas de un sistema que se proveen en los modelos definidos: Computation Independent Model (CIM), Platform Independent Model (PIM) y Platform Specific Model (PSM). El aspecto central del enfoque se encuentra en la transformación de modelos, que según la definición provista por el estándar, es el proceso de convertir un modelo en otro modelo del mismo sistema.

El ciclo de vida en un desarrollo utilizando MDA comienza como en cualquier proceso de desarrollo por la captura de requerimientos. Estos en su mayoría se especifican en forma de texto, sobre los cuales se realiza el análisis de requerimientos que da lugar al primer modelo del sistema que es un modelo conceptual del mismo. Este modelo en el enfoque MDA se conoce como CIM (Computation Independent Model) para el cual en general no se definen transformaciones. A partir del CIM en forma manual se realiza el PIM (Platform Independent Model) que es un modelo de diseño independiente de la implementación, sobre el cual se definen transformaciones que llevarán ese modelo al PSM (Platform Specific Model) a partir del cual se generará el código correspondiente a la plataforma elegida. Es importante notar que a partir de un mismo PIM se pueden generar tantos PSM como plataformas objetivo se definan, y también se puede generar directamente el código sin pasar por un PSM.

A continuación se presenta brevemente el enfoque Model Driven Architecture (MDA), en 4.4.1 se introducen conceptos del enfoque, en **??** se presentan enfoques de desarrollo utilizando MDA, y finalmente en 4.4.3 se presentan aspectos de la conjunción de SOA con MDA. No se pretende en esta sección brindar una descripción exhaustiva del enfoque, sino presentar sus principales conceptos para mostrar porqué y como podría conjuntarse su aplicación con SOA "para el desarrollo de aplicaciones orientadas a servicios basadas en modelos."

4.4.1. Conceptos en MDA

En esta sección se presentan los conceptos que constituyen el enfoque MDA según el estándar [MDA03]: en 4.4.1.1 se describen los tres tipos de modelos que se deben especificar: el Computation Independent Model (CIM), el Platform Independent Model (PIM) y el Platform Specific Model (PSM), así como los conceptos de mappings, marcas y transformaciones de modelos. En 4.4.1.2 se presentan el modelado que plantea OMG y los estándares asociados, que permiten la definición del enfoque MDA.

4.4.1.1. Elementos de MDA

El Computation Independent Model (CIM) es una vista del sistema desde el punto de vista independiente de la computación. No muestra detalles de la estructura del sistema, modelando los requerimientos del sistema sin conocimiento de los modelos o artefactos que serán utilizados para realizar las funcionalidades establecidas. Puede verse como un modelo de dominio o del negocio, y es independiente de cómo el sistema es implementado. Actúa como entrada para el resto de los modelos y como fuente de vocabulario compartido para usar en otros modelos. Los requerimientos establecidos en el CIM deben ser trazables al PIM y al PSM que los implementan, y viceversa. En general no se menciona ni se tiene en cuenta el uso de este modelo en los artículos

que discuten el tema, salvo algunos pocos, ni en las herramientas que implementan el enfoque MDA.

El Platform Independent Model (PIM) es una vista del sistema desde el punto de vista de independencia de la plataforma. Describe el sistema desde el punto de vista del software que lo compone, sin presentar detalles de cómo serán utilizadas las plataformas asociadas. En base a dicha independencia, es adecuado para utilizar en la generación de modelos específicos para varias plataformas distintas. Por ejemplo, un modelo puede asumir la disponibilidad de características de tipo general de plataforma, como ser invocación remota, sin especificar más que eso, luego la característica indicada será implementada según la plataforma.

El Platform Specific Model (PSM) es una vista del sistema desde el punto de vista específico de la plataforma. Un PSM combina las especificaciones en el PIM, con los detalles que definen como el sistema utiliza ese tipo particular de plataforma. El PSM puede ser la implementación del sistema para la plataforma elegida, si provee toda la información necesaria para construir el sistema y ponerlo en operación, o puede actuar como un PIM que se utilizará para refinamientos posteriores en un PSM que pueda ser implementado directamente. En la figura 4.17 se muestra una transformación genérica en MDA, donde a partir de un PIM y otros elementos, se obtiene un PSM para alguna plataforma definida.

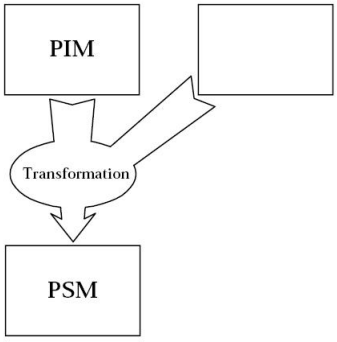


Figura 4.17: Transformación genérica en MDA de [MDA03]

Un mapeo en MDA provee las especificaciones para realizar las transformaciones de un PIM en un PSM para una plataforma específica. El modelo asociado a la plataforma, determinará la naturaleza de los mapeos. Por ejemplo, un PIM especificado en UML mapeado a EJB provee marcas que serán utilizadas para guiar la transformación del PIM al PSM, de forma que marcando una clase con la marca o tag de "session", resulta en la transformación de esa clase según el mapeo establecido, en un "session bean" y otras clases soporte. Un mapeo es especificado utilizando algún lenguaje para describir una transformación de un modelo en otro. La descripción puede ser en lenguaje natural, un algoritmo en un lenguaje de acciones según lo establecido en el Precise Action Semantics incorporado a UML 2.0, o en un lenguaje de mapeo de modelos como se define en el estándar QVT en estudio de aprobación. Para realizar el mapeo de modelos se definen dos enfoques: mapeo de tipos en modelos y mapeo de instancias en modelos.

En un mapeo se pueden utilizar marcas o tags. Las marcas pueden provenir de distintas fuentes: tipos de un modelo (clases, asociaciones, etc.), roles de un modelo (patrones), estereotipos de un perfil UML, elementos de un modelo MOF, elementos de modelos especificados por cualquier metamodelo. Con las marcas se podrían especificar también requerimientos de Quality of Service (QoS) en la implementación, en vez de indicar el objetivo de una transformación se establece un requerimiento sobre ese objetivo. Para que las marcas puedan ser utilizadas adecuadamente, deben ser estructuradas, restringidas y/o modeladas, por ejemplo un conjunto de marcas que indique alternativas mutuamente excluyentes para un concepto, deben ser agrupadas de forma que al marcar el modelo se sepa cuales son las posibles elecciones y que más de una de esas marcas no pueden ser aplicadas al mismo elemento del modelo.

Algunas marcas incluso podrían necesitar parámetros, como las de QoS, por ejemplo "soportar conexiones simultáneas" podría requerir un parámetro que indique cota superior de la cantidad de conexiones que debe soportar, o incluso varios parámetros indicando detalles para time-outs o políticas de conexión. Un conjunto de marcas podría ser especificado por un modelo de marcas independiente de los mapeos, en vez de ser provisto por un mapeo en particular, y así ser usado por distintos mapeos. También un conjunto de marcas podría ser provisto conjuntamente con un perfil UML y varios mapeos distintos podrían ser provistos con ese perfil.

Luego que se tiene el PIM marcado, el siguiente paso es transformarlo en un PSM. Esta transformación puede ser realizada según distintos enfoques: en forma manual, asistido por computadora o en forma automática. La entrada para la transformación es el PIM marcado y el mapeo, el resultado de la misma es el PSM obtenido y el registro de la transformación. La transformación es realizada según el tipo de mapeo que se utilice: de tipos o instancias en modelos, y puede ser directamente a código en vez de a otro modelo. Los resultados de la transformación de un PIM utilizando una técnica particular, son el PSM obtenido en la transformación y el registro de la misma. El registro de una transformación contendrá un mapeo del elemento en el PIM a los correspondientes elementos en el PSM, mostrando que parte del mapeo fue utilizado para cada parte de la transformación. Una herramienta que lleve un registro de la transformación podría mantener la sincronización entre el PIM y el PSM correspondiente, cuando alguno sea modificado.

Otro elemento importante es la posibilidad de construir modelos "ejecutables", que según se describe en [MELL04] "tienen todo lo necesario para producir la funcionalidad deseada en un único dominio. Estos modelos no son esbozos ni planos, sino como su nombre lo sugiere, los modelos ejecutan. ... Una forma de expresar los modelos ejecutables involucra el uso de Executable UML (xUML), o UML ejecutable, que es un perfil de UML que define semántica ejecutable para un subconjunto de UML seleccionado cuidadosamente. Este subconjunto es computacionalmente completo, por lo que un modelo UML ejecutable puede ser ejecutado directamente. ... Los modelos ejecutables también proveen la base para verificación temprana. Como el modelo es ejecutable, se puede demostrar que es correcto (o no) tan pronto como es concebido, sin esperar al diseño o la implementación." En [MDAC02] se propone un enfoque xMDA, o MDA con xUML, basado en dos herramientas para modelado, compilación de modelos, generación y simulación de modelos ejecutables. Un estudio sobre MDA y las herramientas mencionadas puede verse en [MDAD05].

4.4.1.2. Bases de MDA en OMG

El Object Management Group (OMG) [OMG] utiliza una arquitectura de cuatro capas de modelado sobre la cual está basada la definición de sus estándares, y el concepto de metamodelado definido como: el uso de modelos (en una capa de abstracción) para describir modelos (en una capa inferior siguiente). Estas capas se notan M0, M1, M2 y M3; cada una y sus dependencias están definidas por el OMG como sigue:

Capa M3, el Meta-Meta-Modelo: esta capa contiene entidades para definir lenguajes de modelado. El estándar Meta-Object Facility (MOF) que se ubica en esta capa, es el lenguaje de la OMG para definir lenguajes de modelado, por ejemplo una "MOF Class". Los elementos de la capa M3 se definen con instancias de conceptos de la misma capa, o sea que MOF se define también en base a MOF.

Capa M2, el Meta-Modelo: en esta capa se ubican los metamodelos que contienen las entidades de los lenguajes de modelado definidos. Los estándares Unified Modeling Language (UML) y Common Warehouse Metamodel (CWM) se ubican en esta capa, y definen los elementos con los cuales modelar sistemas, por ejemplo "UML Class". Estos elementos son instancias de los elementos en capa M3.

Capa M1, el Modelo: la capa M1 contiene las entidades con que se modela un sistema en particular, según el lenguaje de modelado utilizado, por ejemplo en UML podría ser la Clase

Socio con Atributos Nombre y Apellido. Los elementos en esta capa son instancias de los elementos en la capa M2.

Capa M0, los Datos: la capa M0 contiene los datos específicos que maneja el sistema modelado según la capa M1, por ejemplo el Socio "Juan Perez", que es una instancia de la Clase Socio con Atributos Nombre y Apellido definida en la capa M1. Estos elementos son instancias de los elementos en la capa M1.

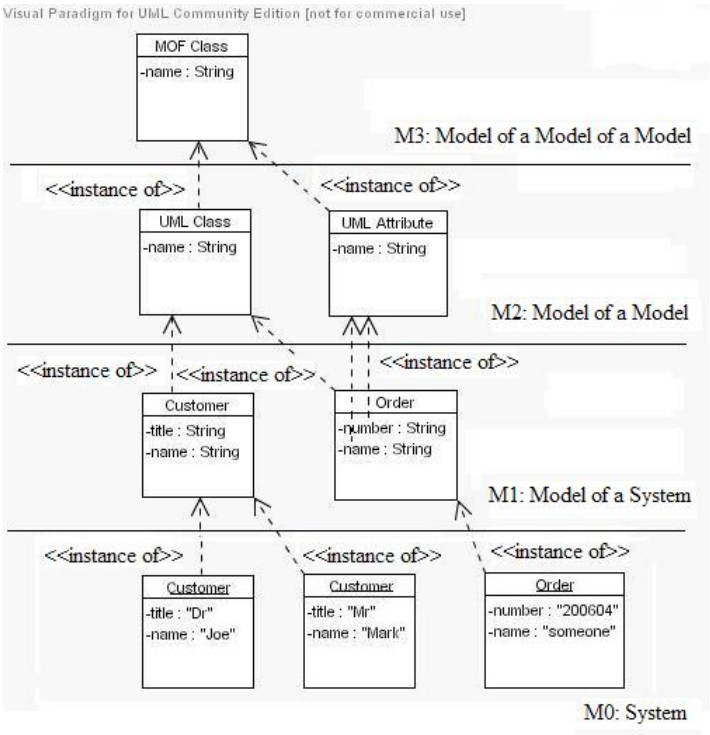


Figura 4.18: Capas de modelado en OMG de [KLEP03]

El interés de presentar el enfoque de modelado del OMG radica en que el enfoque MDA se basa en la existencia del metamodelado para la definición de transformaciones entre modelos, lo que constituye la base de la propuesta. A continuación se presentan los principales estandares sobre los que MDA está definido y que pueden consultarse en OMG [OMG]. La importancia de los mismos es que su utilización permite realizar en forma automática, validación y ejecución de modelos, y transformación de modelos, lo que como ya se mencionó, constituye el elemento central del enfoque MDA.

Meta Object Facility (MOF): este estándar define un lenguaje común para definir lenguajes de modelado, por ejemplo UML, lo que permitirá definir transformaciones sobre los metamodelos en forma estándar. En base a MOF se define un repositorio de modelos que puede utilizarse para especificar y manipular modelos.

XML Metadata Interchange (XMI): estándar que permite el intercambio de modelos via documentos XML. XMI se utiliza para intercambiar modelos en el nivel M1, y para generar formatos de intercambio de metamodelos en la capa M2, según se describe en sección que sigue.

Unified Modeling Language (UML): es el lenguaje de modelado estándar para especificar, visualizar y documentar sistemas de software. Los modelos en MDA son expresados en su mayoría utilizando UML. En UML 2 se integran un conjunto de conceptos para especificar completamente el comportamiento de objetos, el UML Precise Action Semantics, que permite agregar información sobre el comportamiento de los objetos en el modelo de los sistemas.

Perfil UML: es un mecanismo de extensión de UML. Se aplica a la especificación de un lenguaje, especificando un nuevo lenguaje de modelado mediante el agregado de nuevos elementos o restricciones al lenguaje. Un modelo que utiliza un perfil UML es un modelo UML. Actualmente existen perfiles para CORBA, Java, EJB y C++, lo que permitirá construir PSMs específicos para estas tecnologías.

Object Constraint Language (OCL): es un lenguaje de especificación para escribir expresiones sobre modelos, por ejemplo invariantes, pre y postcondiciones, y cuerpo de operaciones. Mediante OCL se podrían especificar transformaciones describiendo los elementos en el modelo origen y destino, aunque el lenguaje estándar propuesto por OMG para definir transformaciones es el que se describe a continuación.

Query, Views and Transformations (QVT): define lenguajes usando MOF para especificar como se realizan las transformaciones entre modelos. Es un estándar en desarrollo con gran relevancia para MDA, ya que actualmente no hay establecido un modo estándar de definir transformaciones entre modelos. Se definen lenguajes para crear vistas de un modelo, realizar consultas sobre modelos y escribir definiciones de transformaciones entre modelos, siendo este último el de mayor relevancia para MDA, el Model Transformation Language (MLT) que utiliza el pattern matching como uno de sus factores clave para permitir la definición de transformaciones entre modelos.

4.4.2. Enfoques y Herramientas para desarrollo MDA

El estándar [MDA03] define conceptualmente el enfoque MDA y sus elementos, como se presentó en 4.4.1, pero no prescribe un proceso o enfoque de desarrollo para su aplicación, en cuanto a actividades a realizar, entregables a obtener, roles involucrados y definición de fases en el desarrollo. Desde la publicación del estándar se han investigado y propuesto distintos enfoques para el desarrollo con MDA, incluyendo por ejemplo la combinación con un enfoque ágil de desarrollo presentada en [MELL04], o las consideraciones de la adopción de MDA con RUP planteadas en [CONA05]. La formalidad de la definición de los enfoques de desarrollo planteados van desde muy informales hasta la definición del proceso con SPEM [SPEM05] realizada en [GAVR04] para el desarrollo de aplicaciones distribuidas con MDA.

Por lo que, la definición de un proceso de desarrollo con enfoque MDA, constituye un trabajo en si mismo, por lo que, como parte de las investigaciones en el marco de la realización de este trabajo de tesis, se propone el proyecto de grado "Extensión MDA (Model Driven Architecture) para el Proceso de Desarrollo de Software del curso Proyecto de Ingeniería de Software (PIS)" [EMDA06], propone actividades, entregables, roles y herramientas a utilizar en un desarrollo MDA con RUP. Una descripción de este trabajo se puede ver en [EMDA07], que permitió entre otros aspectos, evaluar en parte las capacidades y madurez de las herramientas MDA.

Existen múltiples herramientas que implementan en alguna medida la especificación de MDA [MDA03], una lista de las recomendadas por la OMG puede verse en [OMG]. Entre estas herramientas, se pueden destacar como open source AndroMDA [ANDR] que fue la utilizada en la prueba de la Extensión MDA, y como comerciales Rational Software Architect (RSA) [RSA], entre otras. Las características de las herramientas con soporte MDA que se pueden encontrar son bastante disímiles también, un breve resumen de la comparación de las características más importantes de algunas herramientas MDA puede verse en [EMDA07].

4.4.3. Como conjuntar MDA y SOA

Como se presentó en 4.2.2 SOA es un estilo de Arquitectura de Software que se compone de determinados elementos que deben ser modelados, diseñados e implementados para obtener un producto que tenga las características de orientación a servicios. Por otro lado, como se presentó en 4.4.1 MDA es un enfoque para el desarrollo de aplicaciones, en principio utilizando

cualquier estilo de Arquitectura de Software, que se basa en la realización de modelos y en la transformación automática entre los mismos.

Sin embargo, como se presentó en 4.4.2 no se prescribe un proceso de desarrollo para utilizar con el enfoque MDA, por lo que, distintas combinaciones son posibles. Dado que para la construcción de aplicaciones con desarrollo SOA también es posible utilizar distintos procesos de desarrollo, es una línea abierta de trabajo la definición de procesos de desarrollo que incorporen MDA y SOA para "el desarrollo basado en modelos de aplicaciones orientadas a servicios". Un aspecto importante de esta combinación es la capacidad que se necesita que tengan las herramientas MDA para aportar a un desarrollo SOA. No referidas a generación de elementos para implementación con Web Services, que muchos IDE de desarrollo ya tienen incorporadas, sino para la generación conceptual de los elementos de SOA, incluyendo interfaces, componentes, y publicación de los mismos.

Como se establece en [**?**] "SOA es un enfoque arquitectónico que busca alinear los procesos del Negocio con protocolos de servicios y los componentes de software y aplicaciones legadas por debajo que los implementan. Tanto los servicios como los procesos deben ser coordinados cuidadosamente para asegurar una implementación de SOA efectiva. No es posible realizar SOA sin un modelo claro de los procesos del Negocio que deben soportarse. Y no se pueden vincular los procesos del Negocio con los modelos de servicios sin los estándares de modelado que OMG está desarrollando como parte de su estándar MDA". En la figura 4.19 se presenta la visión de SOA de OMG.

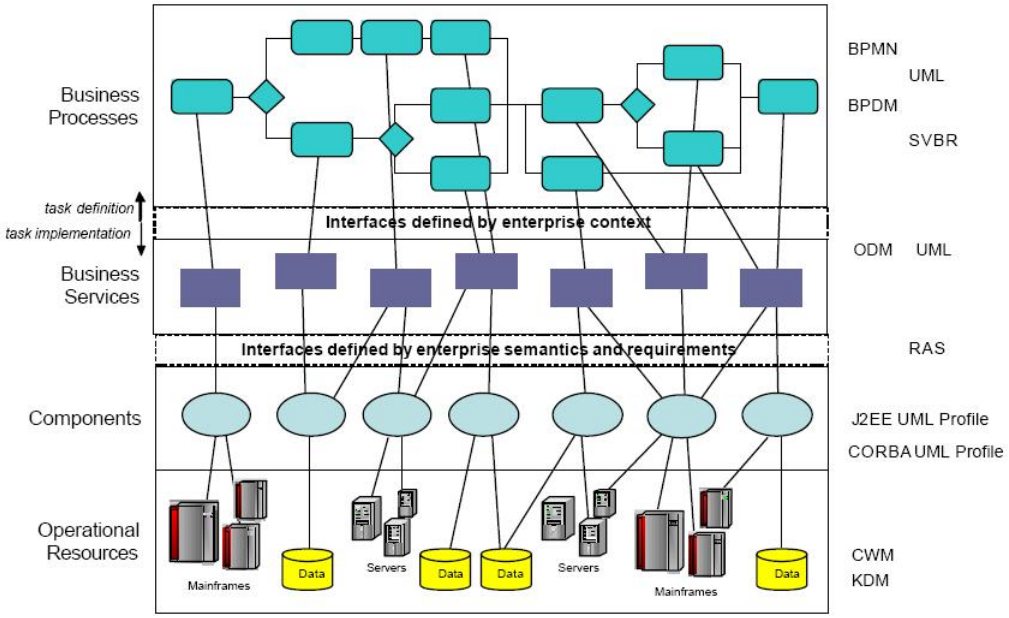


Figura 4.19: Visión de SOA de OMG y los estándares OMG asociados de [**?**]

El planteo que realiza OMG en [**?**] es que "para tener éxito con SOA, las compañías necesitan entender como modelar los procesos, los servicios y los componentes, y como juntar todos los modelos en forma consistente. MDA es sobre modelar y gestionar modelos, en particular trata aspectos relacionados con la independencia de plataformas. Mientras otras organizaciones se han enfocado en estándares específicos para la integración o para protocolos de Web Services (ej. WS-*), OMG ha tomado una visión independiente de la plataforma. Cada una de las capas de SOA puede ser guiada por modelos. Los estándares MDA ofrecen la capacidad de diseñar una solución SOA completa mediante modelos, y minimizar el esfuerzo invertido en tecnologías y protocolos específicos (que siguen cambiando rápidamente)".

OMG ha formado recientemente un nuevo grupo de trabajo para SOA, denominado SOA Special Interest Group (SIG), para coordinar esfuerzos entre OMG y otras entidades de estandarización de SOA como [W3C] y [OASIS]. El grupo SOA SIG tiene como objetivo asegurar que a medida que se estabilicen los enfoques específicos de modelado SOA y las buenas prácticas relacionadas, los estándares OMG evolucionen o se creen nuevos estándares para permitir el modelado de extremo a extremo de aplicaciones SOA. Como se mencionó en 4.3 OMG ha adoptado el estándar BPMN para modelado de procesos del Negocio, y recientemente ha definido el Business Process Definition Metamodel (BPDM) que contiene la información necesaria para generar código BPEL.

Por lo que la tendencia en la comunidad indica que el siguiente paso es conjuntar MDA con SOA, incluyendo el nivel más avanzado de SOA dado por SOA con BPM, como se presentó en 4.3.3. Sin embargo esta conjunción está aún en proceso, y no será una tarea simple, desde mi punto de vista, conjuntar los estándares, herramientas y tecnologías para alcanzar dicha integración. Se detecta la falta de herramientas MDA para generación conceptual de servicios y componentes que los implementan, basadas en estándares aceptados, lo que permitiría entonces utilizar herramientas MDA para generar aplicaciones SOA. Un esfuerzo en este sentido se puede ver en la definición del perfil UML para servicios [SPRO] definido por IBM e incorporado en su herramienta MDA de modelado y generación Rational Software Architect (RSA) [RSA], la cual fue evaluada como herramienta MDA mínimamente en [EMDA06], pero no el perfil UML para servicios.

Capítulo 5

Metodología de desarrollo para aplicaciones SOA

En este capítulo se presenta la Metodología para desarrollo SOA propuesta como extensión al proceso de desarrollo base adaptación del RUP con que cuenta el Grupo de Ingeniería de Software (Gris), denominada Extensión SOA. En primer lugar en la sección 5.1 se presenta el contexto en que se realiza la propuesta de la metodología como trabajo de tesis. En la sección 5.2 se presenta brevemente el proceso base adaptación del RUP, cuya concepción y evolución en el marco del programa de construcción y pruebas de modelos de proceso del Gris [GRIS06] se puede consultar en [ADBP06]. Finalmente en la sección 5.3se describen las Disciplinas, Actividades, Entregables y Roles definidos en la metodología SOA propuesta, así como la relación entre éstos y las Disciplinas, Actividades, Entregables y Roles definidos en el proceso base.

Parte de este capítulo fue publicado como artículo y presentado en la "XXXII Conferencia Latinoamericana de Informática (CLEI'06)" realizada en Santiago de Chile, Chile, en Agosto de 2006 [TMAD06].

5.1. Contexto de la propuesta

Una metodología para el desarrollo orientado a servicios no requiere necesariamente un nuevo proceso de desarrollo, sino que se puede construir sobre el proceso o enfoque de desarrollo que se siga en la Organización, agregando actividades y artefactos específicos para el desarrollo orientado a servicios, y por lo tanto en principio cualquier proceso podría servir de base. Sin embargo, dada la naturaleza compleja y cambiante de los requerimientos del negocio en el contexto de las Organizaciones que podrían incorporar el enfoque SOA, es recomendable que se siga un enfoque iterativo incremental como forma de lidiar con estos cambios.

Procesos como el Rational Unified Process (RUP) presentado en la subsección 3.3.2.1 que es clasificado como "proceso pesado", y eXtremme Programming (XP) presentado en la subsección 2.3.2 que es clasificado como "proceso livano" o ágil, presentan ambos como base un enfoque de desarrollo iterativo incremental, que resulta apropiado para un desarrollo orientado a servicios. De ambos procesos además se cuenta con adaptaciones realizadas en el marco del programa de modelado y pruebas de procesos del Grupo de Ingeniería de Software (Gris) [GRIS06], por lo cual en principio, si bien la adaptación de XP estaba orientada a Genexus, se podía utilizar cualquiera de los dos procesos mencionados como base, realizando los agregados y modificaciones necesarias para el desarrollo SOA.

Dada la importancia estratégica de asumir un proyecto de desarrollo con enfoque SOA donde los conceptos asociados al enfoque son novedosos y poco conocidos, resulta más apropiado un

proceso base más del tipo de los "pesados", de forma de ir obteniendo productos claves en el avance del mismo para ir controlando el desarrollo del proyecto y la aplicación de la metodología SOA. La adaptación de XP además fue puesta en práctica en una sola edición del curso "Proyecto de Ingeniería de Software" donde sería probada la Extensión SOA, y no se ve en profundidad en el curso teórico previo de "Introducción a la Ingeniería de Software".

Por otro lado, la adaptación del RUP se viene probando desde la edición del año 2000 del curso, y sus conceptos más importantes si se presentan a los estudiantes en el curso teórico previo, mediante la realización de trabajos obligatorios relacionados. En este contexto se evaluó que proponer una combinación XP con SOA, si bien sería posible, podría introducir aspectos ajenos al enfoque SOA y la metodología definida, como el aprendizaje a la vez de la metodología SOA y del proceso XP, que pondrían en riesgo la evaluación de la aplicación de la metodología en si misma, y siendo el RUP un proceso bastante conocido y utilizado en la industria de software también, resultaba interesante tomarlo como base para realizar la propuesta de desarrollo SOA, teniendo en cuenta su posible generalización.

En base a los aspectos mencionados es que se decide entonces "extender" el proceso de desarrollo base adaptación del RUP del Gris, para incluir desarrollo con enfoque SOA, que requiere la realización de las actividades definidas en las Disciplinas del proceso base agregando las Disciplinas y actividades específicas para SOA que define la metodología, así como los roles involucrados y entregables específicos del enfoque generados desde las actividades definidas. Para su aplicación en el curso "Proyecto de Ingeniería de Software" en su edición 2005, se nombró el proceso definido como Extensión SOA, y se realizó un sitio Web para la misma [WTAD05] que se presenta en el Capítulo 9 - Anexo A, navegable desde el sitio Web del proceso base del curso para que los grupos que lo llevarían adelante dispusieran de la información correspondiente.

5.2. Proceso base adaptación del RUP

El proceso de desarrollo base es principalmente una adaptación del RUP que mantiene las características más importantes del mismo mencionadas en el capítulo 2: es iterativo e incremental, basado en Casos de Uso y centrado en la Arquitectura. Se modela como el RUP en dos dimensiones: el tiempo y las disciplinas.

En la dimensión del tiempo se definen las cuatro fases: Inicial, Elaboración, Construcción y Transición, en las tres primeras fases se definen dos iteraciones de dos semanas cada una, y en la última fase una iteración de dos semanas, para su aplicación en el curso. Se cuenta además con una agenda definida que indica para cada semana que actividades se deben realizar y que entregables se deben obtener.

En la dimensión de las Disciplinas se definen las disciplinas tradicionales del desarrollo de software: requerimientos, diseño, implementación y verificación, más las disciplinas de soporte: gestión del proyecto, de la calidad, de la configuración e implantación. En cada disciplina se definen actividades, roles encargados y participantes de las actividades, y entregables de entrada y salida de dichas actividades.

Para la definición de elementos en la dimensión de las Disciplinas en su concepción en el año 2000 por parte del proyecto de grado [ADBP00], se incluyeron también otros enfoques principalmente de modelos de mejora de procesos, algunos de los cuales se mencionan en el capítulo 2, que aportaron al agregado de Disciplinas, Actividades, Entregables y Roles principalmente en las disciplinas de soporte. Una descripción completa de la concepción y evolución del proceso base utilizado en el curso "Proyecto de Ingeniería de Software" del Gris puede consultarse en [ADBP06].

En lo que sigue se mencionarán brevemente los aspectos más importantes del proceso base para contextualizar la Extensión SOA en la que se incluye la metodología SOA definida, siguiendo el modelado del RUP en dos dimensiones se presenta en 5.2.1 la adaptación de la dimensión del tiempo, y en 5.2.2 la adaptación de la dimensión de las Disciplinas.

5.2.1. Dimensión del Tiempo

Se presentan a continuación los aspectos principales que componen la dimensión del tiempo en el proceso base, las Fases e iteraciones definidas, objetivos más importantes e instanciación en el tiempo de las Fases e iteraciones en la agenda brindada.

5.2.1.1. Definición de Fases e iteraciones

En la dimensión del tiempo se definen como en el RUP las Fases Inicial, de Elaboración, Construcción y Transición, manteniendo los principales objetivos definidos para las mismas. Así, la Fase Inicial está centrada en el relevamiento de requerimientos y la comprensión del problema, la Fase de Elaboración en la obtención del Ejecutable de la Línea Base de la Arquitectura que constituye la implementación de las principales definiciones sobre la estructura de la aplicación realizadas en la Arquitectura de Software, la Fase de Construcción en la implementación del resto de las funcionalidades acordadas en el Alcance del Proyecto y en la obtención de paralelismo en la implementación, y la Fase de Implantación en la instalación del producto en el ambiente del cliente y la realización de las pruebas de aceptación acordadas con el mismo.

En base a las sucesivas aplicaciones del proceso en el curso "Proyecto de Ingeniería de Software" en las cuales se han probado distintas duraciones para cada Fase y distinta cantidad de iteraciones y duración de las mismas en cada Fase, se ha definido una planificación de tiempos que ha resultado ser adecuada en los últimos años para la obtención de los objetivos planteados en cada Fase según lo previsto. Esta planificación brinda una guía del tiempo que debería llevar cada iteración y Fase, para lograr alcanzar los objetivos planteados para cada una. En la figura 5.1 se presentan las Fases e iteraciones definidas para el proceso base:

Fases	Inicial				Elaboración				Construcción				Transición	
Iteraciones	Iter. I		Iter. II		Iter. I		Iter. II		Iter. I		Iter. II		Iter. I	
Semanas	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Figura 5.1: Definición de Fases e iteraciones en el proceso base

Esta definición se provee como guía a los grupos de desarrollo, ya que de todas formas se debe evaluar la obtención de los objetivos de cada Fase en el momento previsto de finalizar la misma, y si estos no han sido alcanzados no es posible pasar a la Fase siguiente. Dada que la duración total del proyecto es fija dada por la duración de semanas del curso, si se extienden las fases iniciales, no será posible extender la duración total del proyecto por lo que se deberá recuperar el tiempo dentro de las fases definidas, acortando si es necesario, la duración de las fases restantes. Esto es una restricción del curso, pero es posible observarlo también en algunos casos de proyectos reales que presentan fechas previstas no flexibles.

5.2.2. Dimensión de las Disciplinas

A continuación se presentan los elementos principales que componen la Dimensión de las Disciplinas para el proceso base, las Disciplinas, Actividades, Entregables y Roles definidos. Se mencionan como ejemplos los aspectos más importantes asociados a cada elemento, de forma de brindar una idea general de la adaptación realizada. En el capítulo 10 - Anexo B se adjuntan las plantillas definidas para los entregables de la Extensión SOA.

5.2.2.1. Definición de Disciplinas, Actividades, Entregables y Roles

En la Dimensión de las Disciplinas se definen las disciplinas tradicionales del desarrollo de software: requerimientos, diseño, implementación y verificación, más las disciplinas de soporte:

gestión del proyecto, de la calidad, de la configuración e implantación. La definición de las primeras está principalmente basada en el RUP, mientras que las disciplinas de soporte están principalmente basadas en CMM, con aportes de otros modelos de calidad y mejora y procesos de desarrollo mencionados en el capítulo 2. En cada Disciplina planteada por el RUP se mantienen sus actividades principales pero se eliminan actividades que no aplican y se agregan otras que tienen que ver específicamente con la adaptación realizada para el curso.

Como principales Actividades definidas en las Disciplinas de Ingeniería se pueden mencionar en la Disciplina de Requerimientos las Actividades relacionadas con la Especificación de funcionalidades como Modelo de Casos de Uso y la realización del Modelo de Dominio, en la Disciplina de Diseño la Definición de la Arquitectura de Software, el Diseño de Casos de Uso y del Modelo de Datos, en la Disciplina de Implementación la implementación de Clases y Subsistemas definidos, en la Disciplina de Verificación la realización de pruebas Unitarias, de Integración y del Sistema sobre cada liberación en cada iteración y la planificación de la verificación.

Como principales actividades definidas en las Disciplinas de Soporte se pueden mencionar en la Disciplina de Gestión de Calidad la planificación del área y la identificación de propiedades de calidad para el sistema, en la Disciplina de Gestión de Configuración la planificación del área y la definición del ambiente controlado y línea base del proyecto, en la Disciplina de Gestión del Proyecto la planificación del proyecto, la realización de estimaciones de tamaño y esfuerzo y la definición del Alcance del Proyecto en base a los requerimientos planteados, las estimaciones realizadas, y el tiempo y recursos con que cuenta el proyecto.

Se definen también entregables principales en base a los planteados por el RUP, para los que se establecen los roles participantes y un rol responsable por la generación del entregable, y se brinda además una plantilla del contenido asociado como guía para la realización del mismo.

Como principales entregables definidos en las Disciplinas de Ingeniería se pueden mencionar en la Disciplina de Requerimientos el Modelo de Casos de Uso del Sistema y la Especificación de Requerimientos, en la Disciplina de Diseño los Modelos de Diseño y Deployment, el Documento de Arquitectura de Software (SAD por sus siglas en inglés), en la Disciplina de Implementacion el Modelo de Implementación y los ejecutables asociados, en la Disciplina de Verificación los reportes de verificación, pruebas unitarias, de integración y de sistema, así como el plan de verificación.

Como principales entregables definidos en las Disciplinas de Soporte se pueden mencionar en la Disciplina de Gestión de Calidad los reportes de revisión de la calidad y el plan de calidad, en la Disciplina de Gestión de Configuración el plan de Configuración y el control de la línea base definida, en la Disciplina de Gestión del Proyecto el plan del proyecto, el documento de estimaciones de tamaño y esfuerzo y el documento de Alcance, en la Disciplina de Implantación el documento de pruebas de aceptación.

Se definen roles básicos que son: Analista, Arquitecto, Especialista Técnico, Implementador, Responsable de Verificación, Administrador, Responsable de SQA, Responsable de SCM, Documentador de Usuario y Asistente de Verificación, los cuales tienen asociada además una descripción de tareas y perfil para cumplirlo.

Además se define una combinación de roles según la carga de trabajo en el tiempo y la compatibilidad de tareas combinadas, por ej. el Arquitecto/Asistente de Verificación es solamente Arquitecto hasta finales de la Fase de Elaboración, y en la Fase de Construcción también pasa a ser Asistente de Verificación para aprovechar su conocimiento del producto en desarrollo en la búsqueda de errores. A cada persona en el equipo se le asigna un rol combinado con indicación en el tiempo de qué actividades correspondientes a qué rol simple debe realizar en cada Fase e iteración.

En la figura 5.2 se muestran los elementos de modelado utilizados en la Dimensión de las Disciplinas para la Actividad del proceso base Describir la Arquitectura (D2):

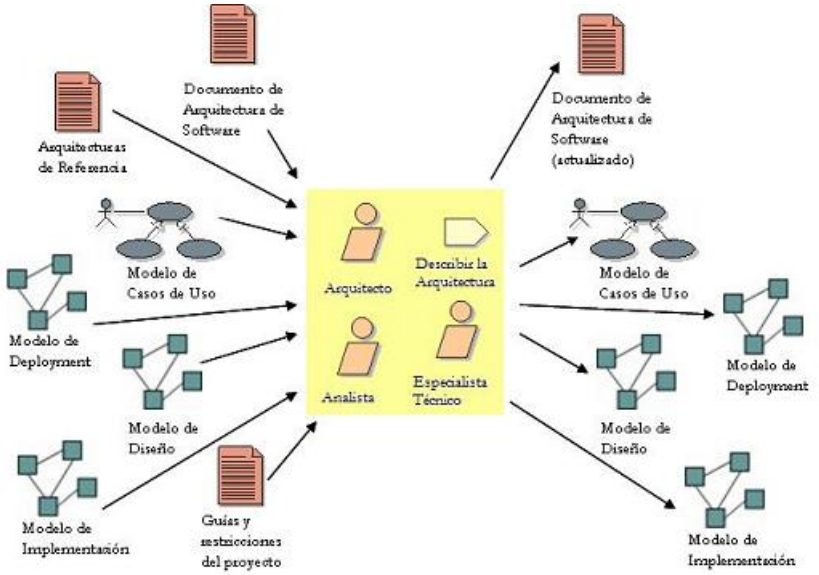


Figura 5.2: Modelado de la actividad Describir la Arquitectura (D2) en el proceso base

5.3. Extensión al proceso base para desarrollo SOA

La Extensión SOA al proceso base implica la realización de todas las actividades previstas en el esqueleto común del proceso más la Extensión OO para desarrollo orientado a objetos, así como la generación de todos los entregables definidos y la participación de roles según está definido tanto en las actividades como en los entregables. De la misma forma implica seguir la forma de desarrollo prevista por el proceso base en cuanto a desarrollo iterativo e incremental, guiado por Casos de Uso y centrado en la Arquitectura. Además se ejecutan las actividades de la Extensión SOA según lo establecido en la relación de cada Disciplina con el proceso base, y se generan los entregables definidos, con la participación de los roles que se indica.

En primer lugar se agrega una disciplina que está en el RUP pero no en el proceso base, que es el Modelado del Negocio. Como se mencionó un aspecto principal en una SOA es el modelado de los procesos de la Organización como forma de conjuntar los aspectos del negocio con las soluciones informáticas provistas, teniendo en cuenta que los servicios orientados a procesos constituirán la traducción de los procesos del negocio a los procesos que se modelan en las aplicaciones de la Organización.

Como parte central de la metodología definida se incorporan cinco actividades en la Disciplina de Diseño, que son las actividades claves del enfoque, en las que se identifican, categorizan, reutilizan, especifican, y definen los servicios de la aplicación, y la orquestación de los mismos para proveer los procesos del negocio modelado. Luego en la Disciplina de Implementación se agrega la actividad de implementación de servicios según fueron definidos en la etapa de diseño. Además para cada actividad se definen los roles responsables, participantes y los entregables entrada y salida de las mismas.

Como entregables se definen dos para la Disciplina Modelado del Negocio, el documento de Evaluación de la Organización Objetivo y el Modelo de Casos de Uso del Negocio, y uno para la Disciplina Diseño, el Modelo de Servicios, que se compone de varias secciones donde realizar la definición, categorización, especificación, etc. de los servicios definidos. Para estos tres entregables se realizaron como parte de la definición de la metodología propuesta, plantillas para su generación que sirvieran de guías en su realización.

Entre los roles definidos los más importantes para la Extensión SOA son: el Arquitecto, quién tiene responsabilidad en el área de Diseño pero participa también en el relevamiento de re-

querimientos y como coordinador del desarrollo; el Analista, quién principalmente participa en el relevamiento de requerimientos; el Especialista Técnico, quién tiene a su cargo el estudio e investigación de las tecnologías disponibles a utilizar y el Implementador, quién tiene a su cargo la codificación de la aplicación.

A continuación se muestran los elementos de modelado correspondientes a las Disciplinas, actividades, entregables y roles definidos en la metodología SOA. Vale aclarar, que se sigue la notación definida para el proceso base, donde las actividades se codifican identificando la Disciplina a la que pertenecen y numerándolas consecutivamente, así las actividades de Requerimientos se notan R1, R2, etc. y las de Diseño D1, D2, etc. Se agrega también para cada Actividad una figura que muestra en forma gráfica el modelado de la misma. En 5.3.1 se presenta la Disciplina Modelado del Negocio, en 5.3.2 se presenta la Disciplina Diseño, en 5.3.3 se presenta la Disciplina Implementación, y en 5.3.4se presenta información sobre las plantillas provistas para los entregables definidos en las Disciplinas presentadas. Finalmente en 5.3.5 se presentan comentarios sobre la inclusión de aspectos de otras Disciplinas en el desarrollo SOA.

5.3.1. Disciplina Modelado del Negocio

El propósito de la disciplina Modelado del Negocio en general es asegurar que clientes, usuarios finales, desarrolladores y otros involucrados tienen un conocimiento común de la Organización objetivo, derivar los requerimientos del sistema de software que son necesarios para apoyar a la Organización y comprender como el sistema de software a implantar se inscribe en la misma. El esfuerzo de modelado del Negocio puede tener distintos alcances dependiendo del contexto y las necesidades, incluyendo por ejemplo un esfuerzo mayor de reingeniería del Negocio.

En el caso particular de la metodología SOA propuesta se está en el escenario de obtener un mapa de la Organización y sus procesos para ganar un mejor entendimiento de los requerimientos de la aplicación en desarrollo. El modelado del Negocio es parte entonces del proyecto de Ingeniería de Software y es realizado principalmente durante la Fase Inicial. En esta disciplina participan el rol de Analista y Arquitecto que son quienes realizan principalmente el relevamiento de requerimientos a partir de reuniones con el cliente. Los entregables que se obtienen de esta Disciplina son la Evaluación de la Organización Objetivo que contiene aspectos claves de la misma, y el Modelo de Casos de Uso del Negocio donde se especifican los procesos identificados en la Organización.

Como se presentó en el estado del conocimiento actual del enfoque Service Oriented Architecture (SOA) en el capítulo 4, una parte importante del enfoque de desarrollo SOA consiste en la identificación y modelado de los procesos del Negocio en la Organización, los que luego serán realizados mediante invocación a una secuencia definida de los servicios implementados para cumplir con estos procesos.

Es un aspecto fundamental de este enfoque entender los procesos del Negocio, modelar sus flujos y participantes, de forma que queden separados de la implementación particular que se realice de los mismos. Esta separación permitirá entonces, que los procesos puedan ser cambiados, mejorados y/o incrementados en forma separada de la infraestructura tecnológica con que se cuente, con mínimo impacto en la implementación de servicios asociada. Y de la misma forma, permitirá actualizaciones y/o cambios en la implementación de los servicios, con mínimo impacto en los procesos del Negocio definidos.

A continuación se describen las actividades propuestas en la Disciplina Modelado del Negocio, en 5.3.1.1 se presenta la actividad Evaluar la Organización Objetivo (MN1), en 5.3.1.2 se presenta la actividad Identificar los procesos del Negocio (MN2), presentando en cada caso el Objetivo de la actividad, su descripción, los roles involucrados y los entregables de entrada y salida asociados con la realización de la misma, una figura que muestra gráficamente el modelado de la actividad, y por último la fundamentación sobre la definición de dicha actividad en la metodología SOA propuesta. Finalmente en 5.3.1.3 se muestra la relación entre la Disciplina Modelado del Negocio agregada en la metodología SOA y el proceso base.

5.3.1.1. Evaluar la Organización Objetivo (MN1)

Objetivo: Esta actividad tiene como objetivo involucrar al equipo de proyecto con la Organización para la cual se está realizando el desarrollo, interiorizando a los participantes sobre aspectos como: el área, funcionamiento, empleados, etc. de la Organización.

Descripción: Se debe describir el estado actual de la Organización en la cual la aplicación será implantada, en términos de sus procesos actuales, herramientas, competencias de las personas, actitudes de las personas, clientes, competencia, desafíos tecnológicos, problemas y áreas de mejora, identificando claramente los involucrados (stakeholders) en el esfuerzo de modelado del negocio.

Roles: El rol responsable de esta actividad es el Analista y participa también el rol de Arquitecto, ya que ambos roles están involucrados en el modelado del Negocio a partir del cual se derivan los requerimientos para la aplicación conjuntamente con el relevamiento de requerimientos que se realiza en la Disciplina de Requerimientos posteriormente.

Entregables de entrada: Esta actividad tiene como entrada información recabada con los clientes del Negocio sobre diversos aspectos de la Organización.

Entregables de salida: Como salida tiene el documento de Evaluación de la Organización Objetivo donde se detallan los aspectos claves relevados.

Modelado de la actividad: en la figura 5.3 se muestra el modelado de la actividad utilizando los elementos de modelado provistos por el RUP.

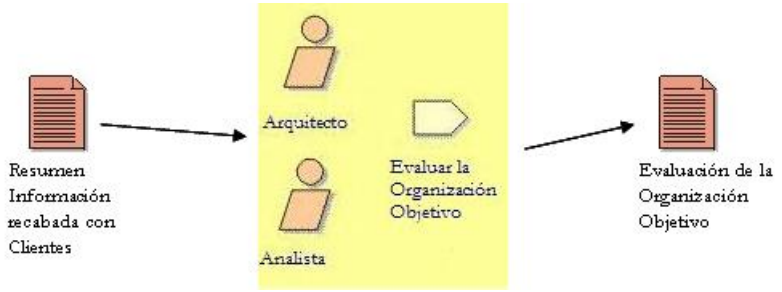


Figura 5.3: Modelado de la actividad Evaluar la Organización Objetivo (MN1) en la Extensión SOA

Fundamentación: Esta actividad se agrega para guíar el enfoque del relevamiento de requerimientos a realizar en un desarrollo SOA. Como se describe en los conceptos teóricos presentados del enfoque en el Capítulo 4, es importante conocer el estado del desarrollo de software en la Organización donde se realizará el desarrollo que se comienza, en particular es importante conocer la infraestructura tecnológica con que se cuenta, los sistemas que informatizan sistemas existentes, la estructura organizacional incluyendo las jerarquías en las distintas áreas, el relacionamiento entre las áreas del Negocio y de TI, las personas que integran el staff de la Organización, en particular las personas de referencia para este desarrollo.

Si bien obtener este conocimiento como parte del relevamiento de necesidades de la Organización para la que se realiza el desarrollo puede ser importante en cualquier proyecto aunque este no sea con enfoque SOA, para este caso resulta imprescindible obtenerlo, ya que parte de las definiciones que se tomarán luego en el diseño y construcción del sistema, se ven influenciadas por el contexto de la Organización, por ejemplo, por la existencia o no de servicios reutilizables a incluir en este desarrollo. Por esta razón se incluye como una actividad específica de la metodología SOA propuesta, de forma que sea realizada obligatoriamente para este tipo de desarrollos.

5.3.1.2. Identificar los procesos del Negocio (MN2)

Objetivo: Esta actividad tiene como objetivo describir y entender los procesos que se realizan en el negocio en lo que tiene que ver principalmente con la aplicación que se va a desarrollar. Para esto los procesos se describen como Casos de Uso del Negocio con los flujos correspondientes y se modelan con Diagramas de Actividad.

Descripción: Esta actividad está compuesta de varias sub-actividades que en conjunto permiten identificar los Procesos del Negocio, describiendo los actores participantes y el flujo de ejecución de los mismos en la Organización. Para esto se deben encontrar los actores y Casos de Uso del Negocio, las reglas del Negocio y los términos que se manejan en el Negocio. Es importante definir los límites del Negocio a ser modelado y quién y qué interactua con el Negocio, describir los procesos del Negocio y plasmarlos en diagramas de actividad en el Modelo de Casos de Uso del Negocio.

Roles: El rol responsable de esta actividad es el Analista y participa también el rol de Arquitecto, igual que en la actividad anterior.

Entregables de entrada: Esta actividad tiene como entrada el documento de Evaluación de la Organización Objetivo y el/las Acta/s de Requerimientos realizada/s en las reuniones de requerimientos con el cliente.

Entregables de salida: Como salida tiene el Modelo de Casos de Uso del Negocio donde se definen los flujos de los procesos identificados en la Organización Objetivo, así como el Glosario con la inclusión de los nuevos términos que vayan apareciendo.

Modelado de la actividad: en la figura 7.1 se muestra el modelado de la actividad utilizando los elementos de modelado provistos por el RUP.

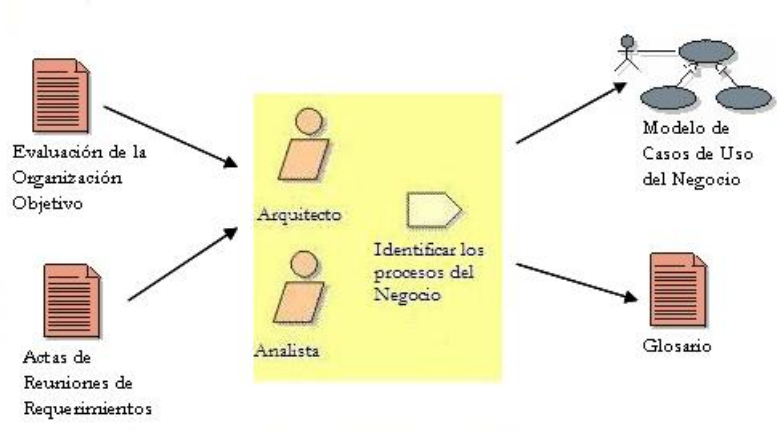


Figura 5.4: Modelado de la actividad Identificar los procesos del Negocio (MN2) en la Extensión SOA

Fundamentación: Esta actividad se agrega ya que resulta imprescindible para desarrollos con enfoque SOA identificar y modelar los procesos del Negocio que serán informatizados en el desarrollo. Como se describe en los conceptos teóricos presentados del enfoque SOA en el Capítulo 4, los procesos del Negocio serán implementados como secuencia de invocaciones a servicios de menor granularidad definidos, en lo que se conoce como orquestaciones de servicios en la Organización, o coreografías de servicios entre varias organizaciones. Esta secuencia de invocaciones a servicios a su vez se realizará como servicios, en servicios centrados en procesos u otra categoría según la clasificación que se utilice.

Por lo que, una entrada fundamental para la identificación y categorización de servicios es el conocimiento de los procesos del Negocio que se realizarán en el desarrollo. El modelado de

estos procesos como Casos de Uso del Negocio es una forma de modelarlos según la metodología planteada por el RUP, en la cual a partir del modelado de procesos del Negocio como Casos de Uso del Negocio, es posible derivar los Casos de Uso del Sistema, de menor granularidad, a partir de los cuales identificar y categorizar servicios de menor granularidad para realizar los Casos de Uso del Negocio especificados.

Si bien obtener este conocimiento como parte del relevamiento de necesidades de la Organización para la que se realiza el desarrollo puede ser importante en cualquier proyecto aunque no sea con enfoque SOA, y puede realizarse en esta Disciplina, con Casos de Uso del Negocio, o de otra forma, describiendo en forma textual o con otras técnicas los procesos del Negocio de la Organización, en este caso resulta imprescindible obtenerlo, ya que uno de los objetivos del enfoque SOA es implementar estos procesos con servicios. Por esta razón se incluye como una actividad específica en la metodología SOA propuesta, de forma que sea realizada obligatoriamente la identificación y modelado de los procesos del Negocio de la Organización.

5.3.1.3. Relación e inclusión de esta Disciplina con el proceso base

En el proceso base la primer Disciplina de trabajo es la de Requerimientos, en la cual se deben realizar actividades como R1 - Especificar los requerimientos y R2 - Definir el Modelo de Casos de Uso del Sistema. La Disciplina de Modelado del Negocio, como plantea el RUP, se realiza en forma previa a la de Requerimientos y constituye una entrada fundamental para la misma.

Los Casos de Uso del Negocio definidos en base al estudio de los procedimientos de la Organización, se analizan para identificar los Casos de Uso del Sistema, que en general serán de menor granularidad dado que identifican interacciones puntuales de los actores con el Sistema, y éstas se encuentran inmersas en procesos de mayor duración y que generalmente cruzan las fronteras de Departamentos de la Organización o incluso de la Organización misma.

En lo que sigue se presenta un ejemplo que relaciona los Casos de Uso del Negocio definidos con los Casos de Uso del Sistema del proceso base, donde en la figura 5.5 se presenta el Caso de Uso del Negocio “Otorgar Préstamo” que se utilizará como ejemplo.

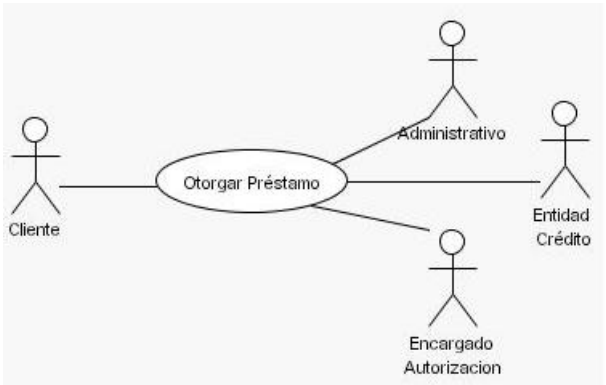


Figura 5.5: Diagrama del Caso de Uso del Negocio ejemplo “Otorgar Préstamo”

El Caso de Uso del Negocio “Otorgar Préstamo” presentado en la figura 5.5 corresponde al proceso del Negocio que tiene definido un banco para realizar los pasos correspondientes al otorgamiento de préstamos a sus clientes. Este proceso comienza cuando un cliente se acerca al banco a solicitar un préstamo, presentando la documentación asociada. Esta solicitud es entonces enviada a la sección de Autorización de préstamos, donde los encargados realizan el estudio de la documentación presentada, así como la evaluación de ingresos-egresos del solicitante, monto pedido, garantías presentadas, así como pedido de informes a alguna entidad de créditos como puede ser por ejemplo un Clearing de Morosos.

La sección expide entonces su resolución aprobando o rechazando la solicitud, y ésta es enviada al área de atención al cliente la que se comunica con el solicitante para informarle de la resolución tomada. En caso de ser aprobada la solicitud, el solicitante debe volver al banco a retirar el efectivo y firmar los pagaré correspondientes. En ambos casos, otorgamiento o rechazo, se archiva la solicitud para tener la historia de solicitudes de la persona y resoluciones asociadas.

Este Caso de Uso del Negocio describe entonces todos los pasos y los distintos actores involucrados en el proceso del Negocio que tiene definido un banco para otorgar préstamos a sus clientes. Cuando debemos pasar a la identificación de los Casos de Uso del Sistema, podemos ver que en el proceso completo se ejecutarán más de un Caso de Uso del Sistema. En la figura 5.6 se muestra el diagrama de Casos de Uso del Sistema que corresponden al Caso de Uso del Negocio ejemplo definido.

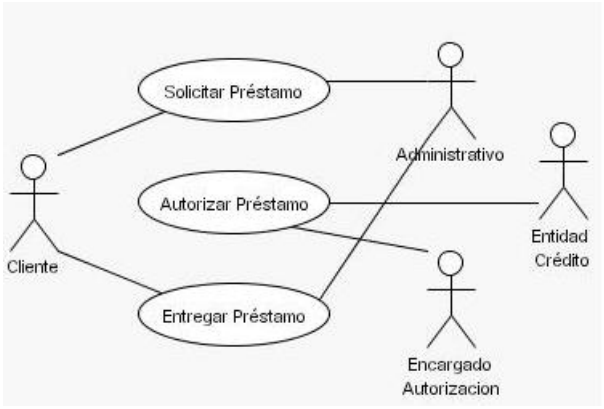


Figura 5.6: Diagrama de Casos de Uso del Sistema a partir del Caso de Uso del Negocio ejemplo "Otorgar Préstamo"

El Caso de Uso del Sistema "Solicitar Préstamo" corresponde a la primer parte de la ejecución del proceso, donde la persona se presenta en el banco a solicitar el préstamo y sus datos y comprobantes son ingresados al sistema, registrándose el pedido y otorgando un número de solicitud asociado. A continuación se ejecutará por parte de la Sección de Autorización de préstamos el Caso de Uso del Sistema "Autorizar Préstamo" donde intervienen no solo los encargados de la sección sino la interacción con la entidad de créditos que proveerá información sobre la morosidad del solicitante en créditos otorgados por otras instituciones previamente.

El Caso de Uso "Autorizar Préstamo" finalizará con la toma de la resolución correspondiente por parte de los encargados luego de evaluar toda la información provista. Finalmente, se ejecutará el Caso de Uso del Sistema "Entregar Préstamo", donde el Área de Atención al cliente se comunicará con el mismo para informar de la resolución, si la resolución es sido favorable el cliente concurrirá al banco a retirar el efectivo solicitado y firmar los pagaré correspondientes, si no únicamente se registrará el rechazo de la solicitud con la evaluación realizada.

Los Casos de Uso del Negocio serán los candidatos a convertirse en servicios orquestadores de servicios, o centrados en procesos en una clasificación posible, donde el flujo del procedimiento se defina según el relevamiento realizado, y los distintos pasos de la ejecución del mismo sean realizados por operaciones correspondientes a servicios de menor granularidad que realizarán los Casos de Uso del Sistema identificados.

5.3.2. Disciplina Diseño

El propósito de la disciplina de Diseño en la metodología SOA definida agrega los siguientes objetivos a los definidos en el proceso base: identificar y catalogar los servicios necesarios para

cumplir con los procesos identificados de la Organización en los Casos de Uso del Negocio definidos especificar estos servicios definiendo las interfaces y sus operaciones, así como los componentes que los implementarán y la reutilización de otros servicios y componentes dentro de la Organización.

Un aspecto importante en esta disciplina es definir la secuencia de invocaciones a servicios necesaria para la ejecución de los procesos del Negocio identificados como Casos de Uso del Negocio, en una orquestación de servicios si el proceso se encuentra bajo el control de la Organización, o en una coreografía de servicios si el proceso corresponde a más de una Organización. En esta propuesta sin embargo, no se incluye una actividad específica para la definición de coreografías, si para la definición de orquestaciones. Para definir las orquestaciones será importante también propender a la utilización de un Business Process Management System (BPMS) donde realizar la definición del flujo del proceso y su ejecución en un motor de procesos.

En esta disciplina se mantiene como en el proceso base como rol principal el Arquitecto, y participan también los Analistas y los Especialistas Técnicos, donde los primeros aportan el conocimiento de los procesos del Negocio especificados como Casos de Uso del Negocio, y la traducción en funcionalidades del sistema especificadas como Casos de Uso del Sistema, y los segundos aportan el conocimiento de las tecnologías que se utilizarán, tanto en sus posibilidades como en sus restricciones para el diseño del sistema. El entregable que se obtiene de las actividades agregadas en esta Disciplina es el Modelo de Servicios con las distintas secciones que lo componen, a partir del cual se agrega a su vez la Vista de Servicios al Documento de Arquitectura de Software (SAD) que es uno de los entregables principales de la Disciplina Diseño en el proceso base.

A continuación se describen las actividades propuestas en la Disciplina Diseño, en 5.3.2.1 se presenta la actividad Identificar y categorizar servicios (D6), en 5.3.2.2 se presenta la actividad Especificar servicios (D7), en 5.3.2.3 se presenta la actividad Investigar servicios existentes (D8), en 5.3.2.4 se presenta la actividad Asignar servicios a componentes (D9), y en 5.3.2.5 se presenta la actividad Definir orquestación de servicios (D10).

En cada caso se presenta el Objetivo de la actividad, su descripción, los roles involucrados y los entregables de entrada y salida asociados con la realización de la misma, una figura que muestra gráficamente el modelado de la actividad, y por último la fundamentación sobre la definición de dicha actividad en la metodología SOA propuesta. Finalmente en 5.3.2.6 se muestra la relación entre las actividades agregadas en la Disciplina Diseño en la metodología SOA y el proceso base.

5.3.2.1. Identificar y categorizar servicios (D6)

Objetivo: Esta actividad tiene como objetivo identificar los servicios necesarios para realizar los procesos del negocio, clasificándolos según el tipo de servicio: básicos, intermediarios, centrados en procesos y públicos empresariales.

Descripción: Teniendo en cuenta la Descripción de la Arquitectura en sus vistas de Casos de Uso del Sistema y Lógica, se realiza en primer lugar un análisis de los subsistemas definidos, asegurándose que existe un mapeo de por lo menos un subsistema a cada área funcional del Negocio, por ejemplo clientes. De la misma forma se identifica claramente en que subsistema se realiza cada Caso de Uso del Sistema definido, comenzando con los Casos de Uso identificados como relevantes a la Arquitectura. A partir de esta correspondencia se identifican los servicios que deben ser provistos por cada subsistema para realizar los distintos Casos de Uso del Sistema, y seguidamente los Casos de Uso del Negocio correspondientes a los mismos. Se brinda una guía para clasificar los servicios en básicos, intermediarios, centrados en procesos y públicos empresariales, como apoyo para realizar la identificación de los mismos.

Roles: El rol responsable de esta actividad es el Arquitecto y participan también los roles de Analista y Especialista Técnico, como forma de conjuntar el conocimiento del Negocio y los Requerimientos del sistema, con el conocimiento técnico de las herramientas a utilizar.

Entregables de entrada: Esta actividad tiene como entrada los entregables Descripción de la Arquitectura (SAD) como está definido en el proceso base, y el Modelo de Casos de Uso del Negocio, para contar con la información técnica del software en diseño y los procesos del negocio identificados para la Organización.

Entregables de salida: Como salida tiene el Modelo de Servicios con la sección correspondiente a la identificación y categorización de servicios informada.

Modelado de la actividad: en la figura 7.2 se muestra el modelado de la actividad utilizando los elementos de modelado provistos por el RUP.

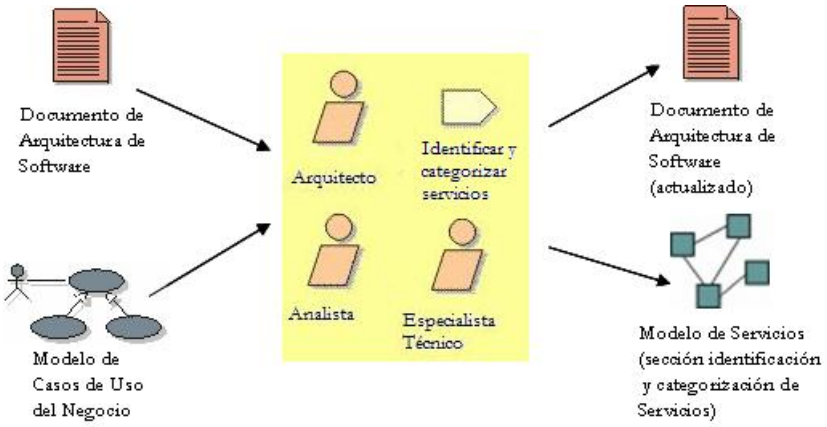


Figura 5.7: Modelado de la actividad Identificar y Categorizar Servicios (D6) en la Extensión SOA

Guías para identificación y categorización de servicios: para esta actividad se brinda material de apoyo que provee una clasificación posible para los servicios, su ubicación en la jerarquía de capas de servicios para el diseño, y la relación entre estos. Esta guía corresponde a la clasificación definida en [KRAF05]. Es importante destacar que así como es posible utilizar la clasificación provista en esta guía, también es posible utilizar otra clasificación que se desee incorporar o adaptar la provista, ya que los elementos de la metodología SOA propuesta no referencian ninguna clasificación en particular, sino que enfatizan la necesidad de utilizar alguna.

Fundamentación: Esta actividad se agrega ya que resulta imprescindible para desarrollos con enfoque SOA identificar y categorizar servicios para basar el desarrollo en servicios. Es una de las actividades medulares, sino la más importante, a realizar en desarrollos con enfoque SOA. Como se describe en los conceptos teóricos presentados del enfoque SOA en el Capítulo 4, los servicios representan la provisión de funcionalidades asociada a los conceptos del negocio, que permite introducir una capa de abstracción que sirve de intermediaria entre los procesos del Negocio y las tecnologías que los implementarán.

La categorización de los mismos permite ordenar conceptual y jerárquicamente la ubicación de estos servicios en la capa de abstracción de servicios, de forma de guíar el desarrollo y evitar la proliferación de servicios que no tengan la granularidad adecuada. La correcta identificación y categorización de servicios es fundamental para desarrollos con enfoque SOA, para obtener productos correctamente modularizados en servicios que luego serán implementados por componentes en las tecnologías y formas definidas en la Organización. Estos servicios corresponderán a servicios de menor granularidad adecuados a la realización de los Casos de Uso del Sistema definidos, y a procesos del Negocio enteros cuando orquestan otros servicios para realizar los Casos de Uso del Negocio.

Un aspecto importante a tener en cuenta en la identificación y categorización de servicios es el bajo acomplamiento que debe existir entre estos, por lo que esta definición debe realizarse con sumo cuidado para permitir obtener servicios autocontenidos que no dependan de otros para

realizar sus funciones (o de los menos posibles) para maximizar la reutilización de las funcionalidades que proveen. Realizar esta identificación y categorización de servicios solo aporta a desarrollos con enfoque SOA, esto es, para otro tipo de desarrollos no tiene sentido realizar esta actividad. Constituye además el primer paso enfocado en servicios que se propone en la metodología SOA, a partir de cuyos resultados se realizarán todas las actividades siguientes propuestas, por lo que se incluye como una actividad específica en la metodología SOA propuesta.

5.3.2.2. Especificar servicios (D7)

Objetivo: Esta actividad tiene como objetivo especificar los servicios identificados, definiendo los contratos de servicio para cada uno incluyendo las interfaces que brindará y sus operaciones, parámetros, etc.

Descripción: Para cada servicio identificado se realiza el contrato funcional especificando para cada interface definida los métodos que deben ser implementados para proveer el servicio acordado para la interface. Para cada operación en la interface se debe especificar la información siguiente: a) nombre del método, b) parámetros requeridos por el método, y para cada parámetro el nombre, tipo y descripción, c) valor retornado, indicando el nombre, tipo y descripción, d) lista de excepciones levantadas, e) breve descripción de la funcionalidad provista, f) precondiciones requeridas para la ejecución exitosa de la operación, g) post-condiciones que serán válidas luego de la ejecución del método.

Roles: El rol responsable de esta actividad es el Arquitecto y participa también el rol de Analista, ya que luego de definidos los servicios la especificación puede ser realizada por ambos.

Entregables de entrada: Esta actividad tiene como entrada el Modelo de Servicios, que tendrá por lo menos la sección de identificación y categorización de servicios informada.

Entregables de salida: Como salida tiene el Modelo de Servicios con la sección correspondiente a la especificación de servicios informada.

Modelado de la actividad: en la figura 5.8 se muestra el modelado de la actividad utilizando los elementos de modelado provistos por el RUP.

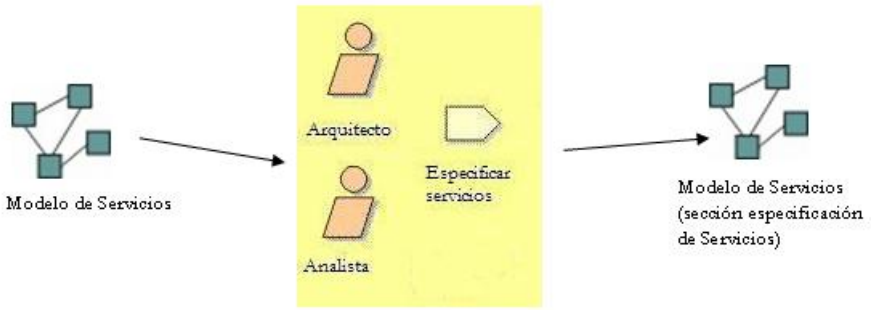


Figura 5.8: Modelado de la actividad Especificar Servicios (D7) en la Extensión SOA

Fundamentación: Esta actividad constituye el siguiente paso en el enfoque de servicios planteado por la metodología SOA propuesta, donde luego de ser identificados y categorizados los servicios, se deben especificar las funcionalidades que provee cada uno. Como se describe en los conceptos teóricos presentados del enfoque SOA en el Capítulo 4, la especificación de servicios en los contratos de servicios es una parte fundamental para definir las relaciones entre estos, mediante la identificación de operaciones, parámetros, etc. que proveen en sus interfaces para ser utilizadas por otros servicios.

Luego, la publicación de estos contratos por los proveedores de servicios, permitirá que estos sean reutilizados por otros desarrollos incluyendo la invocación a los mismos según el contrato

definido para cada uno. La especificación de estos contratos permite también razonar sobre la adecuación de la definición de servicios realizada, y eventualmente realizar ajustes cuando no se alcanzan los objetivos planteados mediante la definición de servicios realizada. Este enfoque es similar al enfoque de orientación a objetos y componentes de definir interfaces que luego sean implementadas por los elementos correspondientes, y constituye una buena práctica de diseño que se debe incorporar obligatoriamente en desarrollos con enfoque SOA.

Realizar esta especificación de servicios en los contratos correspondientes permite abstraer la definición de servicios de los componentes que los implementarán, permitiendo por lo tanto que las implementaciones puedan ser cambiadas sin afectar las interfaces de los mismos, que constituyen la presentación de los servicios al resto del software. Es el segundo paso enfocado en servicios que se propone en la metodología SOA, que permite especificar completamente las funcionalidades que proveerá cada servicio. No parece adecuado dejar liberado al equipo de desarrollo la necesidad de realizar esta especificación según el proyecto, por lo que se incluye como una actividad específica en la metodología SOA propuesta, de forma que se realice obligatoriamente la especificación mediante contratos de los servicios identificados y categorizados previamente.

5.3.2.3. Investigar servicios existentes (D8)

Objetivo: Esta actividad tiene como objetivo encontrar servicios que ya estén implementados en la plataforma SOA y puedan o deban ser reutilizados en la aplicación en desarrollo. En general los servicios que serán más reutilizados son los servicios básicos.

Descripción: Una vez que los servicios necesarios han sido identificados y especificados, se deben investigar los servicios con que se cuenta en la Organización, tanto básicos como de otros niveles, para reutilizar en la aplicación en desarrollo. Puede suceder que se deban implementar servicios intermediarios para reutilizar otros servicios que no se adapten totalmente a los requerimientos de la aplicación lo cual será preferible a implementar nuevamente el servicio desde cero. Los servicios básicos como los de acceso a datos, autenticación, seguridad, sesión, etc. deberán ser reutilizados por todos los servicios definidos en niveles superiores de la SOA.

Roles: El rol responsable de esta actividad es el Arquitecto y participa también el rol de Analista, ya que la investigación de servicios existentes tiene que ver con el modelado del negocio realizado.

Entregables de entrada: Esta actividad tiene como entrada la Evaluación de la Organización Objetivo realizada, donde están indicadas las capacidades existentes en la Organización.

Entregables de salida: Como salida tiene el Modelo de Servicios con la sección correspondiente a los servicios existentes informada.

Modelado de la actividad: en la figura 7.3 se muestra el modelado de la actividad utilizando los elementos de modelado provistos por el RUP.

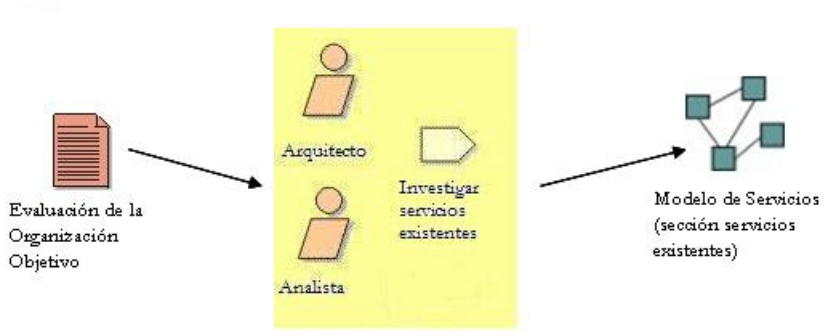


Figura 5.9: Modelado de la actividad Investigar Servicios existentes (D8) en la Extensión SOA

Fundamentación: Esta actividad constituye un paso importante en el enfoque de servicios planteado por la metodología SOA propuesta, donde se maximizan los esfuerzos realizados por la Organización en la definición de servicios para toda la Organización. Como se describe en los conceptos teóricos presentados del enfoque SOA en el Capítulo 4, una vez que la Organización ha aplicado en forma sistemática el enfoque a servicios para la informatización de sus procesos del Negocio, se contará con servicios de granularidad fina y gruesa que proveen funcionalidades relacionadas con el Negocio de la Organización.

Estos servicios ya existentes deben ser tenidos en cuenta para nuevos desarrollos, de forma de reutilizar el diseño e implementación realizado en otros desarrollos previos, evitando la duplicación de definiciones y código para las mismas funcionalidades. Puede pasar que no se encuentre un servicio que provea exactamente las mismas funcionalidades que se necesitan en el desarrollo en curso, en este caso es importante reutilizar lo máximo que se pueda del servicio existente, extendiendo el contrato del servicio para que provea las funcionalidades que son necesarias en este desarrollo.

Realizar esta búsqueda de servicios existentes permite reducir la cantidad de servicios existentes en la Organización, minimizando los costos de los nuevos desarrollos y maximizando la reutilización de desarrollos previos. Este enfoque constituye una buena práctica de la Ingeniería de Software que no siempre es realizada adecuadamente o no es realizada en el contexto de una Organización que tiene cortes verticales de desarrollo. El enfoque SOA constituye por el contrario, un enfoque horizontal de reutilización de componentes, donde en las capas inferiores se cuenta con servicios de menor granularidad que pueden y deben ser reutilizados en desarrollos posteriores.

Es el tercer paso enfocado en servicios que se propone en la metodología SOA, que promueve en forma explícita la reutilización de servicios existentes, lo que es fácilmente realizable dada la independencia que, a diferencia de la reutilización de componentes, brinda la publicación de los contratos de servicios y los componentes que los implementan por debajo. Es por esta razón que se incluye como una actividad específica en la metodología SOA propuesta, de forma que se realice obligatoriamente la investigación de servicios existentes en la Organización. Eventualmente se podrían investigar también servicios existentes en otras organizaciones para ser incorporados en coreografías de servicios, pero en la metodología SOA propuesta el alcance del enfoque SOA está restringido a la Organización de desarrollo, y no se trata el tema de coreografías de servicios.

5.3.2.4. Asignar servicios a componentes (D9)

Objetivo: Esta actividad tiene como objetivo definir los componentes que deberán ser implementados para proveer los servicios especificados, esta correspondencia no tiene porque ser 1:1 con los subsistemas identificados previamente. Asimismo se debe especificar para cada componente los servicios que provee.

Descripción: Para asignar servicios a componentes se debe responder la pregunta de quién (cual componente) proveerá la implementación y gestión de los servicios definidos según las interfaces especificadas. Se debe tener en cuenta los servicios existentes identificados si existen componentes que los brinden o funcionalidades existentes para las cuales se pueda crear algún servicio intermediario que las provea. Para cada componente necesario se realiza la especificación del componente, indicando reglas del Negocio y servicios que implementa, así como otros componentes que utiliza.

Roles: El rol responsable de esta actividad es el Arquitecto y participa también el rol de Especialista Técnico, ya que involucra mapeos del diseño a la implementación.

Entregables de entrada: Esta actividad tiene como entrada el Modelo de Diseño y el Modelo de Servicios, con la información del software diseñado y los servicios identificados, categorizados, especificados y existentes.

Entregables de salida: Como salida tiene el Modelo de Implementación y el Modelo de Servicios, con la asignación de componentes a servicios correspondiente en cada caso.

Modelado de la actividad: en la figura 5.10 se muestra el modelado de la actividad utilizando los elementos de modelado provistos por el RUP.

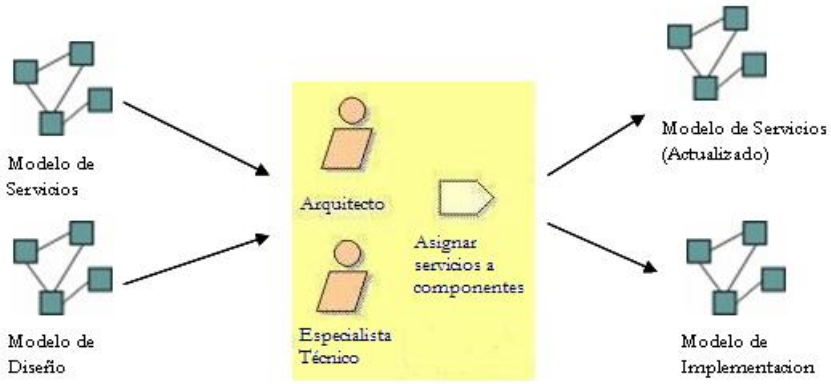


Figura 5.10: Modelado de la actividad Asignar Servicios a componentes (D9) en la Extensión SOA

Fundamentación: En esta actividad se deben definir y asignar los componentes que implementarán los servicios especificados en el Modelo de Servicios. Esta componentización de servicios estará dada por el entorno de programación que se elija, y pueden ser clases en Java, componentes J2EE o .NET, componentizaciones de sistemas legados, procesos definidos en Business Process Modeling Language (BPML) específicamente en BPEL, entre otros. El enfoque fundamental es que la definición de servicios está separada de los componentes que los implementan, permitiendo la integración de diferentes tecnologías.

Esta actividad constituye otro aspecto importante del enfoque a servicios planteado por la metodología SOA propuesta, y que fuerza en forma explícita la separación entre la definición de los servicios y la implementación de los mismos. Este hecho, de la mano del resto de las actividades propuestas, fuerza a la definición e implementación de servicios que cumplen con las características del enfoque SOA presentado en 4, y permite la obtención de los objetivos planteados por el paradigma.

Es el cuarto paso enfocado en servicios que se propone en la metodología SOA, que prescribe la asignación de componentes a servicios para su implementación, aportando a la separación entre los elementos del Negocio que deben realizar los servicios definidos, y las plataformas tecnológicas que se utilicen en la Organización. Esto permitirá evolucionar la implementación de servicios sin afectar las funcionalidades que éstos proveen, especificadas en el contrato correspondiente. Es por esta razón que se incluye como una actividad específica en la metodología SOA propuesta, de forma que se realice obligatoriamente la separación entre la especificación de los servicios y los componentes que los implementan.

5.3.2.5. Definir orquestación de servicios (D10)

Objetivo: Esta actividad tiene como objetivo definir la secuencia de interacción entre servicios que es necesaria para realizar los procesos del Negocio identificados y descritos como Casos de Uso del Negocio.

Descripción: La orquestación de servicios para la implementación de procesos del Negocio se muestra para cada proceso en un diagrama de secuencia que describa la interacción entre los distintos servicios involucrados. Además se utilizará preferiblemente una herramienta BPMS

(Bussiness Process Management System) para definir, ejecutar y gestionar la orquestación del servicio centrado en procesos, mediante la utilización de BPML (Bussiness Process Modeling Language) como BPELWS y el workflow de ejecución del proceso del Negocio descrito.

Roles: El rol responsable de esta actividad es el Arquitecto y participan también el rol de Analista y Especialista Técnico, ya los servicios centrados a procesos tienen un componente importante de visión del negocio y otro de visión técnica.

Entregables de entrada: Esta actividad tiene como entrada el Modelo de Casos de Uso del Negocio y el Modelo de Servicios, que tienen toda la información para definir los servicios centrados en procesos.

Entregables de salida: Como salida tiene el Modelo de Implementación y el Modelo de Servicios con los mapeos correspondientes realizados y la orquestación de servicios definida para los servicios centrados en procesos.

Modelado de la actividad: en la figura 5.11 se muestra el modelado de la actividad utilizando los elementos de modelado provistos por el RUP.

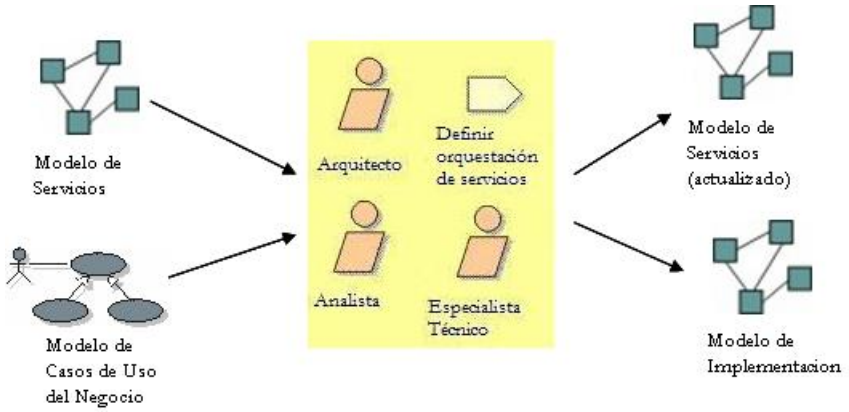


Figura 5.11: Modelado de la actividad Definir orquestación de Servicios (D10) en la Extensión SOA

Fundamentación: En esta actividad se debe definir la secuencia de invocaciones de servicios que se realiza por parte del servicio orquestador, para realizar los procesos del Negocio identificados y modelados como Casos de Uso del Negocio. Se utilizará un motor de ejecución de procesos que sea capaz de interpretar y ejecutar los pasos definidos en el proceso del Negocio modelado. Para esto los procesos serán definidos en Business Process Modeling Language (BPML) específicamente en BPEL, a partir del modelado gráfico de los mismos que convierte dicha definición la especificación en BPEL correspondiente.

La diferencia de este enfoque con la utilización de un BPMS para el modelado y ejecución integral de procesos del Negocio, lo constituye el hecho de que las herramientas utilizadas en esta actividad para realizar ese modelado son internas al desarrollo, por ejemplo mediante diagramas de actividad de UML, que permiten luego crear la definición en BPEL asociada. Este es el caso por ejemplo de la herramienta JBPM provista como parte del servidor de aplicaciones JBoss, que se utilizó en la prueba de la metodología SOA. Por otro lado, al utilizar un BPMS que permita el tratamiento integral de los procesos del Negocio, se espera que el modelado de estos procesos sea realizado por el área de Negocios de la Organización, y no por el desarrollo de software.

Esta actividad constituye otro aspecto importante del enfoque a servicios planteado por la metodología SOA propuesta, que fuerza a la realización de procesos del Negocio mediante la invocación de servicios de menor granularidad definidos, cumpliendo con las características del enfoque SOA presentado en 4. Es el quinto paso enfocado en servicios que se propone en

la metodología SOA, que presenta en forma explícita la necesidad de realizar los procesos del Negocio con servicios orquestadores de otros servicios. Es por esta razón que se incluye como una actividad específica en la metodología SOA propuesta, de forma que se realice obligatoriamente la orquestación de servicios para realizar los procesos del Negocio definidos.

No se define en esta primera propuesta una actividad para definir coreografías de servicios, ya que el enfoque de la metodología corresponde al desarrollo dentro de los límites de una única Organización. De todas formas, definir una coreografía de servicios sería similar al enfoque para la orquestación de servicios, pero habría partes del proceso (sub-procesos) cuya ejecución correspondería a otras Organizaciones con las que se interactúa, y que solamente serían vistos como caja negra, es decir reciben determinada entrada y devuelven determinada respuesta, sin importar como se realiza dicho sub-proceso en la otra Organización.

5.3.2.6. Relación e inclusión de esta Disciplina con el proceso base

La Disciplina de Diseño en el proceso base cuenta con una importante cantidad de actividades que se deben realizar en coordinación con las que plantea la metodología para el enfoque SOA. Se define la agrupación de actividades especificada como DA1 ? Definir Servicios que corresponde a la realización conjunta de las actividades D6 ? identificar y categorizar servicios, D7 ? especificar servicios y D8 ? investigar servicios existentes, ya que las tres se realizan en forma conjunta por la relación que tienen entre sí, la cual se realiza iterando con la actividad del proceso base identificada como D2 ? Describir la Arquitectura, ya que se retroalimentan entre como se explica a continuación.

La definición de la Arquitectura plantea la identificación de los Casos de Uso del Sistema relevantes a la Arquitectura, que son aquellos que por sus requerimientos tantos funcionales como no funcionales, definen el esqueleto de la aplicación a construir en cuanto a definición de subsistemas e interacción entre los mismos y con otras aplicaciones, definiendo por ejemplo la necesidad de aplicar patrones de diseño como la encapsulación del acceso a un dispositivo de hardware o a otro sistema sobre el cual no se tiene control, o el patrón de acceso a datos, entre otros.

En base a dichos Casos de Uso se realiza la modularización del software en subsistemas con alta cohesión y bajo acoplamiento que permitan identificar todas las áreas que involucra el desarrollo, ejercitando distintas hipótesis sobre el comportamiento del mismo. Es teniendo en cuenta esta modularización, más los Casos de Uso del Sistema identificados y los Casos de Uso del Negocio que describen los procesos de la Organización, más la clasificación de servicios presentada, que es posible pensar y definir los servicios en los distintos niveles, que seguirán luego el flujo de trabajo de la disciplina de Diseño en la metodología SOA planteada.

El proceso base tiene como entregables fundamentales de la Disciplina de Diseño, en primer lugar el Documento de Descripción de la Arquitectura o SAD (por sus siglas en inglés Software Architecture Document) en el cual se presentan las distintas vistas que tienen injerencia en la comprensión de la Arquitectura definida para la aplicación, siguiendo el modelo planteado en [KRUC95] que son: Vista del Modelo de Casos de Uso, del Modelo de Diseño (incluye vista lógica y de procesos), del Modelo de Implementación y del Modelo de Deployment (o Despliegue), y en segundo lugar el Modelo de Diseño donde se presenta el diseño para todos los Casos de Uso del Sistema.

La metodología agrega como entregable fundamental de esta disciplina el Modelo de Servicios donde se encuentra toda la información correspondiente a la definición de servicios, el cual acompaña al Modelo de Diseño, y se agrega en el SAD la vista del Modelo de Servicios donde se muestra, como para el resto de las vistas, la información de los servicios definidos a partir de los Casos de Uso relevantes a la Arquitectura.

Por lo tanto, en el Documento de la Arquitectura de Software de la realización de la metodología para desarrollo SOA, se encuentra la vista de Casos de Uso en primer lugar que guían la

definición de la Arquitectura, en las vistas lógica y de procesos la modularización del software en subsistemas, sus interfaces y sus relaciones, en la vista de servicios los servicios identificados como subsistemas, en la vista de deployment el despliegue del software en el hardware y en la vista de implementación la correspondencia del diseño con elementos de implementación.

Esto permite como en el planteo del proceso base, que la información relevante a la Arquitectura de Software esté concentrada en un único documento donde en base a las distintas vistas se describe el conjunto de la Arquitectura, según las distintas perspectivas. Luego en cada modelo se encuentra la información completa correspondiente a esa perspectiva para todo el sistema. Como en el proceso base, se debe prestar especial atención en mantener la correspondencia entre las vistas presentadas en el SAD y los modelos de los cuales dichas vistas son cortes, para mantener la consistencia de la documentación asociada.

Flujo de actividades para la Disciplina de Diseño

En la figura 5.12 se presenta a modo de ejemplo el flujo de actividades para la Disciplina de Diseño modelado como Diagrama de Actividad de UML. Este flujo se ejecuta cada vez que se entra a una iteración en una Fase del proceso al momento de realizar las actividades de diseño que corresponda. Según la Fase e iteración en la que se esté se realizarán todas, algunas o ninguna de las actividades definidas, según las transiciones permitidas en el flujo.

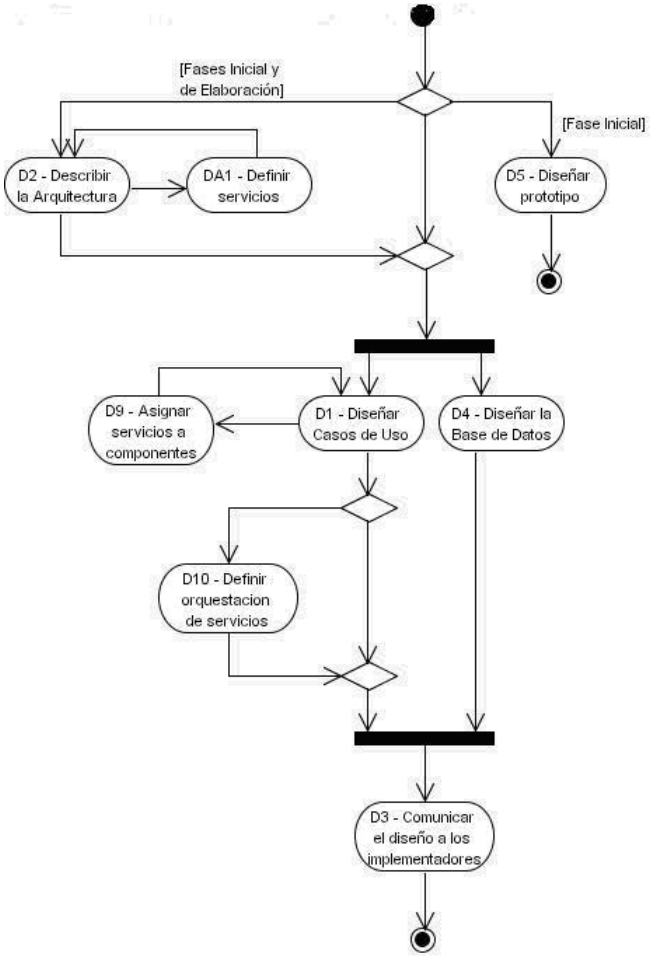


Figura 5.12: Flujo de actividades de la Disciplina Diseño en la Extensión SOA y el proceso base

En la figura 5.12 la actividad DA1 - Definir servicios corresponde a la agrupación de las actividades D6 - Identificar y categorizar servicios, D7 - Especificar servicios y D8 - Investigar servicios

existentes, ya que las tres están muy relacionadas y se realizan a la vez y conjuntamente con la actividad del proceso base D2 - Describir la Arquitectura.

El flujo definido determina que por ejemplo, la actividad D5 - Diseñar prototipo se realiza únicamente al ejecutar este flujo en la Fase inicial, a la vez que se ejecutan las actividades D2 - Describir la Arquitectura y la actividad agrupada de la metodología SOA DA1 - Definir servicios. Para el resto de las Fases es posible realizar el resto de las actividades según está modelado en el diagrama de actividad, pero no la actividad D5 que solamente se realiza en la Fase Inicial.

5.3.3. Disciplina Implementación

El propósito de la disciplina de Implementación en la metodología SOA propuesta agrega los siguientes objetivos a los definidos en el proceso base: implementar los componentes proveedores de servicios según lo establecido por la asignación de servicios a componentes implementar el ligamiento de servicios en las application frontend y en los servicios que llaman a otros servicios, utilizando la estrategia definida en la Organización. A continuación se presenta en 5.3.3.1 la actividad Implementar servicios (I13) incluyendo su objetivo, descripción, roles involucrados y entregables entrada y salida asociados con la realización de la misma, una figura que muestra gráficamente el modelado de la actividad, y finalmente la fundamentación sobre la definición de dicha actividad en la metodología SOA propuesta.

5.3.3.1. Implementar servicios (I13)

Objetivo: Esta actividad tiene como objetivo implementar los servicios definidos, para lo cual se deben tener en cuenta el tipo de servicio, las interfaces diseñadas, la interacción con otros servicios (con o sin repositorio de servicios, con ligamiento en tiempo de desarrollo o de ejecución).

Descripción: En esta actividad se deben implementar los componentes definidos para los servicios identificados, de acuerdo a las interfaces y componentes especificados en la Disciplina de Diseño. Se debe proveer para cada interface los métodos definidos respetando los parámetros (nombre, tipo) especificados, valor retornado (nombre, tipo) y la utilización de otros servicios para la realización de las operaciones definidas, según lo establecido en la especificación de las interfaces y en los componentes definidos para la implementación de servicios.

Roles: El rol responsable de esta actividad y único participante es el Implementador que programa los servicios definidos.

Entregables de entrada: Esta actividad tiene como entrada el Modelo de Implementación y el Modelo de Servicios, donde está definida la correspondencia entre el diseño y la implementación.

Entregables de salida: Como salida tiene el servicio implementado en el componente definido.

Modelado de la actividad: en la figura 5.13 se muestra el modelado de la actividad utilizando los elementos de modelado provistos por el RUP.

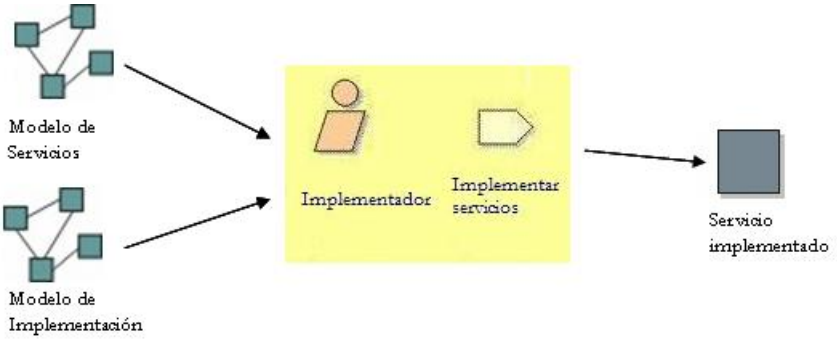


Figura 5.13: Modelado de la actividad Implementar Servicios (I13) en la Extensión SOA

Fundamentación: En esta actividad se deben implementar los componentes asignados a servicios, según fue definido en las actividades de diseño correspondientes. Se cuenta con la definición de servicios en el Modelo de Servicios, y con la definición de los componentes asignados a servicios realizada, por lo que, la implementación debería ser directa.

Esta actividad representa el resultado de las actividades definidas en la Disciplina de diseño, y fuerza a que se respete el diseño realizado de servicios y componentes correspondientes, de forma de mantener las características del enfoque SOA presentado en 4. Es por esta razón que se incluye como una actividad específica en la metodología SOA propuesta, de forma que se realice obligatoriamente la implementación de servicios en los componentes definidos, según la especificación de servicios realizada en el Modelo de Servicios generado.

5.3.4. Plantillas para los entregables en la Extensión SOA

Las plantillas de entregables son parte fundamental de la comprensión de los entregables a realizar al aplicar la metodología SOA propuesta. Además de la descripción de las actividades definidas, proveer una guía para la realización de los entregables que se deben obtener como salida de las mismas resulta un elemento de apoyo importante para su comprensión. Por claridad las plantillas provistas para la realización de los entregables definidos para la Extensión SOA se adjuntan en el Capítulo 10 - Anexo B, pero son parte integral de la metodología SOA propuesta.

5.3.5. Otras Disciplinas en la Extensión SOA

Para la definición de la propuesta de metodología SOA se estudió también el agregado o modificación de actividades, entregables y roles correspondientes a otras Disciplinas como Verificación, Gestión del Proyecto, Gestión de Configuración y Gestión de Calidad. Si bien se entiende que es importante también explicitar el enfoque SOA del desarrollo en esas Disciplinas, en el momento de la definición de la metodología, se evaluó que al realizarse una Extensión sobre el proceso base adaptación del RUP, se podía superponer el enfoque basado en Casos de Uso del Sistema para guiar el desarrollo en dichas Disciplinas, con un enfoque para guiar el desarrollo basado en servicios.

Se optó entonces por mantener el enfoque del proceso base para esas Disciplinas y realizar los agregados y modificaciones de Disciplinas, actividades, entregables y roles más importantes, que brindaran en forma explícita un enfoque SOA hacia el modelado de los procesos del Negocio, el diseño orientado a servicios y la implementación de servicios especificados. Se consideró también que su aplicación sería más controlada si las modificaciones y agregados estaban acotados en esta primera definición, dejando abierta la posibilidad de realizar ajustes y mejoras a la luz de los resultados de la aplicación del conjunto núcleo de la metodología definido inicialmente. Los elementos agregados correspondientes a los ajustes y mejoras a la metodología SOA propuesta, se presentan en el Capítulo 7.

Capítulo 6

Aplicación de la Metodología SOA propuesta

En este capítulo se presenta la aplicación de la metodología SOA propuesta para desarrollo de aplicaciones SOA (Service Oriented Architecture), en la Extensión SOA definida para el proceso base adaptación del RUP de [GRIS06]. En la sección 6.1 se describe el contexto de la prueba realizada, en la sección 6.2 se describe el desarrollo y seguimiento realizado de la prueba, en la sección 6.3 se presentan los resultados obtenidos de la aplicación de la prueba y finalmente en la seccion 6.4 se plantean algunas conclusiones sobre la prueba realizada.

Como parte de las investigaciones en el marco de la realización de este trabajo, se propone en el año 2006 el proyecto de grado "Enfoque SOA (Service Oriented Architecture) con integración de aplicaciones móviles para LIMS (Laboratory Information Management System)" [MSOA06], en el marco del convenio del Instituto de Computación (InCo) con el Instituto Pasteur Montevideo (IPMONT) [LIMS05], para investigar aspectos de SOA para LIMS, aplicando la metodología SOA propuesta y desarrollando un prototipo con la suite de herramientas IBM implementando servicios con los enfoques Service Component Architecture (SCA) [SCAS] y Service Data Object (SDO) [SDOS] presentados en el Capítulo 4. Al momento de comenzar a escribir este trabajo, dicho proyecto estaba en curso y actualmente está en etapa de finalización, por lo que no resulta posible incluir una evaluación y análisis de resultados obtenidos con la metodología, herramientas y enfoque de implementación utilizados, más que parciales, por lo que se omite su presentación en este informe.

Parte de este capítulo fue publicado como artículo y presentado en las "V Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento (JIISIC'06)" realizadas en Puebla, México, en febrero de 2006 [ESOA06], describiendo en forma resumida la metodología propuesta en la Extensión SOA y presentando resultados parciales de la aplicación de la misma obtenidos hasta ese momento, los que se presentan en forma completa en este informe de tesis.

6.1. Contexto de la prueba

La metodología propuesta en la Extensión SOA fue aplicada en la edición 2005 del curso "Proyecto de Ingeniería de Software" que dicta el Grupo de Ingeniería de Software (Gris), donde dos grupos desarrollaron una aplicación de Help-Desk para el Proyecto Link-All de la Unión Europea y el Instituto de Computación.

Este proyecto fue elegido para realizar la aplicación de la metodología SOA debido a que el desarrollo existente en la plataforma Link-All ya tenía una orientación a servicios en cuanto a la definición de servicios básicos de infraestructura, por lo que resultaba interesante realizar

una prueba del enfoque prototipando una aplicación que utilizara servicios existentes y fuera desarrollada también con orientación a servicios. Cabe aclarar que el desarrollo orientado a servicios del proyecto Link-All no se había realizado siguiendo ninguna metodología de desarrollo específica de orientación a servicios.

El Proyecto Link-all está integrado por diecisiete socios de Sudamérica y Europa, está dirigido a asistir en la apropiación de prácticas de tecnologías de la información y comunicaciones (TIC), apoyar la colaboración, promover el intercambio de experiencias, la transferencia de conocimiento y mejorar habilidades en tres sectores objetivo en Latinoamérica: artesanía, eco-agro turismo y cultura. El Sistema Link-all incluye herramientas diseñadas e integradas para proveer de una serie de facilidades de inclusión tecnológica, que fortalezcan la integración de actividades de desarrollo local que posibiliten y faciliten la inserción exitosa de comunidades remotas en el mercado global, a través del desarrollo de una red transnacional europeo-latinoamericana de cooperación.

El sistema brinda servicios a múltiples usuarios distribuidos geográficamente y que poseen un bajo conocimiento informático, por lo que es importante contar con múltiples canales de comunicación y apoyo al usuario. La aplicación de Help-Desk, herramienta de gestión de incidentes y nuevos requerimientos, para apoyar a los usuarios en el uso y extensión del sistema, fue la propuesta de aplicación a desarrollar en el curso "Proyecto de Ingeniería de Software" durante el segundo semestre 2005 y que constituye el caso de estudio de la aplicación de la metodología SOA definida.

6.1.1. Principales requerimientos del producto

El sistema Help-Desk se debía integrar completamente al sistema Link-all, consumiendo los servicios de infraestructura que este brinda, de forma de compartir por ejemplo, el esquema de seguridad. Como objetivo principal se requería gestionar de forma sistemática y eficiente tanto los incidentes en el sistema como los nuevos requerimientos, confeccionando una base de conocimientos que pueda ser explotada por el usuario final, donde estén tipificados los problemas y soluciones o alternativas (workarounds) encontrados. A su vez el Help-Desk debía ofrecer servicios que pudieran ser consumidos por otras aplicaciones u herramientas, por ejemplo que el Help Contextual existente pudiera consumir servicios de la base de conocimientos para proveer ayuda sobre errores conocidos.

La comunicación con los usuarios es fundamental en el Sistema Link-all, por lo que el Help-Desk debía permitir mantener contactos tanto sincrónicos como asincrónicos con los mismos, manteniendo al usuario informado del estado de su incidente, vía mail de hitos importantes y de una interfaz grafica integrada al sistema de los eventos significativos, además de brindar la posibilidad de mantener una conversación entre el usuario y el técnico encargado del incidente, mediante alguna herramienta de Chat. Para el desarrollo del Help-Desk se definió la incorporación de nuevas tecnologías, teniendo en cuenta los objetivos de bajo costo de licenciamiento y compatibilidad con otros productos planteados por el sistema Link-all.

Entre las tecnologías que se debían utilizar se pueden destacar el JBoss jBPM [JBPM] como motor de workflow y BPM, que permite la creación de procesos de negocio que se coordinan entre personas, aplicaciones y servicios, y provee un framework para crear, coordinar y monitorear procesos de negocios, principalmente para definir y gestionar el flujo que debían seguir los incidentes luego de haber ingresado al sistema. En las interfaces de usuario se planteaba la implementación de AJAX [AJAX], técnica de desarrollo web para aplicaciones interactivas mediante la combinación de otras tecnologías web existentes.

6.2. Desarrollo y seguimiento de la prueba

El desarrollo del Help-Desk estaba previsto para ser llevado adelante por dos grupos de 11 estudiantes cada uno, grupos 4 y 5 [GR0405] de la edición 2005 del curso "Proyecto de Ingeniería

de Software", siguiendo el proceso de desarrollo definido por la Extensión SOA. En forma previa al comienzo del proyecto, antes de la Fase Inicial, se realizó una presentación a ambos grupos del enfoque SOA y los agregados de la Extensión SOA para explicar la metodología y forma de trabajo que deberían seguir de forma de encarar el desarrollo orientado a servicios de la aplicación propuesta.

Principalmente los roles involucrados en las Disciplinas, Actividades y Entregables que se agregaban, esto es los Analistas, el Arquitecto, Especialistas Técnicos e Implementadores, debían estudiar los conceptos del enfoque y los agregados correspondientes a la Extensión SOA, brindados en el sitio Web agregado al proceso base en [WTAD05]. Este sitio Web se presenta en el capítulo - 9 - Anexo A y contiene la definición de las Disciplinas, Actividades, Entregables presentada en el capítulo 5, las plantillas para los Entregables que se adjuntan en el capítulo 10 - Anexo B, y material de estudio asociado al enfoque SOA. Fue realizado como parte del trabajo de preparación, definición y prueba de la metodología SOA en la Extension SOA.

Ambos grupos eran dirigidos por el docente responsable de la definición de la metodología de la Extensión SOA y tenian como clientes a dos integrantes del Proyecto Link-All. Según está estipulado en el proceso base, las reuniones con el cliente son frecuentes al comienzo para el relevamiento de requerimientos, y más espaciadas cuando el grupo comienza a tener liberaciones, que son presentadas al cliente y validadas con él. En las reuniones con el cliente se agrega en la Extensión SOA la realización de las actividades y entregables correspondientes al Modelado del Negocio, como primer acercamiento a la identificación y especificación de los procesos del Negocio previo a la especificación de las funcionalidades asociadas a la aplicación. El director de proyecto está en estrecho contacto con el cliente para hacer el seguimiento del desarrollo y sus validaciones y contar con la perspectiva del cliente sobre el desarrollo del proyecto, además de la del grupo.

Durante la Fase Inicial, ambos grupos estudiaron los materiales de la Extensión SOA, además de los previstos normalmente en el curso sobre el proceso de desarrollo, actividades, entregables y roles. Sin embargo a principios de la Fase de Elaboración el grupo 4 planteó la posibilidad de no realizar el enfoque de diseño orientado a servicios, dado que con el cliente definieron que el prototipo de dicho grupo se enfocara hacia la realización de la interfaz de usuario del Help-Desk con AYAX como aspecto principal del mismo. El énfasis del desarrollo para el grupo 5 quedaría entonces en el prototipado del Help-Desk orientado a servicios para ser integrado consumiendo servicios de la plataforma Link-all. Dado que tanto el cliente como los grupos estaban de acuerdo en dichas definiciones, se optó por liberar al grupo 4 de la aplicación de la Extensión SOA en su proyecto, por lo que la prueba se continuó solamente con el grupo 5.

En lo que sigue se presenta el seguimiento realizado del desarrollo del proyecto, utilizando las herramientas estipuladas en el proceso base según las definiciones realizadas en su primer adaptación en [ADBP00]. En 6.2.1 se describe la revisión de entregables, en 6.2.2 los monitoreos del grupo con el Director de Proyecto, en 6.2.3 las Auditorías al proceso realizadas y en 6.2.4 los cuestionarios finales a los estudiantes y cliente realizados.

6.2.1. Revisión de entregables

Los grupos realizan la entrega semanal como parte de las actividades de evaluación del curso, y directamente también al director de proyecto, para que se pueda realizar la revisión de entregables en forma previa al monitoreo. Esta revisión consiste en relevar en primer lugar que los entregables que se debían obtener en dicha semana están presentes, lo que en principio indica que las actividades correspondientes fueron realizadas.

En segundo lugar se revisa el contenido de los entregables más importantes para comprobar el apego a las plantillas del proceso y que el avance es el que corresponde para la semana que está siendo evaluada. En el caso de la Extensión SOA se agregaron entregables correspondientes a las actividades incorporadas, por lo que éstos eran los principales entregables a revisar. Las plantillas realizadas para dichos entregables se pueden consultar en el Capítulo 10 - Anexo B.

Los entregables de la Extensión SOA que se agregan son los correspondientes a las Disciplinas de Modelado del Negocio y Diseño, donde se encuentran las actividades que componen el núcleo de la metodología, como se describió en 5. Dichos entregables tienen definida la Fase, iteración y semana en que deben ser entregados, según la realización de actividades que los generan. Durante el desarrollo del proyecto esta planificación se vió desfasada por los atrasos que el grupo tuvo debidos a diversas razones que afectaron la realización en tiempo y forma de las actividades y entregables previstos. En la tabla 6.1 se muestra la planificación de entregables por Fase, iteración y semana prevista en la Extensión SOA, y el desfasaje ocurrido en la entrega de los mismos.

Fase	Iteración	Semana	Disciplina	Entregable	Responsable	Prioridad	Entrega
Inicial	Iteración 1	1	Modelado del Negocio	Evaluación de la Organización Objetivo	Arquitecto	Media	SI
Inicial	Iteración 1	1	Modelado del Negocio	Modelo de Casos de Uso del Negocio	Arquitecto	Media	SI
Inicial	Iteración 2	3	Modelado del Negocio	Evaluación de la Organización Objetivo	Arquitecto	Alta	NO
Inicial	Iteración 2	3	Modelado del Negocio	Modelo de Casos de Uso del Negocio	Arquitecto	Alta	SI
Inicial	Iteración 2	4	Diseño	Modelo de Servicios	Arquitecto	Media	NO
Elaboración	Iteración 1	5	Modelado del Negocio	Modelo de Casos de Uso del Negocio	Arquitecto	Media	NO
Elaboración	Iteración 1	6	Diseño	Modelo de Servicios	Arquitecto	Alta	NO
Elaboración	Iteración 2	8	Diseño	Modelo de Servicios	Arquitecto	Alta	SI
Construcción	Iteración 1	10	Diseño	Modelo de Servicios	Arquitecto	Media	Semana12

Cuadro 6.1: Planificación entregables de la Extensión SOA y entrega real

6.2.2. Monitoreos con el Director del proyecto

El director de proyecto tiene contacto con los grupos una vez por semana, donde se revisa el trabajo realizado en la semana previa y se discute la planificación de la semana siguiente, dentro de las Fases e iteraciones que indica el proceso base. A estas reuniones asisten en forma obligatoria los roles de Arquitecto, Administrador, Responsable de Verificación y Responsable de Calidad, pero están abiertas a que cualquier rol que necesite tener contacto con el director pueda ir cuando lo considere.

Para poder realizar el seguimiento del grupo, como ya fue mencionado, el director revisa en forma previa a la reunión los entregables más importantes de la semana, para detectar desviaciones en la realización de actividades y contenido de los entregables, y evaluar el avance del proyecto que luego se discute con el grupo. Al finalizar cada Fase está estipulada una reunión de todo el grupo con el director donde el grupo presenta la evaluación que hizo de la Fase y se verifica la obtención de los objetivos planteados para realizar el pasaje a la siguiente Fase. En caso que no todos los objetivos se hayan alcanzado se debe continuar el trabajo en la Fase hasta concretar dichos objetivos, para poder encarar las actividades de la siguiente.

En el caso de la Extensión SOA el rol del director de proyecto lo cumplía el docente responsable de la definición de la Extensión SOA, por lo que además de las actividades propias del rol, desempeñó tareas también de capacitador en cuanto a aspectos técnicos del enfoque, como soporte para la definición de los servicios, comprensión de los conceptos asociados, explicación de las actividades a realizar y como se debían realizar, y contenido esperado de los entregables según las plantillas brindadas. Principalmente en lo que tiene que ver con las tareas del Arquitecto y Analistas, los monitoreos de seguimiento se extendían luego de las actividades normales del mismo, para tratar únicamente temas asociados al enfoque SOA evacuando las dudas generadas y brindando guía para continuar el trabajo según lo previsto.

La Fase Inicial estaba previsto que se realizara en dos iteraciones de dos semanas cada una, en las cuales el grupo tiene dos reuniones semanales con el Cliente para realizar el relevamiento de requerimientos, y en el caso de la Extensión SOA también el modelado del Negocio. Sin embargo al evaluar la obtención de objetivos para la Fase Inicial en el hito previsto para su

finalización, se vió que no habían sido conseguidos los más importantes en forma completa. Se estimó finalizar con la Fase Inicial en el correr de la semana siguiente, pero esto tampoco fue conseguido por lo que finalmente se pudo completar la misma una semana después de lo previsto.

Los problemas principales se debieron a no tener completa la documentación correspondiente a la definición de la Arquitectura inicial y por lo tanto, el diseño de los Casos de Uso para comenzar la implementación en la Fase de Elaboración, y la definición del Alcance del proyecto, que no pudo ser acordado con el Cliente en los tiempos previstos. La inexperiencia del grupo tanto en el enfoque SOA para identificación de servicios en el diseño y utilización de los servicios brindados por la plataforma del Link-all, como en la gama de nuevas tecnologías a utilizar, como ser el JBPM para la definición de flujos de la aplicación con orquestación de servicios, fueron las causas principales de estos atrasos.

En la Fase de Elaboración se realizó la implementación de la Línea Base de la Arquitectura pero con bastante dedicación horaria de los implementadores, a pesar de lo cual no fue posible absorber el atraso que se tenía desde la Fase Inicial, el cual fue incrementado en una semana más al atrasarse la obtención del Ejecutable de la Línea Base de la Arquitectura. En la Fase de Construcción se siguió con el esfuerzo de implementación, la primer iteración tuvo la duración prevista y la segunda se acortó a una semana y media para poder tener unos días de Fase de Transición en la cual implantar la versión para pruebas de aceptación en el ambiente del Cliente.

En la tabla 6.2 se presenta la duración prevista por el proceso para las fases e iteraciones según lo establecido en 5.2 del 5 y la duración real del proyecto en la prueba con el grupo 5.

Fases	Inicial				Elaboración				Construcción				Transición	
Iteraciones	Iter. 1		Iter. 2		Iter. 1		Iter. 2		Iter. 1		Iter. 2		Iter. 1	
Semanas	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Real	Inicial				Elaboración				Construcción				Tran.	

Cuadro 6.2: Duración prevista y real de las Fases e iteraciones en la prueba de la Extensión SOA

6.2.3. Auditorías al proceso de desarrollo

Las Auditorías al proceso de desarrollo son parte de las actividades previstas en el proceso de desarrollo base, para obtener una visión complementaria a la del director del proyecto sobre el desarrollo del mismo. La forma en que están definidas, los cuestionarios previos a la reunión que se envían a los estudiantes, las semanas en que se realizan y las Disciplinas a auditar, entre otros elementos, se definieron para la edición del año 2000 del curso, como parte del trabajo realizado en [ADBP00].

Según [ADBP00] "La IEEE define una Auditoria como "la revisión independiente de un producto o de un conjunto de productos para evaluar el cumplimiento con las especificaciones, estándares, acuerdos contractuales u otros criterios" [IEEE90]. Una Auditoria tiene como principal propósito la revisión de las actividades de Ingeniería de Software para verificar su ajuste al proceso de software definido, verificando que las desviaciones del proceso y los productos del software se documenten y gestionen de acuerdo con un procedimiento establecido, aplicándose en varias etapas del proceso de Software."

Las Auditorías son realizadas por los docentes del curso, intercambiando grupos, por lo que un docente nunca hace Auditorías de los grupos que dirige. Al finalizar cada fase se realiza una Auditoría con roles predefinidos, que están seleccionados según el énfasis de trabajo en la fase, por ejemplo para la Fase de Elaboración que tiene como objetivo principal la obtención del Ejecutable de la Línea Base de la Arquitectura, es importante contar con la visión y asistencia del rol Arquitecto.

En forma previa a la realización de la Auditoría se envían cuestionarios para cada rol asistente, los cuales deben ser devueltos contestados unos días antes de la reunión. Esto permite procesar las respuestas también en forma previa, identificando aquellos puntos que es importante tratar con mayor profundidad en la reunión, ya sea porque plantean discrepancias en las respuestas o no quedan claras las mismas o indican un posible desvío al proceso.

Se realizaron dos Auditorías al grupo durante la prueba de la Extensión SOA, una al finalizar la Fase Inicial y otra al finalizar la Fase de Elaboración. Los roles participantes en las Auditorías son elegidos de acuerdo a las Disciplinas cuyo trabajo reviste la mayor importancia en la Fase que se va a auditar, en la primera esas Disciplinas son Requerimientos, Diseño y las de Gestión: del Proyecto, Calidad y Configuración, por lo que participaron los roles Analistas, Arquitecto, Responsable SQA, Responsable SCM y Administrador. En la segunda las Disciplinas más importantes son Diseño, Implementación, Verificación, y las de Gestión: del Proyecto y Configuración, por lo que participaron los roles Arquitecto, Implementadores, Responsable de Verificación, Responsable SCM y Administrador.

Para la realización de ambas Auditorías se enviaron además de los cuestionarios para los roles participantes, cuestionarios genéricos para todo el grupo, de forma de complementar la información de la Fase con la visión del grupo entero sobre los aspectos más importantes. Luego de realizada la Auditoría el equipo de auditores realiza un resumen de la reunión, que se envía a los estudiantes para corroborar que el resumen es adecuado, y luego son enviados a los Directores de proyecto para que cuenten con una visión externa del desarrollo del proyecto y los problemas detectados, que aporte a la mejora del mismo.

6.2.4. Cuestionarios finales

Al finalizar el curso se realizan dos tipos de cuestionarios finales, a los estudiantes y a los clientes. En cada uno se presentan distintas preguntas asociadas al producto en desarrollo y al proceso utilizado, para su evaluación. Posteriormente se procesan las respuestas de los cuestionarios como otro insumo para incorporar mejoras al proceso para la siguiente edición del curso. Estos cuestionarios van evolucionando y mejorando también en las sucesivas ediciones del curso. Los cuestionarios finales de los estudiantes deben entregarse al terminar el curso como parte de la evaluación del mismo, el Cliente tiene un tiempo más después de la finalización del curso para realizar el cuestionario, ya que se le pide que realice pruebas del producto implantado para responderlo.

Cuestionarios finales a los estudiantes

Los cuestionarios finales a los estudiantes se realizan al finalizar el curso, como forma de evaluar la aplicación del proceso utilizado y el proyecto realizado, desde la visión de quienes lo llevaron adelante. El diseño inicial de este cuestionario fue realizado en [ADBP00] y fue modificandose en las sucesivas ediciones del curso. El cuestionario se divide en ocho secciones donde se incluyen preguntas sobre los distintos del proceso en aplicación y proyecto en desarrollo, donde algunas respuestas están tabuladas en escala del 1 al 5 tanto cuantitativa como cualitativa, y otras respuestas son del tipo Si/No.

Las secciones definidas son: 1 - Sobre los roles y sus actividades, 2 - Sobre los entregables de su rol, 3 - Sobre el grupo y el proyecto, 4 - Sobre el rol de Director y el mecanismo de comunicación con el proyecto, 5 - Sobre los requerimientos y el cliente, 6 - Sobre el producto obtenido, 7 - Sobre el modelo de proceso propuesto y 8 - Sobre las Auditorías realizadas al grupo, más una última sección 9 - Comentarios, sugerencias, lecciones aprendidas, que no está tabulada y es para incluir cualquier comentario que se quiera realizar.

Cuestionarios finales a los clientes

Los cuestionarios de satisfacción del cliente se envían a los clientes de los proyectos del curso, para que den su visión del producto obtenido luego de realizar pruebas sobre la última versión del producto implantada, y del trabajo del grupo durante el proyecto. Asimismo es de interés

la mejora del trabajo de la Organización con el Cliente en cuanto a interacción y apoyo al rol por ejemplo. Este cuestionario se divide también en distintas secciones agregando varias subsecciones para el tratamiento de distintos aspectos en la misma sección, y las respuestas están tabuladas de la misma forma que el de los estudiantes.

Las secciones definidas son: 1 - Sobre el proceso de generación del producto, 2 - Sobre el producto obtenido, 3 - Otros aspectos de interés y 4 - En general, donde se resume la visión del cliente sobre el proyecto, el grupo y la Organización. Este cuestionario intenta ser de menor extensión que el de los estudiantes, enfocado principalmente en los aspectos más importantes, para que su realización no constituya un esfuerzo adicional para el Cliente.

6.3. Resultados obtenidos

En esta sección se presentan los resultados obtenidos del desarrollo y seguimiento de la prueba de la Extensión SOA realizados. Se brinda principalmente información sobre las mediciones realizadas en cuanto a entregables de la Extensión SOA, contenido de los mismos, y procesamiento de Auditorías y cuestionarios finales a estudiantes y cliente realizados.

6.3.1. Resultados por Fase e iteración

En esta sección se describen los resultados obtenidos por Fase e iteración en cuanto a versiones de los entregables de la Extensión SOA realizados y entregados, más los principales entregables relacionados del proceso base, cantidad de horas totales realizadas por área y rol destacando las correspondientes a la Extensión SOA, y cantidad de Casos de Uso del Negocio, Casos de Uso del Sistema, Casos de Uso relevantes a la Arquitectura identificados y servicios definidos.

En la Fase Inicial se entregó una única versión del documento de Evaluación de la Organización Objetivo en el cual se identificaron los aspectos principales de la Organización en que se realizaría el desarrollo. Se entregaron también dos versiones del Modelo de Casos de Uso del Negocio con la identificación de los procesos del negocio realizada y la especificación de los flujos correspondientes. A partir de este Modelo se realizaron y entregaron también dos versiones del Modelo de Casos de Uso del Sistema, y una versión de la Descripción de la Arquitectura de Software con la definición inicial de la Arquitectura. Del documento del Alcance se entregaron dos versiones, ninguna de las cuales constituyó el Alcance definitivo que fue acordado en la fase siguiente.

En la Fase de Elaboración no se entregaron nuevas versiones de los entregables correspondientes al modelado del Negocio, si se entregaron dos nuevas versiones del Modelo de Casos de Uso del Sistema, de la Descripción de la Arquitectura y una nueva versión del Alcance del Sistema que sería la definitiva. Se entregó la primera versión del Modelo de Servicios, cuya versión final sería entregada recién en la última fase del proyecto.

En la Fase de Construcción se entregó la última versión del Modelo de Casos de Uso del Sistema y de la Descripción de la Arquitectura, y dos versiones del Modelo de Diseño correspondiente. En esta Fase el equipo estaba abocado a la implementación del sistema completo y la verificación de las versiones obtenidas del mismo. En la Fase de Transición se entregó la última versión del Modelo de Servicios. En la tabla 6.3 se muestra el resumen de las entregas realizadas para los principales entregables, discriminando por Fase e iteración.

Entregable	Cantidad versiones	Fase entrega	Iteración	Corresponde
Evaluación Organización Objetivo	1	Inicial	1	Extensión SOA
Modelo de Casos de Uso del Negocio	2	Inicial	1 y 2	Extensión SOA
Modelo de Casos de Uso del Sistema	4	Inicial Elaboración Construcción	1 y 2 2 2	Proceso base
Alcance del Sistema	3	Inicial Elaboración	1 y 2 2	Proceso base
Descripción de la Arquitectura de Software	4	Inicial Elaboración Construcción	2 1 y 2 1	Proceso base
Modelo de Servicios	2	Elaboración Transición	2 1	Extensión SOA
Modelo de Diseño	2	Construcción	1 y 2	Proceso base

Cuadro 6.3: Cantidad versiones entregables principales en cada Fase e iteración

Las horas de dedicación totales del grupo 5, si bien fue el grupo que reportó más horas, no se alejan demasiado de las horas de dedicación de otros grupos del mismo curso que no realizaban la Extensión SOA sino solamente el proceso base propuesto, como por ejemplo el grupo 2 que reportó solamente 38 horas menos. En la tabla 6.4 se muestran datos correspondientes a los grupos del curso 2005 incluyendo cantidad de integrantes, horas totales y líneas de código (LOCs) totales del proyecto, Cliente del proyecto y modelo de proceso seguido por cada grupo.

Año 2005					
Grupo	Integrantes	Horas	LOCs	Cliente	Modelo
1	12	--	--	Gris	OO Java
2	11	4310	41705	Plan Obras	OO Java
3	11	3524	--	Plan Obras	OO Java
4	11	3029	28845	Link-all	Extensión SOA
5	11	4348	38401	Link-all	Extensión SOA
6	12	3472	9374	Link-all	OO Java
7	12	3735	18171	Link-all	OO Java
8	12	4212	8722	UTE	OO .Net

Cuadro 6.4: Cantidad de horas grupos curso 2005 Extensión SOA y Proceso base

Las horas totales desglosadas por Fase, iteración y Disciplina muestran los picos de esfuerzo realizados en cada Fase según el énfasis correspondiente a cada área, por ejemplo en la Fase Inicial se puede ver el esfuerzo en Requerimientos y Diseño, donde las horas de Requerimientos incluyen las realizadas también en el Modelado del Negocio que no fueron desglosadas, en la Fase de Elaboración se puede ver el esfuerzo en Diseño e Implementación, y en la Fase de Construcción el esfuerzo en Implementación y Verificación. En la tabla 6.5 se muestra el desglose de horas.

Area	Inicial	Elaboracion	Construccion	Transicion	Totales
Requerimientos	310	42,5	8,5	0	361
Diseño	99	139,5	28,5	5	272
Implementación	138	720,5	625	35	1518,5
Verificación	44,5	131	174,5	26	376
Gestión de Calidad	85,5	76	74	5	240,5
Gestión de Configuración	55,5	34	6,5	5	101
Gestión del Proyecto	273,5	219,5	128	15	636
Implantación	0	11,5	50,5	4	66
Comunicación	30	31	26	4	91
Estudio	184	48	14	0	246
Otros	134,5	154	151,5	0	440
Totales	1354,5	1607,5	1287	99	4348

Cuadro 6.5: Horas totales desglosadas por Fase, iteración y Disciplina en la Extensión SOA

Las horas totales desglosadas por Disciplina muestran el esfuerzo realizado principalmente en el área de Implementación con 2178,5 horas que representan el 38 % del total, seguido en las áreas de desarrollo por el esfuerzo de Verificación con 576,5 horas que representan el 10 % del total, Requerimientos que incluye el Modelado del Negocio con 369,5 horas representando un 6 % del total y Diseño con 305,5 horas equivalentes al 5 % de las horas totales.

El esfuerzo en implementación se explica principalmente por el uso de nuevas tecnologías que ya se ha mencionado sumado a las dificultades en el diseño siguiendo el enfoque de servicios y el atraso en la documentación del mismo y por lo tanto del pasaje a implementación del diseño realizado. Se vió luego con el grupo que muchas horas de implementación se debían también a no haber realizado una correcta división de las tareas en las primeras etapas, lo que trajo duplicación de horas por estar varios implementadores en la misma asignación de tareas.

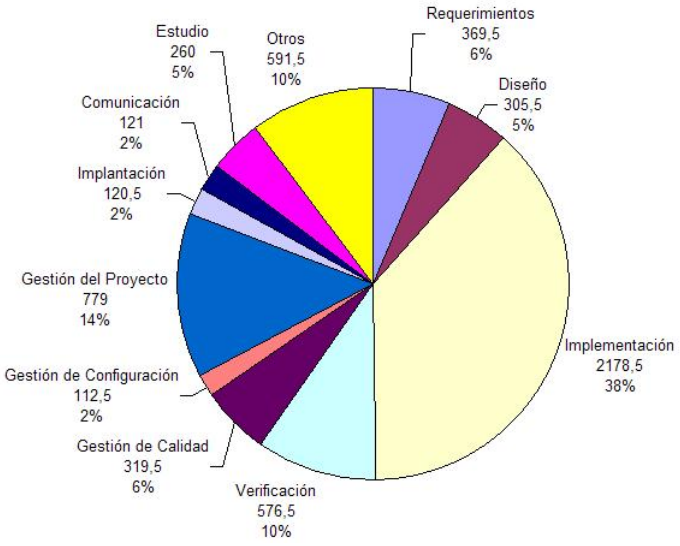


Figura 6.1: Horas totales desglosadas por Disciplina en la Extensión SOA

En la metodología propuesta se debían definir en primer lugar los Casos de Uso del Negocio correspondientes a los procesos del Negocio identificados mediante las reuniones con el Cliente. Luego, a partir de estos procesos y de las reuniones con el Cliente se debían derivar los Casos de Uso del Sistema que en general serán de menor granularidad que los Casos de Uso del Negocio,

por lo que es esperable tener menos Casos de Uso del Negocio que Casos de Uso del Sistema, y que cada uno del Negocio se corresponda con varios del Sistema para realizar dicho proceso.

Luego a partir de los Casos de Uso del Negocio y del Sistema se realiza la definición de la Arquitectura partiendo de los Casos de Uso del Sistema relevantes seleccionados, para definir los servicios y subsistemas asociados que permitirán realizar los Casos de Uso del Sistema y componiendolos, se realizarán los procesos del Negocio identificados. En la tabla 6.6 se muestra la cantidad de elementos registrados en los entregables en cada Fase e iteración.

Fase	Inicial		Elaboración		Construcción		Transición	Totales
Elemento	Iter. 1	Iter. 2	Iter. 1	Iter. 2	Iter. 1	Iter. 2	Iter. 1	--
Casos de Uso del Negocio	5	6	--	--	--	--	--	6
Casos de Uso del Sistema	13	27	--	36	--	38	--	38
Servicios Nuevos	--	--	--	8	--	--	17	17
Servicios Existentes	--	--	--	2	--	--	2	2

Cuadro 6.6: Cantidad de elementos registrados en los entregables en cada Fase e iteración

6.3.2. Resultados por Disciplina, actividad y entregable

En esta sección se presentan los resultados obtenidos en las Disciplinas Modelado del Negocio, Diseño e Implementación en las cuales la Extensión SOA agrega las principales actividades y entregables para el desarrollo orientado a servicios. Se presenta también la Disciplina de Requerimientos ya que como se mencionó al presentar la metodología SOA en el Capítulo 5, la definición de Casos de Uso del Sistema se realiza también teniendo en cuenta los Casos de Uso del Negocio definidos en la Disciplina Modelado del Negocio agregada por la Extensión SOA.

Disciplina Modelado del Negocio

En la Disciplina Modelado del Negocio se realizaron las dos actividades previstas por la metodología: MN1- Evaluar la Organización Objetivo y MN2 - Identificar los procesos del Negocio, obteniendo los entregables correspondientes, cuyas plantillas se adjuntan en el capítulo 10 - Anexo B.

En el entregable Evaluación de la Organización Objetivo se identificaron aspectos del proyecto Link-all y de los sistemas y plataforma SOA existentes que sirvieron de contexto al desarrollo. En el entregable Modelo de Casos de Uso del Negocio se identificaron seis procesos principales luego de dos versiones del entregable, los que se describieron como Casos de Uso del Negocio identificando Actores que participan, flujo principal y flujos alternativos.

Los Casos de Uso del Negocio fueron descritos en forma textual y en forma gráfica mediante Diagramas de Actividad que luego sirvieron para realizar la definición del flujo correspondiente en la herramienta JBPM [JBPM], el motor de definición, ejecución y monitoreo de procesos utilizado. En la figura 6.2se muestra el diagrama de Casos de Uso del Negocio correspondiente.

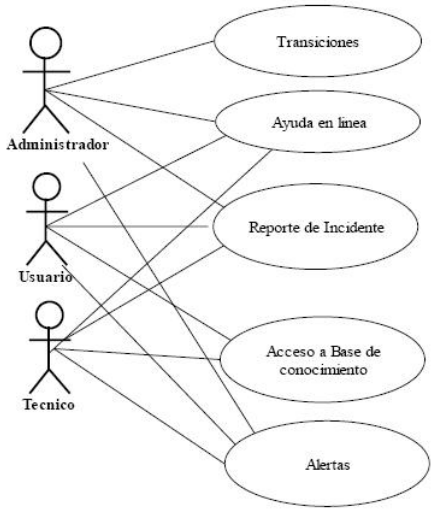


Figura 6.2: Diagrama de Casos de Uso del Negocio en la prueba de la Extensión SOA

Si bien los Casos de Uso del Negocio identificados deberían corresponder todos a procesos del Negocio, se observa que en algunos casos éstos no presentan la granularidad correspondiente a un proceso del Negocio completo, por ejemplo Alertas y Transiciones, sino más bien a un paso en algún otro proceso macro, lo que sería equivalente a un Caso de Uso del Sistema en todo caso. Esta es una de las dificultades encontradas en el seguimiento del enfoque, la identificación de los procesos con la granularidad correcta, que permita especificar los procesos del Negocio en forma completa, para luego traducirlos en Casos de Uso del Sistema de menor granularidad que los realicen mediante pasos sucesivos en la ejecución del Sistema.

De los Casos de Uso del Negocio definidos el que está asociado al principal proceso de la Organización es el de Reporte de Incidente, el cual se compone de los pasos: registro de incidentes, asignación a técnico, resolución de incidentes asignados, ingreso a la Base de Conocimiento del problema resuelto y aviso de resolución al usuario para su aprobación. En la tabla 6.7 se presenta como ejemplo la descripción para el Caso de Uso del Negocio Reporte de Incidente.

Caso de Uso del Negocio	Drescripción
Reporte de Incidente	Un usuario reporta un Incidente, el cual será resuelto por los técnicos, los que luego reportarán la solución del mismo al usuario.

Cuadro 6.7: Ejemplo descripción del Caso de Uso del Negocio Reporte de Incidente en la prueba de la Extensión SOA

Disciplina de Requerimientos

Si bien la Disciplina de Requerimientos y las actividades y entregables definidas corresponden al proceso base, la realización del Modelo de Casos de Uso del Sistema es parte importante de la metodología de la Extensión SOA, ya que a partir de los Casos de Uso del Negocio identificados más los requerimientos especificados se derivan los Casos de Uso del Sistema que luego se utilizan como entrada para la Definición de la Arquitectura y también de los servicios asociados.

La cantidad de Casos de Uso del Sistema definidos es de treinta y ocho por lo que no se mostrarán en su totalidad, sino que se presentará un subconjunto de los mismos seleccionado aplicando el criterio de la importancia definida para los mismos: los Casos de Uso identificados como relevantes a la Arquitectura que son cinco y los Casos de Uso que se derivan del Caso de Uso del Negocio Reporte de Incidente que son seis, de los cuales cuatro también están identificados

como relevantes, lo que hace un total de siete Casos de Uso del Sistema a presentar. En la figura 6.3 se muestra el Diagrama de Casos de Uso del Sistema relevantes a la Arquitectura.

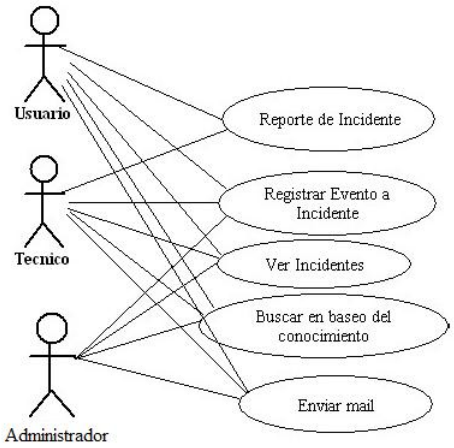


Figura 6.3: Casos de Uso del Sistema relevantes a la Arquitectura en la prueba de la Extensión SOA

El Caso de Uso del Negocio Reporte de Incidente se desglosa en varios Casos de Uso del Sistema para realizar el proceso del Negocio correspondiente: Reportar Incidente, Asignar Técnico a Incidente, Buscar en Base de Conocimiento, Registrar evento a Incidente, Enviar E-mail, Ingresar datos en Base de Conocimiento. Se puede observar que el nombre del Caso de Uso del Sistema Reportar Incidente es demasiado similar al del Caso de Uso del Negocio correspondiente, podría ser más adecuado nombrarlo como Ingresar Incidente o Registrar Incidente, ya que solo realiza la primer parte del proceso del Negocio general.

El flujo principal del Caso de Uso del Negocio Reporte de Incidente comienza cuando un usuario reporta un incidente al Sistema para el cual se asigna un técnico si lo hay disponible, el técnico entonces busca en la base de conocimiento por alguna solución al incidente reportado si la hay. A medida que se trabaja en el incidente reportado se registran los eventos asociados, tales como inicio del trabajo en el incidente, fin de trabajo en el incidente, solución hallada para el mismo. Al cerrar el incidente se envía un e-mail al usuario informando que el incidente fue solucionado para que acepte o no el cierre del mismo. Si el cierre es aceptado el técnico ingresa en la Base de Conocimiento la solución hallada si corresponde.

En la tabla 6.8 se muestra la descripción para los Casos de Uso del Sistema relevantes a la Arquitectura y/o asociados al Caso de Uso del Negocio Reporte de Incidente.

Caso de Uso del Sistema	Descripción	Importancia
Reportar Incidente	El usuario reporta un incidente al sistema, asociandole descripción y descripción corta del problema, categoría, atributos adicionales (si corresponde) y archivo. Se le asigna un número de ticket al mismo, la prioridad y un técnico para su resolución.	Relevante a la Arquitectura – parte del CU Negocio Reporte Incidente
Asignar Técnico a Incidente	En este caso de uso se maneja la posibilidad de que el Administrador elija uno de los Técnicos que se encuentran trabajando para asignarle un incidente que se encuentra en el estado NoAsignado.	parte del CU Negocio Reporte Incidente
Registrar evento a incidente	Registra en el correr del tiempo, los distintos eventos dentro de un incidente que se encuentra en el Sistema. Se registra, quién ingresó el evento, fecha de inicio, fecha fin, descripción, tipo y tarea (si corresponde). Esta funcionalidad es utilizada tanto por los Técnicos, como por los Usuarios y el Administrador.	Relevante a la Arquitectura – parte del CU Negocio Reporte Incidente
Buscar en Base de Conocimiento	Se realizan búsquedas en la Base de Conocimiento a fin de encontrar problemas similares antes reportados por parte de los Técnicos, Usuarios y él Administrador. La información a mostrar, es filtrada dependiendo de quién sea el que esta realizando la consulta.	Relevante a la Arquitectura – parte del CU Negocio Reporte Incidente
Enviar un e-mail	En este caso de uso se maneja el envío de mensajes vía e-mail, siendo los destinatarios alguno(s) de los loggin del Sistema. Esto se da tanto en el momento en que se dispara una alerta en el Sistema que da lugar a tal caso de uso, o cuando el Usuario llama a la funcionalidad.	Relevante a la Arquitectura – parte del CU Negocio Reporte Incidente
Ingresar datos en Base de Conocimiento	El Técnico luego de resolver un incidente, ingresa en la Base de Conocimiento datos relacionados al problema planteado por el usuario, como ser la solución que encontró al mismo, descripción del problema, descripción corta, archivo adjunto, ranking, fecha de ingreso y categoría.	parte del CU Negocio Reporte Incidente
Ver incidentes	Se podrán ver los atributos de un incidente ingresado en el Sistema. Para el mismo se mostrarán, número de ticket, categoría, descripción corta, descripción del problema, prioridad, atributos adicionales, Técnico responsable (si corresponde) y estado en el cual se encuentra.	Relevante a la Arquitectura

Cuadro 6.8: Descripción de los principales Casos de Uso del Sistema en la prueba de la Extensión SOA

Disciplina de Diseño

En la Disciplina de Diseño se realizaron las actividades definidas para el diseño orientado a servicios: D6 - Identificar y categorizar servicios, D7 - Especificar servicios, D8 - Investigar servicios existentes, D9 - Asignar servicios a componentes y D10 - Definir orquestación de servicios. Las tres primeras actividades se realizan conjuntamente con la actividad de Definición de la Arquitectura del proceso base.

En dicha actividad, a partir de los Casos de Uso del Sistema identificados como relevantes a la Arquitectura que se presentan en la vista de Casos de Uso del Documento de la Arquitectura (SAD), se realiza la descomposición del software en subsistemas en la vista Lógica, más la identificación de procesos en la vista de Procesos, el mapeo a la implementación de dichos elementos de software en la vista de Implementación y el despliegue del software en el hardware en la vista de Deployment.

La metodología SOA agrega el Modelo de servicios donde se incluye la información correspondiente a los servicios identificados, categorizados y especificados, del cual se extrae la vista de Servicios que se incluye en el SAD, donde se muestra la identificación de servicios realizada, para luego presentar también la asignación de servicios a componentes en la vista de Implementación, y su despliegue en la vista de Deployment. Los servicios fueron identificados y categorizados

siguiendo la guía de clasificación propuesta, en servicios centrados en procesos para la definición de servicios que orquestan otros servicios y corresponden a procesos del Negocio, servicios intermediarios que proveen funcionalidades más complejas utilizando servicios básicos del negocio y de acceso a datos, los que a su vez utilizan servicios básicos de infraestructura provistos por la plataforma Link-all.

Se definieron dos aplicaciones independientes basadas en servicios, una para el sistema Help-Desk y otra para la Gestión de la Base de Conocimiento, que utilizan algunos servicios definidos comunes a ambas. Esto permite que se puedan ejecutar en forma independiente, e incluso integrar con otras aplicaciones desarrolladas en la plataforma Link-all, ya que cada una solamente necesita tener disponibles los servicios definidos para si misma y los comunes. Por ejemplo, se podría hacer uso de la Gestión de la Base de Conocimiento desde otra aplicación sin tener relación con la aplicación definida para el Help-Desk.

Se definió también un servicio básico del negocio para proveer la funcionalidad de directorio de servicios, de forma de desacoplar la ubicación de los servicios de la utilización de los mismos, que es usado por ambas aplicaciones y podría ser reutilizado por otras en la plataforma SOA del Link-all también. En la tabla 6.4 se presenta una selección de la identificación y categorización de servicios realizada para la aplicación Help-Desk, la Gestión de la Base de Conocimientos define servicios similares.

Servicio	Categorización	Propósito
GestionBPM	Centrado en Procesos	encapsula la gestion del proceso de resolución de incidentes
ServiceFacade	Intermediario	Provee acceso y funcionalidades agregadas a partir de servicios básicos
GestionIncidentes	Básico del Negocio	encapsular todas las funcionalidades relacionadas a los incidentes
GestionTecnicos	Básico del Negocio	Manejar la información relativa a los técnicos registrados en el sistema
GestionSeguridadHD	Básico del Negocio	proporciona los mecanismos de seguridad que necesita el helpdesk
GestionSesionHD	Básico del Negocio	encapsula las funcionalidades relacionadas a la sesión que esta utilizando el HelpDesk
Directorio de Servicios	Básico del Negocio	brinda ubicación de los servicios en forma desacoplada
DBMaintenanceHD	Básico de Acceso a Datos	brinda acceso a la base de datos HelpDesk

Figura 6.4: Ejemplo de Identificación y categorización de servicios en la prueba de la Extensión SOA

Siguiendo la plantilla brindada en la Extensión SOA para el Modelo de Servicios, según se adjunta en el Capítulo 10 - Anexo B, se debe indicar para cada servicio: en la Sección 1 - Identificación y Categorización de servicios, el nombre del servicio, las funcionalidades provistas, su categorización, los Caso/s de Uso asociado/s y subsistema de Diseño asociado; en la Sección 2 - Especificación de servicios, para cada servicio su contrato funcional indicando métodos, parámetros de entrada y salida, pre y post condiciones, excepciones; en la Sección 3 - Servicios existentes, el nombre del servicio, las funcionalidades provistas, categorización y contrato funcional.

En la tabla 6.9 se muestra a modo de ejemplo la información correspondiente a la Sección 1 - Identificación y Categorización de servicios del Modelo de Servicios para el servicio Gestión de Incidentes, que es parte del proceso del Negocio Reporte de Incidente.

Servicio	Funcionalidades provistas	Categorización	Caso/s de Uso asociado/s	Subsistema de Diseño asociado
Gestión de Incidentes	Reportar un incidente Registrar evento a incidente Asignar técnico a incidente Asociar workflow a incidente Obtener proceso de incidente Obtener incidente de proceso Ver incidente Listar incidentes Listar incidentes de usuario Listar incidentes de login responsable Listar categorias Listar eventos de incidente para administrador Listar eventos de incidente para técnico Listar eventos de incidente para usuario	Servicio básico del Negocio	Reportar incidente Registrar evento a incidente Ver incidente Asignar técnico a incidente Listar categorias Listar incidentes Listar eventos de un incidente Reasignar tarea de un incidente	GestionIncidentes

Cuadro 6.9: Especificación del servicio Gestión de Incidentes en la prueba de la Extensión SOA

En la tabla 6.10 se muestra el contrato funcional definido para el método ReportarIncidente del servicio Gestión Incidente, como ejemplo de la información correspondiente a la Sección 2 - Especificación de servicios del Modelo de Servicios.

Método	ReportarIncidente		
Parámetros	Incidente	IncidenteVO	Datos del incidente
Valor de retorno	Ticket	Long	Ticket asignado por el sistema al incidente
Excepciones	DBException		
Descripción	Se ingresa en el sistema el incidente con los datos en IncienteVO El sistema le asigna un número de ticket		
Precondiciones	N/A		
Postcondiciones	Se registra un incidente en el sistema con los datos de IncidenteVO. El número de ticket asociado al incidente es el valor de retorno.		

Cuadro 6.10: Contrato funcional del método ReportarIncidente en la Especificación del servicio Gestión de Incidentes en la prueba de la Extensión SOA

Disciplina de Implementación

En el Modelo de Implementación se indica la correspondencia entre las definiciones realizadas en el Diseño y la Implementación definida a partir de estas. En particular la asignación de servicios a componentes realizada en la actividad D9, definiendo los packages de implementación y los ejecutables a generar para realizar el deployment de la aplicación. También se indica la correspondencia con subsistemas de diseño definidos por ejemplo para el diálogo de la presentación, lo que constituye el Application frontend definido para la aplicación.

6.3.3. Resultados de las Auditorías al proceso

Las Auditorías al proceso fueron realizadas al finalizar las Fases Inicial y de Elaboración, y los resultados obtenidos en las mismas concordaban con la visión del grupo que tenía el Director del proyecto en el seguimiento semanal, por lo que en general sirvieron para confirmar dicha visión. A continuación se presenta un resumen del resultado de cada una de las Auditorías realizadas.

En la Fase Inicial los principales problemas expresados por los integrantes del proyecto en la Auditoría fueron la dificultad para entender SOA, dificultad para aplicar el enfoque iterativo incremental en la ejecución de las actividades y realización de entregables definidos en las disciplinas, inexperiencia en las tecnologías que se debían utilizar por lo que se debían invertir más horas de las que se creía en solucionar los aspectos técnicos del proyecto, la dificultad para estimar correctamente y para seguir la planificación realizada, que en general no se cumplía por contratiempos como los mencionados y dificultades de coordinación y comunicación que el grupo manifestó a lo largo del proyecto. El Arquitecto centralizó la realización de los documentos del diseño por lo que el pasaje a implementación se vió atrasado realizandose de manera informal en las primeras reuniones con los implementadores.

En la Fase de Elaboración la mayor preocupación expresada por el grupo en la Auditoría era la cantidad de horas que estaban dedicando al proyecto sin poder recuperar el atraso de la Fase Inicial, las estimaciones y planificación no se veían mejoradas, sumandose en esta fase el incremento en horas por integración de la implementación y preparación del ambiente de testeo y posterior verificación de los ejecutables. Los casos de prueba los hicieron a partir de los Casos de Uso del sistema, y realizaron verificación unitaria, de integración y del sistema. También tuvieron problemas con la gestión de configuración especificamente con el CVS que debían usar para hacerla, lo que redundó en las horas de más realizadas en las integraciones. Se pusieron al día con la documentación del diseño lo que permitió estabilizar la Arquitectura y concentrarse en la implementación del Ejecutable de la Línea Base de la Arquitectura, pero finalmente se atrasaron una semana más en lograr dicho ejecutable.

6.3.4. Resultados de los cuestionarios finales

En este capítulo se presentan los resultados obtenidos a partir del procesamiento de los cuestionarios finales realizados a los estudiantes y a los clientes, sobre la percepción del proyecto realizado y el proceso seguido. Este procesamiento se realizó siguiendo las definiciones dadas en [PCFE02], primer documento del procesamiento de cuestionarios finales realizado como parte de las tareas en el [GRIS06]. Se presentan las respuestas a las preguntas que están más relacionadas con la aplicación de la Extensión SOA, no presentando aquellas preguntas que tienen mayor relación con el proceso base o con el curso.

Como se mencionó en la presentación de los cuestionarios en 6.2.4 se utilizan dos tipos de escalas, con valores del 1 al 5 y con opción SI/NO, en el caso de la escala del 1 al 5 además la valoración puede ser cuantitativa o cualitativa, con el siguiente significado: en el caso de valoración cualitativa el significado de la escala es 1 - Malo, 2 - Regular, 3 - Aceptable, 4 - Bueno, 5 - Muy bueno; en el caso de valoración cuantitativa el significado de la escala es 1 - Nada, 2 - Poco, 3 - Medio, 4 - Bastante, 5 - Mucho.

Si las preguntas no son contestadas el valor se toma como N/C y en preguntas que dependen de respuestas a preguntas anteriores se puede utilizar el valor N/A si no aplica. Asimismo hay algunas preguntas en el cuestionario al Cliente cuyas respuestas están tabuladas en valores de porcentaje, por ejemplo menos 20 % o más 75 %, para indicar probabilidad de ocurrencia del aspecto sobre el cual se está preguntando.

Cuestionarios finales a los estudiantes

A continuación se presenta el procesamiento realizado de los cuestionarios finales a los estudiantes mostrando en forma de tablas los valores de las respuestas obtenidas en las secciones más importantes para las principales preguntas. En el caso de preguntas en la escala del 1 al 5, en las primeras columnas de las tablas se presenta la cantidad de respuestas dadas a cada valor en la escala, y en las dos últimas columnas se presenta el promedio de las respuestas a cada pregunta y el total de estudiantes que la respondieron, siendo el máximo 11 que es el total de intregrantes del grupo 5 al cual corresponde la prueba realizada. En el caso de preguntas en la escala SI/NO, en las primeras columnas se presenta la cantidad de respuestas dadas a cada

valor, y en la última columna el total de estudiantes que respondieron la pregunta, ya que no es posible presentar un promedio en este tipo de preguntas.

Como se mencionó en la presentación de los cuestionarios en 6.2.4, las secciones definidas para los cuestionarios finales de los estudiantes son: 1 - Sobre los roles y sus actividades, 2 - Sobre los entregables de su rol, 3 - Sobre el grupo y el proyecto, 4 - Sobre el rol de Director y el mecanismo de comunicación con el proyecto, 5 - Sobre los requerimientos y el cliente, 6 - Sobre el producto obtenido, 7 - Sobre el modelo de proceso propuesto y 8 - Sobre las Auditorías realizadas al grupo, más una última sección 9 - Comentarios, sugerencias, lecciones aprendidas, que no está tabulada y es para incluir cualquier comentario que se quiera realizar. A continuación se presenta el procesamiento realizado sobre las respuestas de los cuestionarios finales de los estudiantes, para cada sección definida.

Sección 1 - Sobre los roles y sus actividades

Esta sección contiene preguntas relacionadas con la propuesta de actividades específicas para cada rol, su descripción en el proceso seguido, la comprensión y realización de las mismas en el proyecto por los roles asociados. En la tabla 6.11 se presenta el procesamiento de respuestas para las preguntas de la Sección 1 en la escala del 1 al 5.

SECCION 1 - Sobre los roles y sus actividades	RESPUESTA					
PREGUNTA	2	3	4	5	Media	Total
¿Entendió las descripciones de las Actividades propuestas para su rol?		5	5	1	3,64	11
¿Las Actividades propuestas para su rol estaban bien definidas?	1	5	4	1	3,45	11
¿Leyó las descripciones de las Actividades propuestas para su rol?		1	2	8	4,64	11
¿Realizó las Actividades propuestas para su rol?	1		8	2	4,00	11
¿Se apegó a las descripciones de las Actividades propuestas para su rol?		4	6	1	3,73	11
Total	2	15	25	13	3,89	55

Cuadro 6.11: Respuestas Sección 1 - Sobre los roles y sus actividades - escala 1 al 5

Como se puede observar en la tabla 6.11 la mayoría de las respuestas a las preguntas de la Sección 1 se encuentran en los valores mayores o iguales que 3, lo que significa que la comprensión, proposición, descripción, realización y apego a la descripción de las actividades correspondientes a los distintos roles, fue Media, Bastante y Mucha, además el promedio en general para cada pregunta está sobre el valor 3,5 indicando una buena valoración sobre el tema de las actividades definidas para los roles.

Sección 2 - Sobre los entregables de su rol

Esta sección contiene preguntas sobre la definición de entregables y realización de los mismos para los distintos roles. En la tabla 6.12 se presenta el procesamiento de respuestas para las preguntas de la Sección 2 en la escala del 1 al 5.

SECCION 2 – Sobre los entregables de su rol	RESPUESTA				
PREGUNTA	2	3	4	Media	Total
¿Considera que los entregables definidos por semana para su rol fueron adecuados?	1	4	6	3,45	11
¿Logró realizar los entregables de su rol en los plazos previstos?	2	3	6	3,36	11
Total	3	7	12	3,41	22

Cuadro 6.12: Respuestas Sección 2 - Sobre los entregables de su rol - escala 1 al 5

Como se puede observar en la tabla 6.12 la mayoría de las respuestas a las preguntas de la Sección 2 se encuentran en el valor 4 que significa que la definición de entregables en semanas

y su realización en los plazos previstos por parte de los roles fue con adecuación Bastante, indicando una buena valoración sobre el tema de entregables del rol; sin embargo el promedio está cercano a 3,4 por las respuestas de valores 2 y 3 dadas, que en cada pregunta suman casi la mitad de las respuestas. En esta sección también hay preguntas en la escala SI/NO cuyas respuestas se presentan en la tabla 6.13 a continuación:

SECCION 2 – Sobre los entregables de su rol	RESPUESTA		
PREGUNTA	N	S	Total
¿Considera que ciertos entregables debieran ser opcionales o prescindibles?	6	5	11
¿Considera que el contenido definido en las plantillas de los entregables fue adecuado?	1	10	11
Total general	7	15	22

Cuadro 6.13: Respuestas Sección 2 - Sobre los entregables de su rol - escala SI/NO

En la tabla 6.13 se puede observar que la apreciación sobre que algunos entregables sean opcionales o prescindibles está dividida casi a la mitad, con 6 respuestas por NO y 5 respuestas por SI, sin embargo, en los comentarios sobre cuales serían estos entregables, se mencionan principalmente los correspondientes al proceso base y no a la Extensión SOA. Sobre las plantillas brindadas para los entregables la mayoría de las respuestas, 10 en 11, corresponden a SI, lo que indica una buena valoración del material entregado como parte de la realización de los entregables del rol, incluyendo las plantillas brindadas por la Extensión SOA.

Sección 3 - Sobre el grupo y el proyecto

Esta sección muestra aspectos internos del trabajo del grupo, en cuanto a relacionamiento interpersonal y niveles de comunicación y coordinación en el desarrollo del proyecto. Otras preguntas en esta sección no tienen respuestas tabuladas, brindando la posibilidad de realizar comentarios al respecto del funcionamiento del grupo, problemas detectados y problemas no detectados a tiempo que repercutieron en el proyecto, así como una valoración personal del proyecto realizado. En la tabla se presenta el procesamiento de respuestas para la Sección 3 en la escala del 1 al 5.

SECCION 3 – Sobre el grupo y el proyecto	RESPUESTA						
PREGUNTA	2	3	4	5	N/C	Media	Total
¿Cómo fue el relacionamiento entre los integrantes de su grupo?		3	5	2	1	3,90	11
El nivel de comunicación y coordinación en su grupo fue:	3	8				2,73	11
Total general	3	11	5	2	1	3,29	22

Cuadro 6.14: Respuestas Sección 3 - Sobre el grupo y el proyecto - escala 1 al 5

Como se puede observar en la tabla 6.14 el relacionamiento del grupo tiene respuestas en valores igual o mayores a 3, lo que significa que se aprecia entre Regular y Muy bueno según los distintos integrantes. Esto puede deberse a que el grupo al inicio del proyecto tuvo problemas de relacionamiento debido a que los estudiantes no se conocían, lo que finalmente fue superado por el grupo.

Sin embargo, en la pregunta sobre los niveles de comunicación y coordinación en el grupo las respuestas están en valores igual o menores que 3, lo que significa que fue Regular y Aceptable según la mayoría de los integrantes del grupo. Esta dificultad para coordinar el trabajo interno repercutió en el desarrollo del proyecto sobre todo en los atrasos registrados en las extensiones de las Fases Inicial y de Elaboración como se mostró en la tabla 6.2 de la Sección 6.2.2.

Sección 4 - Sobre el rol del Director de proyecto

Esta sección evalúa la adecuación del seguimiento realizado al grupo y la utilidad para la realización de las actividades del rol. Como se comentó previamente, en este caso el rol del Director de proyecto lo desempeño el docente que propone la Extensión SOA seguida por el grupo, por lo que también se realizaron actividades de apoyo en la metodología explicando las actividades y entregables agregados al proceso base. En la tabla 6.15 se presenta el resumen de respuestas para la Sección 4 en la escala del 1 al 5.

SECCION 4 – Sobre el rol del Director de proyecto	RESPUESTA					
PREGUNTA	2	3	4	5	Media	Total
¿El seguimiento hecho por el docente al grupo fue de utilidad para las actividades de su rol?	2	2	3	4	3,82	11
Cree que el seguimiento hecho por el docente al grupo fue:	1		4	6	4,36	11
Total general	3	2	7	10	4,09	22

Cuadro 6.15: Respuestas Sección 4 - Sobre el rol del Director de proyecto - escala 1 al 5

Como se puede observar en la tabla 6.15 la mayoría de las respuestas están en valores mayores o igual a 4 cuyo significado es Bueno en la escala planteada, siendo el promedio de las mismas también cercano al valor 4, por lo que se aprecia que la valoración del seguimiento y apoyo brindado por el Director de proyecto al grupo es de Buena y Muy buena. Cabe destacar que a los monitoreos solo concurrían en forma obligatoria los roles de Arquitecto, Administrador, y Responsables de Calidad y Verificación, por lo que las respuestas en valores de 2 y 3 en general tienen como comentario el hecho de que al no concurrir a los monitoreos y tener contacto con el Director de proyecto solo al finalizar las Fases, el seguimiento realizado no incidía directamente en dichos roles.

Sección 5 - Sobre los requerimientos y el Cliente

Esta sección contiene preguntas relacionadas con las actividades que plantea el proceso que involucran la participación del Cliente en el desarrollo del proyecto, incluyendo semanas marcadas para la validación de entregables y entregables a validar en cada etapa, así como la comunicación del Cliente con el grupo. En la tabla 6.16 se presenta el resumen de las respuestas correspondientes a la Sección 5 en la escala 1 al 5.

SECCION 5 – Sobre los requerimientos y el cliente	RESPUESTA				
PREGUNTA	3	4	5	Media	Total
¿Cómo califica en general el grado de involucramiento y participación del Cliente?		8	3	4,27	11
¿Le parece que las semanas marcadas para la presentación y validación con el Cliente fueron adecuadas?	4	6	1	3,73	11
¿Le parece que los entregables que se debían validar fueron adecuados?	4	5	2	3,82	11
Considera que la comunicación Cliente - grupo de proyecto fue :	2	6	3	4,09	11
Total general	10	25	9	3,98	44

Cuadro 6.16: Respuestas Sección 5 - Sobre los requerimientos y el Cliente - escala 1 al 5

Se puede observar en la tabla 6.16 que las respuestas a las distintas preguntas están todas en valores igual o mayores a 3, siendo la mayoría en cada pregunta el valor 4, lo que significa una valoración de Bueno y Bastante en general, según aplica la escala cualitativa o cuantitativa. Sobre las preguntas que tienen mayor relación con el proceso planteado, las semanas marcadas para las validaciones con el cliente parecen adecuadas teniendo una mayoría de respuestas en el valor 4 con significado Bastante y promedio de 3,73, y la adecuación de los entregables a validar en cada caso también tiene una mayoría de respuestas en el valor 4 con significado Bastante y promedio de 3,82, por lo que la valoración en general de las propuestas del proceso es Buena. Esto incluye los entregables a validar de la Extensión SOA como el Modelo de Casos de Uso

del Negocio y el Modelo de Servicios, que se agregan a las validaciones previstas por el proceso base.

Sección 6 - Sobre el producto obtenido

La sección 6 - Sobre el producto obtenido tiene preguntas sobre la valoración del grupo del producto desarrollado, incluyendo el cumplimiento del alcance acordado con el Cliente y estado del producto para su utilización, y evaluación de atributos de calidad asociados al mismo, que pueden o no haber sido explicitados como parte de los requerimientos del Cliente para el producto. En la tabla 6.17 se presenta el procesamiento de respuestas para la Sección 6 en la escala del 1 al 5.

SECCION 6 – Sobre el producto obtenido	RESPUESTA							
PREGUNTA	**1**	**2**	**3**	**4**	**5**	**N/C**	**Media**	**Total**
¿Considera que su producto está listo como para poder ser utilizado?		1	5	4	1		3,45	11
¿Cree que el producto entregado cumplió con el alcance del proyecto acordado con el Cliente?			3	7	1		3,82	11
Nivel alcanzado por su producto en cuanto a:								
Amigabilidad				7	4		4,36	11
Confiabilidad		1	6	4			3,27	11
Correctitud			5	6			3,55	11
Funcionalidad			6	4	1		3,55	11
Interoperabilidad	1		3	1	6		4,00	11
Performance	1	2	5	2		1	2,80	11
Portabilidad		1	3		6	1	4,10	11
Robustez		5	4	2			2,73	11
Verificabilidad			4	7			3,64	11
Total general	2	9	36	33	17	2	3,57	121

Cuadro 6.17: Respuestas Sección 6 - Sobre el producto obtenido - escala 1 al 5

Como se puede observar en la tabla 6.17 sobre el estado del producto para ser utilizado la mayoría de las respuestas están en los valores 3 y 4, con significado Medio y Bastante, incluyendo en los comentarios observaciones sobre errores remanentes que hacen que algunas partes del producto estén aún inestables. Sobre el cumplimiento del alcance acordado con el Cliente, la mayoría de las respuestas están en el valor 4 con significado Bastante, por lo que la valoración es la de haber entregado el producto pactado.

Sobre los atributos de calidad evaluados para el producto, se puede ver que los mejor valorados son Amigabilidad, Correctitud, Funcionalidad y Verificabilidad que presentan valores igual o mayores a 3, significando nivel Aceptable, Bueno y Muy bueno, e Interoperabilidad y Portabilidad que si bien presentan valores en 1 y 2, con significado Malo y Regular, también los presentan en 5 con significado Muy bueno, por lo que tienen un promedio de 4, mejor que el promedio de los anteriores que ronda en 3,5. Los atributos peor valorados son Performance y Robustez, que presentan valores en toda la escala con mayoría en valores bajos, dando como resultado promedios menores a 3 que es Aceptable. Esto se debe según los comentarios, al uso de tecnologías y a la inestabilidad en algunas partes del producto mencionada anteriormente.

Sección 7 - Sobre el Modelo de proceso propuesto

Esta sección plantea preguntas sobre el grado de apego al proceso en el desarrollo del proyecto, esfuerzo de entendimiento para cada rol, y niveles del proceso en cuanto a complejidad, utilidad de entregables y grado de realización de actividades propuestas y no propuestas, y actividades no realizadas. En la tabla 6.18 se presenta el resumen de respuestas para la Sección 7 en la escala del 1 al 5.

SECCION 7 – Sobre el modelo de proceso propuesto	RESPUESTA						
PREGUNTA	1	2	3	4	5	Media	Total
¿En cuánto se apegó su grupo al Modelo de Proceso propuesto?			4	6	1	3,73	11
Al estudiar el Modelo de Proceso, ¿qué esfuerzo le implicó entender el trabajo que correspondía a su rol?		2	3	4	2	3,55	11
Califique el Modelo de Proceso propuesto en cuanto a:							
Complejidad			5	6		3,55	11
grado de utilidad de los entregables propuestos			6	5		3,45	11
proporción de actividades no propuestas que hicieron	6	2	3			1,73	11
proporción de actividades propuestas que no se hicieron	1	5	3	2		2,55	11
proporción de actividades propuestas que se hicieron			3	8		3,73	11
Total general	7	9	27	31	3	3,18	77

Cuadro 6.18: Respuestas Sección 7 - Sobre el Modelo de proceso propuesto - escala 1 al 5

Como se puede ver en la tabla 6.18 la mayoría de las respuestas están en valores de 3 y 4, con significado Medio y Bastante, el apego al proceso tiene promedio de 3,73 indicando un seguimiento de Bastante al proceso planteado, incluyendo la Extensión SOA en la valoración, el esfuerzo para entender el proceso tiene promedio de 3,55 indicando también una valoración entre Medio y Bastante, así como la complejidad del proceso que tiene el mismo promedio, por lo que la valoración sugiere un grado de complejidad y esfuerzo de comprensión importante para su correcta utilización.

El grado de utilidad de los entregables se evalúa también en ese rango, con un promedio de 3,45, y la proporción de actividades realizadas tiene un promedio de 3,73 indicando un alto grado de realización de las mismas. En cuanto a actividades propuestas que no se hicieron y actividades no propuestas que se hicieron, los valores son menores a 3 y 2 significando Medio y Poco, lo que resulta una buena valoración en cuanto a actividades que no se hicieron y actividades faltantes.

La Sección 7 también tiene varias preguntas cuya escala de respuestas es SI/NO, relacionadas con cambios a realizar al proceso, duración de Fases e iteraciones definidas y sus objetivos, cumplimiento de actividades en las semanas previstas, y utilidad y seguimiento del proceso. En la tabla se presenta el procesamiento de respuestas correspondientes a la Sección 7 en escala SI/NO.

SECCION 7 – Sobre el modelo de proceso propuesto	RESPUESTA		
PREGUNTA	N	S	Total
¿Cambiaría, quitaría o agregaría algo con respecto al Modelo de Proceso propuesto?	5	6	11
¿Considera que la cantidad de iteraciones planteadas en cada fase, así como su duración y objetivos fueron adecuadas para el proyecto?	1	10	11
¿Cree que el Modelo de Proceso propuesto fue una guía útil para el desarrollo del proyecto?		11	11
¿Pudieron cumplir con las actividades en las semanas en las que estaban planificadas?	6	5	11
¿Su grupo siguió el Modelo de Proceso propuesto?		11	11
Total general	16	50	66

Cuadro 6.19: Respuestas Sección 7 - Sobre el Modelo de proceso propuesto - escala SI/NO

Como se puede observar en la tabla 6.19 las respuestas unánimes del grupo corresponden a haber seguido el proceso planteado y la utilidad del mismo para el desarrollo del proyecto, que tiene las 11 respuestas en el valor SI, lo que indica una buena valoración del grupo sobre el proceso seguido, incluyendo la Extensión SOA. En cuanto a la adecuación de la duración de Fases e iteraciones y objetivos planteados, la amplia mayoría contestó también que SI y solamente una

respuesta correspondió a NO, lo que también indica una buena valoración de la dimensión del tiempo prevista en el proceso.

En cuanto al cumplimiento de actividades según lo previsto y cambios a realizar al proceso las respuestas están divididas casi a la mitad, lo que indica distintas percepciones, debidas principalmente a la visión de los roles, donde en algunos casos se menciona sobrecarga de actividades y entregables en algunos momentos del proceso, y los cambios principalmente en cuanto a definición de roles y eliminación de entregables asociados.

Otras secciones del cuestionario

La Sección 8 - Sobre las Auditorías realizadas al grupo no se presenta en este análisis ya que dichas valoraciones corresponden sobre todo al curso, y la Sección 9 - Comentarios, sugerencias, lecciones aprendidas tampoco se presenta debido a que los comentarios refieren sobre todo también a aspectos relativos a la organización del curso.

Cuestionarios finales al Cliente

En esta sección se presenta el procesamiento del cuestionario final realizado al Cliente, mostrando principalmente las preguntas más importantes en relación al proceso propuesto y la aplicación del mismo en el curso 2005. En las tablas se presentan la sección y subsección que se está procesando, pero a diferencia de las tablas presentadas para el procesamiento de los cuestionarios finales de los estudiantes, las tablas en esta sección tienen solamente una respuesta en cada pregunta, la del Cliente del proyecto, por lo que se muestra directamente el valor asignado a cada una en la escala correspondiente, en vez de la cantidad de respuestas en cada valor. En las preguntas con respuestas tabuladas en valores de porcentaje, se muestra directamente el valor asignado por el cliente, por ejemplo menos 20 % o más 75 %.

Sección 1 - Sobre el proceso de generación del producto

Esta sección presenta varias subsecciones con distintas preguntas sobre el proceso de generación del producto, en cuanto al relevamiento de requerimientos realizado por el grupo, Fases e iteraciones y productos intermedios generados, y validación de los mismos.

SubSección 1.1 - En relación a la metodología utilizada para identificación de requerimientos

En esta subsección las preguntas refieren al trabajo en requerimientos planteado por el proceso y realizado por el grupo, en cuanto a frecuencia y duración de las reuniones, documentos tratados en las mismas, agenda prevista, comprensión y sugerencias realizadas por parte del grupo, incluyendo las actividades agregadas por la Disciplina Modelado del Negocio en la Extensión SOA, las que se realizan también a partir de las reuniones con el cliente, principalmente durante la Fase Inicial. En la tabla 6.20 se muestran las respuestas del Cliente a dichas preguntas.

SECCION 1 – Sobre el proceso de generación del producto			
SubSección 1.1. En relacion a la metodologia utilizada para identificacion de requerimientos:	**RESPUESTA**		
PREGUNTA	**4**	**5**	**Total**
¿Considera que los integrantes del grupo comprendian correctamente los requerimientos?		5	5
¿Esta de acuerdo con la frecuencia de las reuniones?	4		4
¿Los documentos tratados en las reuniones le parecieron utiles para lograr los objetivos?		5	5
¿Los integrantes del grupo realizaron sugerencias y/o aportes en estas reuniones?		5	5
¿Se plantearon adecuadamente los temas de la agenda de forma de lograr reuniones efectivas?	4		4

Cuadro 6.20: Respuestas Sección 1 - SubSección 1.1 - En relación a la metodología utilizada para la identificación de requerimientos- escala 1 al 5

Como se puede observar en la tabla 6.20 la mayoría de las respuestas están en valores de 4 y 5 con significado Bastante y Mucho, por lo que la propuesta de actividades en el proceso y

trabajo realizado en el relevamiento de requerimientos por el grupo se valora positivamente. Es de destacar que la utilidad de la documentación prevista por el proceso para tratar en dichas reuniones, Modelo de Casos de Uso del Negocio y del Sistema, documento de requerimientos, está evaluada con valor 5 de significado Mucho, por lo que dicha propuesta se valora como adecuada.

SubSección 1.4 - En relación a las Fases/iteraciones y productos intermedios manejados en el proyecto

En esta subsección se evalúa la adecuación de los productos intermedios obtenidos durante el proyecto, en cantidad y contenido, según la propuesta del proceso, así como su utilidad para evaluar el avance del proyecto y la adhesión a los requerimientos planteados, y la utilidad de las reuniones al finalizar cada Fase/iteración. En la tabla 6.21 se presentan las respuestas a las preguntas planteadas.

SECCION 1 – Sobre el proceso de generación del producto			
SubSección 1.4 - En relación a las fases/iteraciones y productos intermedios manejados en el proyecto	**RESPUESTA**		
PREGUNTA	**4**	**5**	**Total**
¿Esta de acuerdo con la cantidad de productos intermedios obtenidos durante la ejecucion del proyecto?		5	5
¿Esta de acuerdo con los productos intermedios obtenidos y su contenido?		5	5
¿Le parecio adecuado que existieran interacciones con Ud. al final de las fases/iteraciones seleccionadas?	4		4
¿Le resultaron los productos de utilidad para evaluar el avance del proyecto y su adhesion a los requerimientos planteados?		5	5

Cuadro 6.21: Sección 1 - Subsección 1.4 - En relación a las Fases/iteraciones y productos intermedios manejados en el proyecto - escala 1 al 5

Como se puede observar en la tabla 6.21 tres de las cuatro preguntas planteadas tienen respuestas con valor 5 con significado Mucho, por lo que la adecuación de los productos intermedios obtenidos tanto en cantidad como contenido y su utilidad para evaluar el avance del proyecto y la adhesión a los requerimientos planteados se valora como muy adecuada a los objetivos planteados. En cuanto a las reuniones con el Cliente propuestas al finalizar las Fases/iteraciones se evalúan con valor 4 con significado Bastante por lo que la valoración resulta también positiva.

SubSección 1.5 - En relación a la metodología utilizada para la validación de los productos en cada Fase/iteración

Esta subsección presenta preguntas asociadas a la validación de productos intermedios que propone el proceso base, incluyendo aquellos que agrega la Extensión SOA como el Modelo de Casos de Uso del Negocio y el Modelo de Servicios. Al finalizar cada Fase/iteración se proponen reuniones con el Cliente, en las que se presenta al mismo el avance del proyecto y un resumen de aspectos más importantes como ser principales procesos del Negocio y requerimientos identificados, Casos de Uso relevantes a la Arquitectura y Arquitectura de la solución, entre otros. Se plantean preguntas también sobre el rol del Cliente en estas validaciones, en cuanto a validación formal de documentos, omisión de entregables en las validaciones por parte del grupo o rechazo de los mismos por parte del Cliente.

SECCION 1 – Sobre el proceso de generación del producto				
SubSección 1.5 - En relacion a la metodologia utilizada para la validacion de los productos de cada fase/iteracion	RESPUESTA			
PREGUNTA	1	2	4	Total
¿Le resultaron efectivas las presentaciones realizadas al fin de cada fase/iteracion?			4	4
¿Valido y/o observo Ud. formalmente los documentos y/o entregables presentados?		2		2
¿Rechazo Ud. entregables por algun motivo?	1			1
¿Se tuvieron en cuenta sus observaciones?			4	4
¿En alguna de las entregas se omitio entregar productos intermedios acordados?	1			1

Cuadro 6.22: Sección 1 - Subsección 1.5 - En relación a la metodología utilizada para la validación de los productos de cada Fase/iteración - escala 1 al 5

Como se puede observar en la tabla 6.22 las presentaciones al Cliente realizadas al finalizar cada Fase/iteración resultaron efectivas, evaluandose con valor 4 de significado Bastante, así como la incorporación de las observaciones realizadas por el Cliente en las mismas. Sobre omisiones de productos intermedios, rechazo de entregables y validaciones formales de los mismos por parte del Cliente, la valoración está en valores de 1 y 2 con significado Nada y Poco, por lo que la propuesta y desarrollo de las validaciones en general se evalúa positivamente.

Sección 2 - Sobre el producto obtenido

Esta sección contiene varias subsecciones con distintas preguntas sobre el producto obtenido, en cuanto a requerimientos funcionales y no funcionales implementados y valoración de los mismos, adecuación a las restricciones de diseño planteadas, y adecuación de la documentación asociada.

SubSección 2.1 - En relación a los requerimientos funcionales y no funcionales

En esta subsección se presentan los porcentajes asignados al cumplimiento de requerimientos funcionales y no funcionales del producto entregado según las expectivas del cliente, así como la identificación de requirimientos y modelado del Negocio en las primeras semanas del proyecto, en las reuniones del grupo con el cliente.

SECCION 2 – Sobre el producto obtenido	RESPUESTA
SubSección 2.1 - En relacion a los requerimientos funcionales y no funcionales	Porcentaje
¿Que porcentaje de los requerimientos funcionales fueron identificados dentro de las primeras cuatro semanas del proyecto?	menos 75%
¿Que porcentaje de los requerimientos no funcionales fueron identificados dentro de las primeras cuatro semanas del proyecto?	menos 90%
¿Que porcentaje de los requerimientos funcionales fueron implementados a satisfaccion del cliente a la finalizacion del proyecto?	mas 90%
¿Que porcentaje de los requerimientos implementados no logro cubrir sus expectativas?	menos 20%

Cuadro 6.23: Sección 2 - Subsección 2.1 - En relación a los requerimientos funcionales y no funcionales - porcentajes

Como se puede observar en la tabla 6.23 los requerimientos funcionales fueron implementados a satisfacción del cliente en más del 90 %, y de los requerimientos implementados en general solamente menos del 20 % no cubrieron las expectativas del cliente, por lo que se evalúa positivamente el cubrimiento de expectativas del cliente en cuanto al producto entregado. Sobre la identificación de requerimientos funcionales y no funcionales en las primeras semanas del proyecto, se observa que la mayoría pudieron ser identificados, evaluandose como positivo el relevamiento de requerimientos y modelado del Negocio que se realiza también a partir de las reuniones con el cliente durante la Fase Inicial.

SubSección 2.2 - Respecto al cumplimiento de los requerimientos no funcionales planteados y los atributos de calidad del software

Las subsecciones 2.2 y 2.3 se presentan en forma conjunta ya que están relacionadas, presentando preguntas sobre el cumplimiento de requerimientos y atributos de calidad para el software, así como de las restricciones de diseño planteadas por el Cliente. En la tabla 6.24 se presenta la evaluación del producto realizada por el Cliente.

SECCION 2 – Sobre el producto obtenido						
SubSección 2.2 - Respecto al cumplimiento de los requerimientos no funcionales planteados y los atributos de calidad del software	**RESPUESTA**					
PREGUNTA	**1**	**2**	**3**	**4**	**5**	**Total**
Confiabilidad				4		4
Eficiencia				4		4
Funcionalidad				4		4
Mantenibilidad					5	5
Portabilidad	1					1
Usabilidad					5	5

Cuadro 6.24: Sección 2 - Subsección 2.2 - Respecto al cumplimiento de los requerimientos no funcionales planteados y los atributos de calidad del software - escala 1 al 5

Como se observa en la tabla 6.24 en la subsección 2.2 la evaluación de atributos de calidad para el producto obtenido está mayoritariamente en valores de 4 y 5 con significado Bueno y Muy bueno, menos la portabilidad que está evaluada con valor 1 con significado Malo.

Si se compara con la evaluación de los atributos realizada por los integrantes del grupo en el cuestionario final a los estudiantes, se puede observar que la confiabilidad tenía un promedio de 3,27 mientras el cliente la evalúa en 4, la eficiencia evaluada como performance tenía un promedio de 2,80 mientras el cliente la evalúa en 4, la funcionalidad tenía un promedio de 3,55 mientras el cliente la evalúa en 4 también, la mantenibilidad y usabilidad no estaban evaluadas en el cuestionario de los estudiantes pero el cliente las evalúa como 5 ambas, y la portabilidad es la que tiene percepción más disimil, mientras para el grupo tenía un promedio de 4,10 el cliente la evalúa en 1, por lo que en cuanto a dicho atributo hay gran diferencia en la evaluación.

Subsección 2.3 - Evaluación de otros aspectos del producto

En esta subsección se presenta una pregunta sobre la adecuación a las restricciones de diseño impuestas por el cliente, cuya evaluación es 5 con significado Mucho, aclarando en los comentarios que el grupo había seguido la metodología de desarrollo con enfoque SOA propuesta en la Extensión SOA para que la aplicación tuviera un diseño orientado a servicios, y se habían utilizado todos los servicios de la plataforma Link-all necesarios, interactuando con ésta en forma adecuada. En la tabla 6.25 se presenta la evaluación realizada.

SECCION 2 – Sobre el producto obtenido		
SubSección 2.3 – Evaluación de otros aspectos del producto	**RESPUESTA**	
PREGUNTA	**5**	**Total**
Califique la adecuacion del producto a las restricciones de diseño impuestas por el cliente	5	5

Cuadro 6.25: Sección 2 - Subsección 2.3 - Evaluación de otros aspectos del producto - escala 1 al 5

Se evalúa entonces en general que el producto tiene un buen cumplimiento de requerimientos y atributos de calidad planteados en forma explícita o implícita, y su diseño se corresponde con lo solicitado por el Cliente, cumpliendo con las restricciones impuestas para el mismo.

SECCION 2 – Sobre el producto obtenido		
SubSección 2.5 - Respecto a la documentacion del producto entregada	RESPUESTA	
PREGUNTA	5	Total
Califique la completitud de la documentacion tecnica entregada	5	5
Califique la completitud y contenidos de la documentacion de usuario entregada. ¿Es correcta y comprensible?	5	5
Califique los contenidos de la documentacion tecnica entregada. ¿Esta actualizada y consistente con la version del producto entregada?	5	5

Cuadro 6.26: Sección 2 - Subsección 2.5 - Respecto a la documentación del producto entregada - escala 1 al 5

SubSección 2.5 - Respecto a la documentación del producto entregada

En esta subsección las preguntas corresponden a la documentación entregada con el producto, evaluando aspectos como completitud y contenido en documentación técnica y de usuario, así como la consistencia entre la documentación entregada respecto a la versión del producto liberada. En la tabla 6.26 se presentan las respuestas correspondientes.

Como se observa en la tabla 6.26 todas las respuestas a las preguntas en esta subsección tienen valor 5 con significado Mucho, por lo que la valoración de toda la documentación entregada se encuentra en el valor máximo de la escala, evaluando como muy positivo este aspecto en el producto entregado, el cual agrega un importante valor para el Cliente.

Sección 3 - Otros aspectos de interés

Esta sección contiene subsecciones con preguntas sobre varios aspectos del producto obtenido, de las que se presenta la subsección 3.1 correspondiente a la probabilidad de implantación del producto en el ambiente del Cliente.

SubSección 3.1 - Asigne una probabilidad de implantación del producto entregado

En esta subsección se muestran los porcentajes asignados al cliente para la posible implantación del producto entregado, en el estado en que fue entregado, con cambios menores o mayores. En la tabla 6.27 se muestran las respuestas del cliente sobre este aspecto.

SECCION 3 – Otros aspectos de interés	
SubSección 3.1 - Asigne una probabilidad de implantacion del producto entregado	RESPUESTA
PREGUNTA	Porcentaje
Asigne una probabilidad de implantacion del producto entregado	
Sin cambios	menos 20%
Con cambios menores	menos 50%
Con cambios mayores	menos 75%

Cuadro 6.27: Sección 3 - Subsección 3.1 - Asigne una probabilidad de implantación del producto entregado - porcentajes

Como se puede observar en la tabla, una posible implantación del producto estaría requiriendo la realización de cambios mayores, aunque el producto obtenido se evalúa como adecuado en otras subsecciones, debe recordarse que se trataba de un prototipo del cual el cliente no tenía expectativas de implantación. Sin embargo, por información posterior brindada por el cliente, el producto fue instalado en la plataforma LInk-all y utilizado para hacer varias demostraciones a los socios internacionales sobre las posibilidades del sistema Help-Desk.

Sección 4 - En general

En esta sección se presenta en general la evaluación de la satisfacción del Cliente con producto recibido y con su participación como Cliente en el curso, así como la evaluación global otorgada al grupo de proyecto.

SECCION 4 – En general		
SubSección 4.2 - De una calificacion global para el grupo	RESPUESTA	
PREGUNTA	12	Total
teniendo en cuenta el producto obtenido y el desarrollo del proyecto por parte del grupo segun su vision de cliente del mismo:	12	12

Cuadro 6.29: Sección 4 - Subsección 4.2 - Calificación global para el grupo - escala 1 al 12

SubSección 4.1 - Sobre el grado de satisfacción del cliente

La subsección 4.1 califica el nivel general de la satisfacción del cliente respecto al producto recibido y su participación como cliente en el curso, en la tabla 6.28 se presentan las respuestas correspondientes.

SECCION 4 – En general		
Subsección 4.1 - Sobre el grado de satisfaccion del cliente respecto al producto recibido	RESPUESTA	
PREGUNTA	5	Total
Califique su nivel de satisfaccion	5	5
¿Volveria a ser cliente de la asignatura Proyecto de Ingeniería de Software?	5	5

Cuadro 6.28: Sección 4 - Subsección 4.1 - Sobre el grado de satisfacción del cliente - escala 1 al 5

Como se puede apreciar en la tabla 6.28 todas las preguntas en esta subsección tienen asignado el valor máximo, en cuanto al nivel de satisfacción sobre el producto y participación como cliente en el curso, el valor asignado es 5 con significado Mucho, siendo el máximo de la escala por lo que se evalúa como muy positivo este aspecto.

Subsección 4.2 - Calificación global para el grupo

En esta subsección se presenta la calificación global asignada al grupo por el Cliente, la cual es tenida en cuenta como uno de los factores en la evaluación final del grupo que se realiza en el contexto del curso. Otros factores que se consideran es la evaluación del seguimiento del proceso por parte del grupo según la visión del Director de Proyecto y las Auditorías realizadas, y la presentación del trabajo realizado por el grupo mediante presentaciones del producto obtenido y del proceso seguido. La escala para la calificación del grupo es en todos los casos del 1 al 12. En la tabla 6.29 se presenta la nota asignada.

Como se puede observar en la tabla la calificación otorgada al grupo en la escala del 1 al 12 es 12, con significado Excelente, en la evaluación global del trabajo realizado. Por lo tanto, se evalúa como muy positivo el desarrollo global del proyecto en cuanto a los aspectos relacionados con el rol del Cliente y la obtención del producto requerido por el mismo.

6.4. Conclusiones de la aplicación de la metodología SOA

El Grupo 5 siguió la metodología SOA propuesta y definió una Arquitectura basada en servicios, repositorio de servicios en un directorio, y orquestación de servicios en JBPM. Sobre los servicios básicos e intermediarios se construyen servicios centrados en procesos que resuelven la lógica de negocio del Help-Desk, empleando el motor de workflow y BPM, JBPM del servidor de aplicaciones JBoss, extendido y configurado para el sistema. Se definieron, implementaron y entregaron dos aplicaciones autónomas, una para el Help-Desk y otra para la Base de Conocimiento, definidas en base a servicios, que se integraron en la plataforma SOA del Link-all utilizando servicios básicos de infraestructura que la misma provee a sus aplicaciones.

Una de las dificultades encontradas en la aplicación de la metodología SOA fue la identificación de los procesos del Negocio con la granularidad correcta, que permita especificar los procesos del Negocio en forma completa, para luego traducirlos en Casos de Uso del Sistema de menor granularidad que los realicen mediante pasos sucesivos en la ejecución del Sistema. Pese al soporte realizado sobre el enfoque SOA y los conceptos del mismo, la diferencia entre Casos de Uso del Negocio y Casos de Uso del Sistema no se vió reflejada en la documentación del modelado entregada por el grupo, y en algunos casos la definición de Casos de Uso del Negocio no fue correcta.

Sin embargo, la mayor dificultad se presentó en la Disciplina de Diseño, donde en el transcurso del seguimiento de la aplicación de la metodología SOA se detectó que no resultaba fácil iterar en la definición de subsistemas de la Arquitectura y la definición de servicios adecuados a estos. Si bien la dificultad en la definición de la granularidad correcta para los servicios identificados es reconocida en la literatura que trata con el enfoque SOA, se cree que en esta experiencia parte de la misma provenía de la falta de experiencia del equipo en diseño en general y dificultades en la conceptualidad del enfoque SOA, como se mencionó, a pesar del soporte realizado sobre estos.

Otros problemas en el transcurso del proyecto estuvieron dados por razones que son tangenciales a la metodología SOA propuesta, como ser el atraso en la culminación de la Fase Inicial que se debió principalmente a dificultades con las tecnologías que se debían utilizar, la mayoría de las cuales les eran desconocidas a los integrantes del equipo, dificultades de coordinación y comunicación en el equipo dado que los integrantes del mismo en general no se conocían entre sí desde antes, por lo que les llevó un tiempo acomodar y ajustar el trabajo en equipo con los distintos roles definidos y personalidades, y también con el estudio del modelo de proceso en general y de la metodología SOA en particular, que tuvieron que conceptualizar para poder aplicar el enfoque SOA al desarrollo.

Sin embargo, al finalizar el proyecto, la evaluación que hacen los estudiantes de la metodología SOA seguida para el desarrollo es positiva, como se presentó en 6.3.4, valorando la definición y realización de actividades, entregables y roles, las plantillas asociadas a los entregables, las validaciones de productos intermedios definidas con el cliente y el producto final Help-Desk obtenido. Sobre aspectos relativos al proceso entero, se valora en general como complejo de comprender, y que plantea demasiados entregables de los cuales algunos podrían ser opcionales. Se debe tener en cuenta en esta valoración sin embargo, que algunos entregables se definen en el contexto del curso como parte de las actividades de control de trabajo del grupo, en una asignatura de la carrera, y que si podrían ser opcionales en otro caso de aplicación.

En los cuestionarios realizados a los clientes, como se presentó en 6.3.4, se observa también una valoración positiva de los aspectos del proceso que incluían interacción con el cliente, como ser las reuniones para recabar información sobre procesos del Negocio y requerimientos, y las validaciones de productos intermedios. Sobre el producto obtenido se puede destacar el cumplimiento a satisfacción del cliente de más del 90 % de los requerimientos funcionales planteados, y buen cumplimiento en general de los requerimientos de atributos de calidad definidos, cumpliendo con el alcance pactado para el sistema Help-Desk. Finalmente el grado de satisfacción del cliente con el trabajo desarrollado por el grupo queda establecido en la nota global asignada al mismo, que fue de 12 en la escala del 1 al 12.

Capítulo 7

Ajustes, mejoras y generalización de la Metodología SOA

7.1. Introducción

Para su puesta en práctica en el curso "Proyecto de Ingeniería de Software", se definió un conjunto núcleo de Disciplinas, Actividades, Entregables y Roles, para extender el proceso base adaptación del RUP para desarrollo con enfoque SOA. A partir de la puesta en práctica de dicho conjunto núcleo, se pudieron detectar correcciones a realizar en los elementos que integran la Metodología SOA propuesta.

Como se tenía previsto desde el comienzo del trabajo, éste conjunto núcleo sería extendido agregando elementos en otras Disciplinas que se consideran importantes pero se pueden ver como de soporte a las definidas en el núcleo de la metodología, en lo que constituye las mejoras a la metodología propuesta. Por lo que se contará con dos versiones de la metodología SOA que se podrán utilizar: la versión con ajustes que corresponde al núcleo probado y ajustado, y la versión con mejoras que corresponde al núcleo probado y ajustado extendido con nuevas disciplinas y actividades.

Luego, otro aspecto interesante a discutir sobre la metodología propuesta, es si sus elementos pueden generalizarse para ser adoptados por otros procesos de desarrollo base, que no sigan necesariamente el enfoque del RUP. En este sentido se realiza una generalización de la metodología propuesta para ser incluida en el modelo de procesos de COMPETISOFT[COMP06], como Perfil SOA para el desarrollo. También se investigan nuevos enfoques metodológicos surgidos recientemente como parte de la investigación en el enfoque SOA que se realiza en forma continua en la comunidad.

En este sección se discuten y presentan los aspectos mencionados relativos a los ajustes, mejoras y generalización de la Metodología SOA. En la subsección 7.1.1 se presentan los ajustes realizados al conjunto núcleo de elementos definidos, en la subsección 7.1.2 se presentan las mejoras realizadas para extender este conjunto núcleo de elementos, y en la subsección 7.1.3 se presenta la generalización de la metodología SOA propuesta, y en se provee una breve descripción de nuevos enfoques metodológicos surgidos durante la realización de este trabajo de tesis.

7.1.1. Ajustes a la Metodología SOA propuesta

Los ajustes a la metodología SOA propuesta se realizan sobre los elementos núcleo de la misma, a la luz de la prueba realizada en el curso "Proyecto de Ingeniería de Software". En la ejecución de la Extensión SOA definida, se pudieron apreciar algunas inconsistencias en los entregables

de entrada y salida de las actividades definidas. En cuanto a la definición de las Disciplinas, actividades y roles no se encontraron mayores ajustes a realizar, ya que la evaluación de su adecuación para el desarrollo con Enfoque SOA fue ampliamente positiva. A continuación se presentan los ajustes realizados en las Disciplinas Modelado del Negocio y Diseño.

7.1.1.1. Disciplina Modelado del Negocio

En la Disciplina de Modelado del Negocio se realiza un único ajuste en una de las dos actividades que la integran, agregando un entregable de entrada que solo figuraba como salida de la misma. A continuación se muestra el ajuste realizado.

Identificar los procesos del Negocio (MN2)

Se agrega como entrada también el glosario de términos, ya que los términos manejados en las reuniones de requerimientos con el Cliente deben ser utilizados en el modelado de los procesos del Negocio. El glosario figuraba únicamente como salida de la actividad, cuando en realidad es el glosario actualizado el que se genera, si se encuentra necesario modificar las definiciones o agregar términos nuevos que surjan en la realización de esta actividad. En la figura 7.1 se muestra el modelado definitivo para esta actividad.

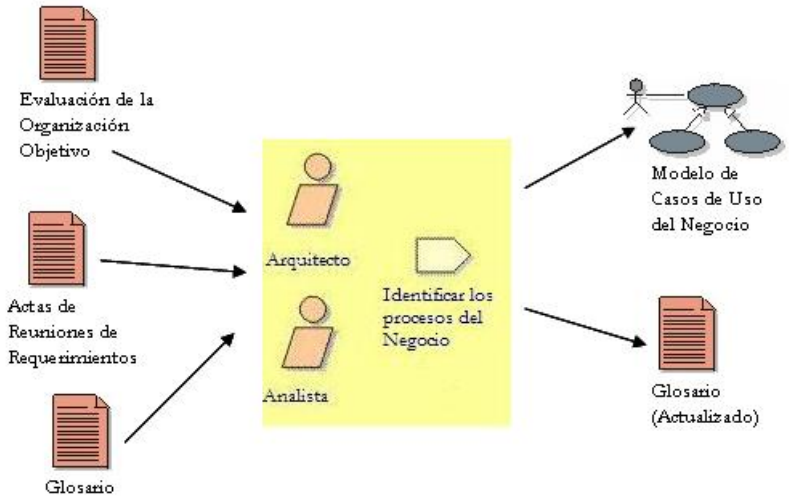


Figura 7.1: Modelado de la actividad Identificar los procesos del Negocio (MN2) en la Extensión SOA ajustada

7.1.1.2. Disciplina Diseño

En la Disciplina Diseño se encontraron inconsistencias en los entregables de entrada y salida como estaban definidos para dos de las cinco actividades definidas. A continuación se muestran los ajustes realizados en las dos actividades.

Identificar y categorizar servicios (D6)

Se agregan como entrada el Modelo de Casos de Uso del Sistema y el Modelo de Servicios. El Modelo de Casos de Uso del Sistema tiene la definición de Casos de Uso del Sistema que corresponden a los Casos de Uso del Negocio. Como en esta actividad se identifican y categorizan todos los servicios, el Modelo de Casos de Uso del Negocio que ya se incluía como entrada, sirve para identificar servicios centrados en procesos que componen otros servicios, pero para la identificación de los servicios de menor granularidad que los compondrán, se deben tener en cuenta los Casos de Uso del Sistema.

El Modelo de Servicios que se menciona como salida con la sección correspondiente a identificación y categorización de servicios informada, también debe ser entrada de la actividad, ya que al ser un procesos iterativo e incremental, al realizar la actividad en forma sucesiva, ya se contará con información de los servicios en dicho Modelo que deberá ser tenida en cuenta. En la figura 7.2 se muestra el modelado de la actividad con los ajustes mencionados.

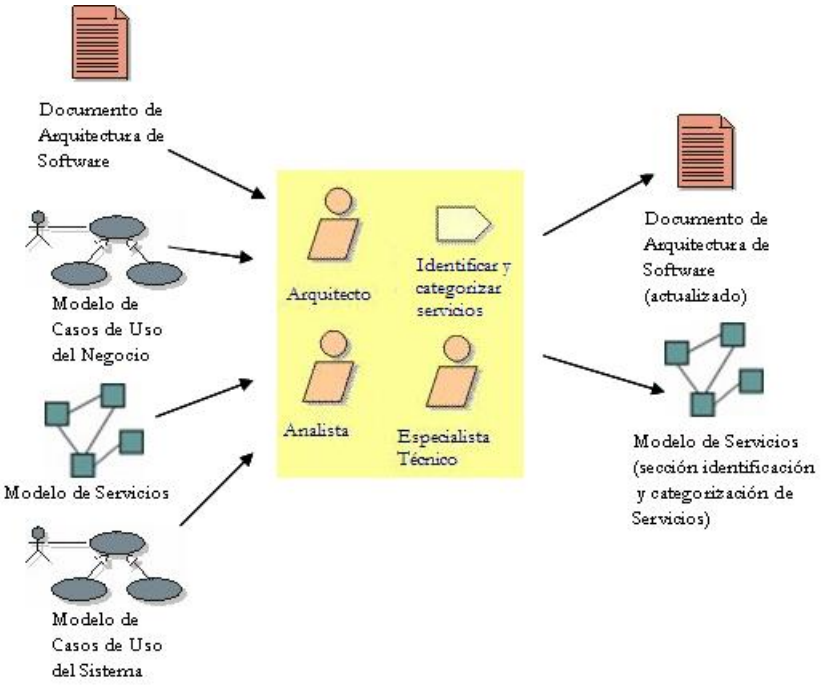


Figura 7.2: Modelado de la actividad Identificar y categorizar servicios (D6) en la Extensión SOA ajustada

Investigar servicios existentes (D8)

Se incluye como entrada el Modelo de Servicios, por la misma razón que se menciona en la actividad anterior, ya que al ser iterativa incremental la ejecución de actividades, en sucesivas iteraciones se debe tener en cuenta también la información que éste modelo contenga. En la figura 7.3 se muestra el modelado de la actividad ajustada.

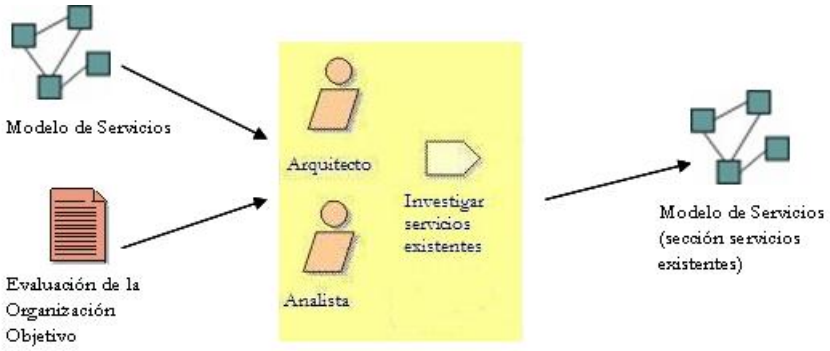


Figura 7.3: Modelado de la actividad Investigar servicios existentes (D8) en la Extensión SOA ajustada

7.1.2. Mejoras a la Metodología SOA propuesta

El objetivo de las mejoras a la metodología SOA propuesta es extender el conjunto de elementos núcleo definido inicialmente, que brinda las bases para realizar un desarrollo con enfoque SOA, principalmente con elementos que sirvan de soporte a los definidos inicialmente. Se intentó agregar por lo menos una actividad específica para servicios en cada una de las Disciplinas que no constituyen el conjunto núcleo definido, haciendo énfasis en el enfoque a servicios desde dicha Disciplina para acompañar el enfoque de la metología SOA propuesta.

Para realizar estas mejoras se tuvieron en cuenta las guías provistas en el enfoque SOA de [KRAF05], tomando como base la definición realizada para los elementos del conjunto núcleo de la metodología. En este enfoque se mencionan principalmente las áreas de Verificación, Gestión del Proyecto y Gestión de la Configuración, por lo que teniendo en cuenta dichas recomendaciones y la prueba de la metodología SOA realizada, se agregan en las Disciplinas correspondientes Actividades, Entregables y Roles para el desarrollo con enfoque SOA. Se agregan también elementos de soporte a la gestión y verificación en la Disciplina de Implementación, que ya estaba definida en la metodología SOA propuesta y en la Disciplina de Gestión de la Calidad para la identificación específica de propiedades de calidad sobre los servicios.

Como las Disciplinas a las que se agregan elementos en esta mejora ya existen en el proceso base, la convención de las extensiones al mismo es que los elementos definidos en la extensión suplantan a los elementos existentes. Esto es, si una actividad definida en una extensión tiene la misma numeración que una actividad existente en el proceso base, la sobreescribe, debiendose ejecutar la actividad de la extensión en lugar de la actividad del proceso base. A continuación se describen estos elementos siguiendo la notación presentada en el Capítulo 5 donde se presenta la metodología SOA propuesta.

7.1.2.1. Disciplina Implementación

El propósito de la disciplina de Implementación en la metodología SOA propuesta ya agregaba los siguientes objetivos a los definidos en el proceso base: implementar los componentes proveedores de servicios según lo establecido por la asignación de servicios a componentes implementar el ligamiento de servicios en las application frontend y en los servicios que llaman a otros servicios, utilizando la estrategia definida en la Organización. En esta mejora se agregan los siguientes: planificar y realizar la integración de componentes asignados a servicios en forma sucesiva en cada iteración, realizando la verificación unitaria de los servicios antes de su liberación para integración y verificación.

En esta Disciplina el único Rol definido es el Implementador que participa en la actividad Implementar Servicios (I13) definida, se agrega en esta mejora el Rol de Responsable de la Integración, que tendrá asociadas las actividades relativas a la integración de componentes asignados a servicios para construir las versiones del sistema en cada iteración, y el rol Coordinador de Desarrollo que tendrá a su cargo la gestión del desarrollo. Los dos roles agregados también estaban definidos en el proceso base.

Para cumplir con los objetivos agregados se modifican las siguientes actividades existentes en el proceso base: Planificar la integración de la iteración (I4), Integrar el sistema (I5) y Realizar la verificación unitaria de servicios (I7). El resto de las actividades existentes en la Disciplina Verificación no se modifican ya que de la forma en que están definidas son adecuadas para el enfoque SOA. A continuación se describen estas actividades y el resto de los elementos definidos, siguiendo la forma de presentación de la metodología SOA propuesta.

Planificar la integración de la iteración (I4)

Objetivo: El objetivo de esta actividad es elaborar el Plan de integración de la iteración. Para esto se deben establecer claramente las fechas en que se realizarán las integraciones de los componentes asignados a los servicios, en forma sucesiva hasta el final de la iteración, así como las responsabilidades sobre los distintos servicios a integrar.

Descripción: Se deben definir que componentes asignados a servicios se integrarán en la iteración, de acuerdo al plan de desarrollo definido para la iteración, ajustando si fuera necesario las estimaciones de tiempos de implementación definidas para incorporar las integraciones de componentes asignados a servicios. Se deben identificar claramente las dependencias entre los componentes asignados a los servicios que se integrarán en esta iteración, para definir la estrategia más adecuada a utilizar. Por ejemplo si es bottom-up, se deben identificar también que otros componentes deben estar disponibles como stubs.

Roles: El rol responsable de esta actividad es el Responsable de integración, que será el implementador que toma ese rol para esa iteración, los implementadores, el Arquitecto que tiene el conocimiento del diseño y la implementación, el Coordinador de desarrollo que es el responsable del plan de desarrollo y el Responsable de Verificación para coordinar las fechas de las integraciones con las fechas de pruebas.

Entregables de entrada: Esta actividad tiene como entregables de entrada el Plan de Desarrollo, la Descripción de la Arquitectura (SAD), el Modelo de Diseño, el Modelo de Servicios, y el Modelo de Implementación, para la identificación de dependencias entre los componentes asignados a los servicios. Adicionalmente se podrán utilizar los Modelos de Casos de Uso del Negocio y de Casos de Uso del Sistema cuando surjan dudas sobre dependencias o definiciones.

Entregables de salida: Esta actividad tiene como entregable de salida el Plan de Integración de la iteración.

Modelado de la actividad: en la figura 7.4 se muestra el modelado de la actividad utilizando los elementos de modelado provistos por el RUP.

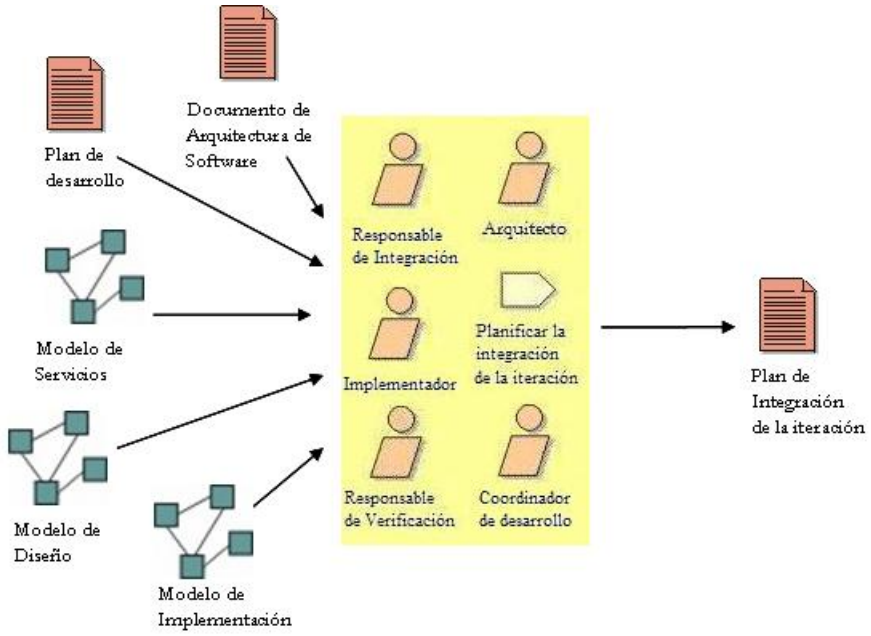


Figura 7.4: Modelado de la actividad Planificar la integración de la iteración (I4) en la Extensión SOA mejorada

Fundamentación: Esta actividad se agrega para dar soporte a la gestión del desarrollo basado en servicios, donde la planificación del desarrollo, integración y verificación de la iteración, se realiza referenciando los componentes asignados a servicios que se obtendrán como resultado de la misma. Es una actividad de soporte ya que lo que agrega es el foco en la unidad de gestión que son los servicios, pero mantiene el concepto definido para la actividad que es realizar la planificación de la Integración de la iteración, para trabajar con un cronograma de desarrollo, integración y verificación definido para la iteración, coordinado entre las distintas áreas.

Integrar el sistema (I5)

Objetivo: El objetivo de esta actividad es realizar la integración de servicios como está previsto en el Plan de Integración de la iteración. Por lo que, hasta llegar a la integración final de la iteración, se ejecuta varias veces para realizar integraciones intermedias, sobre las que se van agregando componentes asignados a servicios en cada integración según está definido en dicho plan.

Descripción: Se debe realizar la integración de los componentes asignados a servicios en cada momento, como está definido en el Plan de Integración de la iteración. Para cada integración que se realiza se debe asegurar la existencia de los elementos necesarios para simular componentes que no están disponibles, esto es stub y drivers. Para obtener el incremento sobre el sistema que corresponde a cada integración, se deben integrar los componentes definidos generando un sistema intermedio, que se tomará para la siguiente integración y se le agregarán los componentes que corresponda para obtener el nuevo incremento. Tanto los sistemas intermedios como el ejecutable final de la iteración son liberados a verificación para verificar la integración realizada.

Roles: El rol responsable de esta actividad es el Responsable de integración, que será el implementador que toma ese rol para esa iteración, los implementadores que tienen asignados componentes clave, que puedan ayudar a resolver los problemas que ocurran en la integración, el Arquitecto por su conocimiento del diseño e implementación y el coordinador de desarrollo por las planificaciones de desarrollo e integración.

Entregables de entrada: Esta actividad tiene como entregables de entrada el Plan de Integración de la iteración, la Descripción de la Arquitectura (SAD), el Modelo de Diseño, el Modelo de Servicios y el Modelo de Implementación, para la identificación de dependencias entre los componentes asignados a los servicios. Adicionalmente se podrán utilizar los Modelos de Casos de Uso del Negocio y Casos de Uso del Sistema cuando surjan dudas sobre las dependencias o las definiciones realizadas.

Entregables de salida: Esta actividad tiene como entregable de salida el sistema integrado con los componentes asignados a servicios definidos para integrar en la iteración.

Modelado de la actividad: en la figura 7.5 se muestra el modelado de la actividad utilizando los elementos de modelado provistos por el RUP.

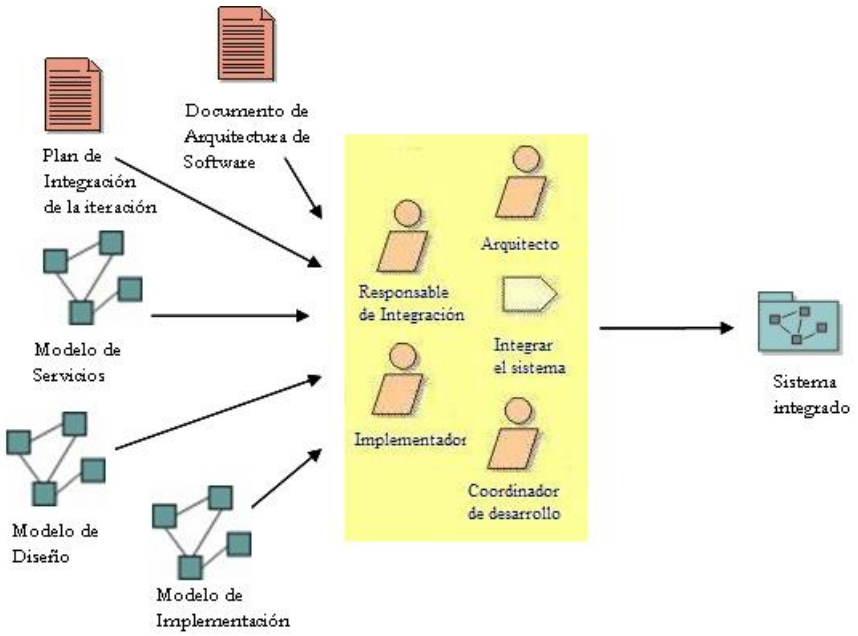


Figura 7.5: Modelado de la actividad Integrar sistema (I5) en la Extensión SOA mejorada

Fundamentación: Esta actividad se agrega para dar soporte a la gestión del desarrollo basado en servicios, de forma de enfocarse en los componentes asignados a servicios que se obtendrán como resultado de la misma. Se debe seguir la planificación realizada en base a servicios para obtener la integración de componentes asignados a servicios correspondiente a cada etapa definida. Es una actividad de soporte ya que lo que agrega es el foco en la unidad de gestión que son los servicios, pero mantiene el concepto definido para la actividad que es realizar la integración definida para la iteración.

Realizar verificación unitaria de servicios (I7)

Objetivo: El objetivo de esta actividad es realizar la verificación unitaria de los componentes asignados a servicios en el ambiente de desarrollo, para poder detectar fallas en los mismos antes de integrarlos con el resto de los servicios.

Descripción: En esta actividad se debe realizar la verificación unitaria de los componentes asignados a servicios que se hayan implementado, en el ambiente de desarrollo. Si existe un caso de prueba unitaria definido en el Plan de pruebas de la iteración se ejecutará el mismo. De lo contrario, según la especificación del servicio se definen las pruebas a realizar sobre el componente que lo implementa, verificando la correctitud de los parámetros (nombre, tipo) especificados, valor retornado (nombre, tipo) y la utilización de otros servicios para la realización de las operaciones definidas. En este último caso se deberá simular dicha interacción mediante stubs de los servicios invocados.

Roles: El rol responsable de esta actividad es el Implementador que programa los servicios definidos. El Responsable de Verificación debe controlar que esta actividad se realice y que los Reportes de verificación unitaria sean generados.

Entregables de entrada: Esta actividad tiene como entregables de entrada el componente que implementa el servicio a probar, el Modelo de Servicios y el Plan de verificación de la iteración, realizando las pruebas del servicio contra la especificación del mismo mediante los casos de prueba unitaria definidos para éste.

Entregables de salida: Esta actividad tiene como entregable de salida el Reporte de verificación unitaria de servicios.

Modelado de la actividad: en la figura 7.6 se muestra el modelado de la actividad utilizando los elementos de modelado provistos por el RUP.

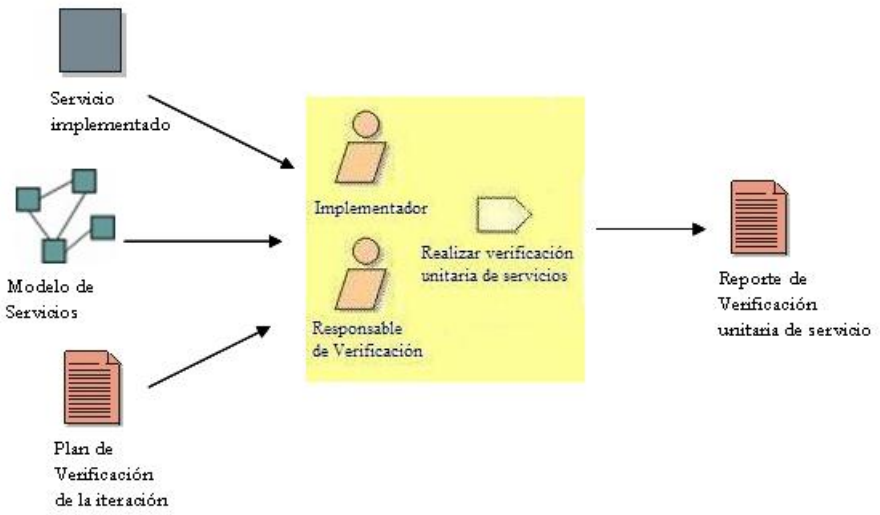


Figura 7.6: Modelado de la actividad Realizar verificación unitaria de servicios (I7) en la Extensión SOA mejorada

Fundamentación: Esta actividad se agrega para dar soporte a la calidad de los servicios liberados desde implementación para su integración con el resto de los servicios definida para la iteración y hasta conformar el sistema como un todo. La actividad definida para la Disciplina de Implementación, Implementar Servicios (I13) hace énfasis en la construcción de los componentes asignados a servicios como está definido en los modelos de Diseño y Servicios correspondientes, en esta actividad se hace énfasis en la verificación unitaria del cumplimiento de dichas definiciones, por parte del componente implementado, antes de su liberación.

7.1.2.2. Disciplina Verificación

El propósito de la disciplina de Verificación en la metodología SOA propuesta agrega los siguientes objetivos a los definidos en el proceso base: utilizar como unidad de verificación los servicios para realizar las pruebas del software, realizando verificación unitaria, de integración y de sistema, así como pruebas de regresión para asegurar que se mantiene el mismo funcionamiento del software en versiones siguientes a la ya testeada. Para la realización de tests para los que no se cuente con todas las unidades requeridas, se utilizarán stubs y drivers según corresponda, que podrán ser provistos desde la Disciplina Implementación.

En esta Disciplina se agregan los roles definidos en el proceso base Responsable de Verificación que tendrá a su cargo la gestión administrativa y técnica de las actividades de verificación. Para cumplir con los objetivos agregados para la Disciplina de Verificación, se modifican las siguientes actividades existentes en el proceso base: Planificar las pruebas de la iteración (V3), Especificar los Casos de Prueba (V4) y Ejecutar las pruebas (V7). El resto de las actividades existentes en la Disciplina Verificación no se modifican ya que de la forma en que están definidas son adecuadas para el enfoque SOA. A continuación se describen estas actividades y el resto de los elementos definidos, siguiendo la forma de presentación de la metodología SOA propuesta.

Planificar las Pruebas de la iteración (V3)

Objetivo: Esta actividad tiene como objetivo planificar las pruebas para la iteración. Para esto se deben establecer claramente las fechas en que se realizarán las verificación sobre los componentes asignados a los servicios y sus integraciones, en forma sucesiva hasta el final de la iteración, así como las responsabilidades sobre la verificación de los distintos elementos.

Descripción: Se deben planificar las pruebas de la iteración, teniendo en cuenta las planificaciones realizadas de desarrollo y de integración de la iteración, con las que se deben coordinar las fechas y procedimientos de trabajo. Esta planificación debe incluir los elementos que se deben verificar en la iteración, esto es los componentes asignados a servicios y las sucesivas integraciones que se realizarán de los mismos en la iteración, quien debe hacer cada verificación, cuando se debe comenzar a realizar cada verificación y en que fecha debe entregar el informe de verificación correspondiente. Para esto se debe confirmar si ya existen los casos de prueba necesarios para verificar los componentes asignados a servicios correspondientes a la iteración en el Modelo de Casos de Prueba, y si no es así, se deben definir ejecutando la actividad Especificar Casos de Prueba para los servicios correspondientes. También debe planificar la fecha de comienzo de la Evaluación de la verificación, quien realiza la evaluación y en que fecha se debe entregar el documento Evaluación de la Verificación.

Roles: El rol responsable de esta actividad es el Responsable de Verificación, y participan también los Asistentes de Verificación integrantes del equipo de verificación, y el Coordinador de desarrollo que conoce el plan de desarrollo y el plan de integración de la iteración.

Entregables de entrada: Esta actividad tiene como entrada el Plan de desarrollo y el Plan de Integración de la iteración, planificaciones con las que debe estar alineada la verificación a realizar en la iteración, y el Modelo de Casos de Prueba para confirmar si existen o no los casos de prueba correspondientes a los componentes asignados a servicios que se probarán en la iteración.

Entregables de salida: Esta actividad tiene como entregable de salida el Plan de Pruebas de la iteración.

Modelado de la actividad: en la figura 7.7 se muestra el modelado de la actividad utilizando los elementos de modelado provistos por el RUP.

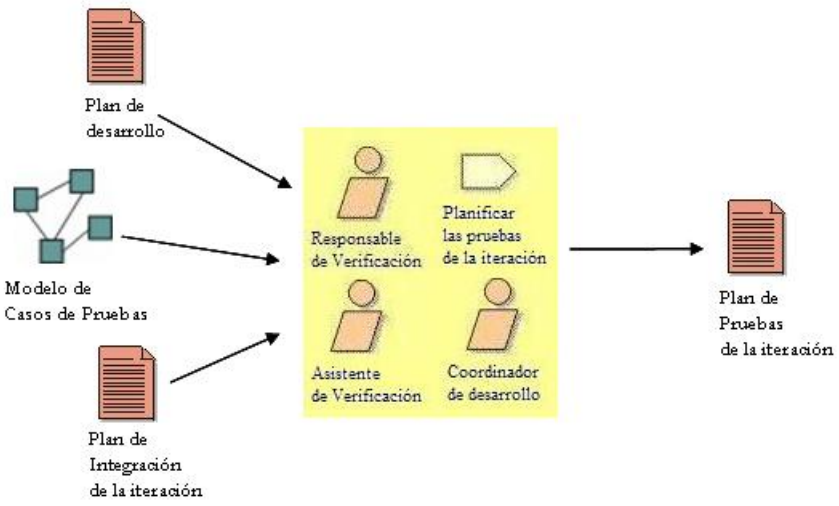


Figura 7.7: Modelado de la actividad Planificar las pruebas de la iteración (V3) en la Extensión SOA mejorada

Fundamentación: Esta actividad, de la misma forma que la actividad Planificar la Integración de la iteración (I4), se agrega para dar soporte a la gestión del desarrollo basado en servicios, donde la planificación del desarrollo, integración y verificación de la iteración, se realiza referenciando los componentes asignados a servicios que se obtendrán como resultado de la misma. Es una actividad de soporte ya que lo que agrega es el foco en la unidad de gestión que son los servicios, pero mantiene el concepto definido para la actividad que es realizar la planificación de las pruebas de la iteración, para trabajar con un cronograma de desarrollo, integración y verificación definido y coordinado para la iteración, entre las distintas áreas involucradas.

Especificar los Casos de Prueba (V4)

Objetivo: El objetivo de esta actividad es realizar la especificación completa de los casos de prueba para el sistema bajo desarrollo. Los casos de prueba deben corresponder a las pruebas unitarias a realizar sobre los servicios definidos en la Disciplina de Implementación, a las pruebas de integración de componentes asignados a servicios a realizar en las sucesivas iteraciones, y a la prueba del sistema de extremo a extremo, incluyendo las pruebas correspondientes a los procesos del Negocio definidos como Casos de Uso del Negocio. También se especificarán casos de prueba para los requerimientos de calidad definidos sobre los servicios y el sistema.

Descripción: El Responsable de Verificación, junto con los Asistentes de Verificación realizarán la especificación de los casos de prueba para la verificación de los servicios implementados en componentes del sistema, integración de los mismos y el sistema en sí. Se deberán basar en los Casos de Uso del Sistema y del Negocio para diseñar los casos de pruebas para pruebas de caja negra, y en la especificación de los servicios en el Modelo de Servicios, buscando combinaciones de actores, entradas, salidas y estados iniciales que generen escenarios interesantes que empleen los objetos participantes en los diagramas. Se deben especificar también casos de prueba de carga y otros requerimientos de calidad especificados para los servicios y el sistema. La especificación de cada caso de prueba debe contener: Número de caso de prueba, componente que implementa el servicio sobre el que se realiza, o componentes si es una integración, funcionalidad que se verifica, entrada, salida esperada y salida real.

Roles: El rol responsable de esta actividad es el Responsable de Verificación, y participa también el rol de Asistente de Verificación.

Entregables de entrada: Esta actividad tiene como entrada el Modelo de Casos de Uso del Negocio, Modelo de Casos de Uso del Sistema, el Modelo de Diseño, el Modelo de Servicios, el Modelo de Implementación y el Plan de Pruebas de la iteración para tener la información de los componentes asignados a servicios y funcionalidades que serán probadas en dicha iteración.

Entregables de salida: Esta actividad tiene como salida el Modelo de Casos de Pruebas.

Modelado de la actividad: en la figura 7.8 se muestra el modelado de la actividad utilizando los elementos de modelado provistos por el RUP.

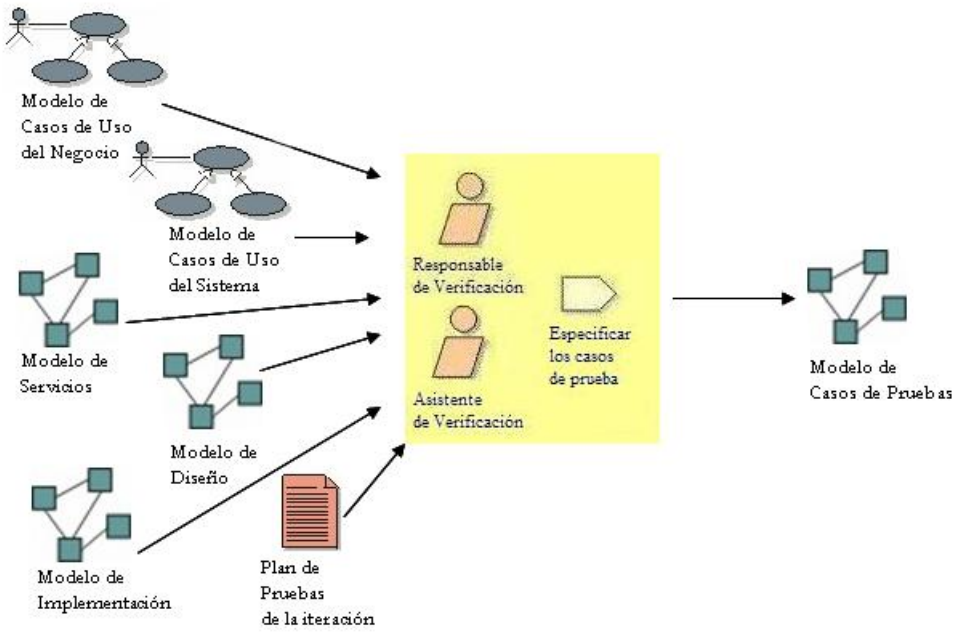


Figura 7.8: Modelado de la actividad Especificar los Casos de Prueba (V4) en la Extensión SOA mejorada

Fundamentación: Esta actividad se agrega para dar soporte a la calidad de los servicios construidos, desde el punto de vista de la verificación del cumplimiento de los mismos por parte de los componentes asignados a servicios implementados. Hace énfasis en la investigación de servicios definidos para realizar los Casos de Uso del Negocio y del Sistema, verificando el cumplimiento de las funcionalidades y atributos de calidad asignados a los mismos, al especificar los casos de prueba que serán ejecutados contra éstos.

Esta actividad es importante para la liberación de un producto compuesto por servicios que cumplen su especificación y para la confianza que se pueda tener por parte de otros proyectos que los reutilicen, sobre el funcionamiento de los mismos y el cumplimiento de los contratos publicados, por parte de los componentes que los implementan. Sin embargo, la verificación de servicios en un desarrollo SOA es un área en si misma, que no se pretende cubrir completa ni exhaustivamente con la inclusión de esta actividad, simplementa brindar guías a partir de la definición de actividades específicas, que sean tenidas en cuenta como primer aproximación para la verificación cabal de los servicios definidos.

Ejecutar las Pruebas (V7)

Objetivo: Esta actividad tiene como objetivo realizar las pruebas sobre los componentes asignados a servicios que se implementan en la iteración, así como sobre las sucesivas integraciones que se realicen en la iteración.

Descripción: Esta actividad consiste en verificar un componente asignado a un servicio, o varios luego que se integran al resto del sistema. El Responsable de Verificación distribuirá las tareas de verificación entre los Asistentes de Verificación, incluyéndose él mismo, con los Casos de prueba correspondientes a los componentes asignados a servicios y a la integración de los mismos que se realizará en la iteración. Los Asistentes de Verificación ejecutarán los casos de prueba asignados y devolverán los reportes de dichas pruebas al Responsable de Verificación. Éste consolidará estos informes en un informe de Verificación de sistema donde se especificarán las fallas encontradas. Esta verificación se debe realizar por personas que no hayan participado directamente en la implementación de el o los servicios que se están verificando.

Roles: El rol responsable de esta actividad es el Responsable de Verificación, y participa también el rol Asistente de Verificación.

Entregables de entrada: Esta actividad tiene como entrada el Modelo de Casos de Prueba, el Plan de Pruebas de la iteración, el Plan de desarrollo, el Plan de Integración de la iteración y los componentes a probar.

Entregables de salida: Esta actividad tiene como salida los Reportes de Pruebas y el Reporte de pruebas de la Integración.

Modelado de la actividad: en la figura 7.9 se muestra el modelado de la actividad utilizando los elementos de modelado provistos por el RUP.

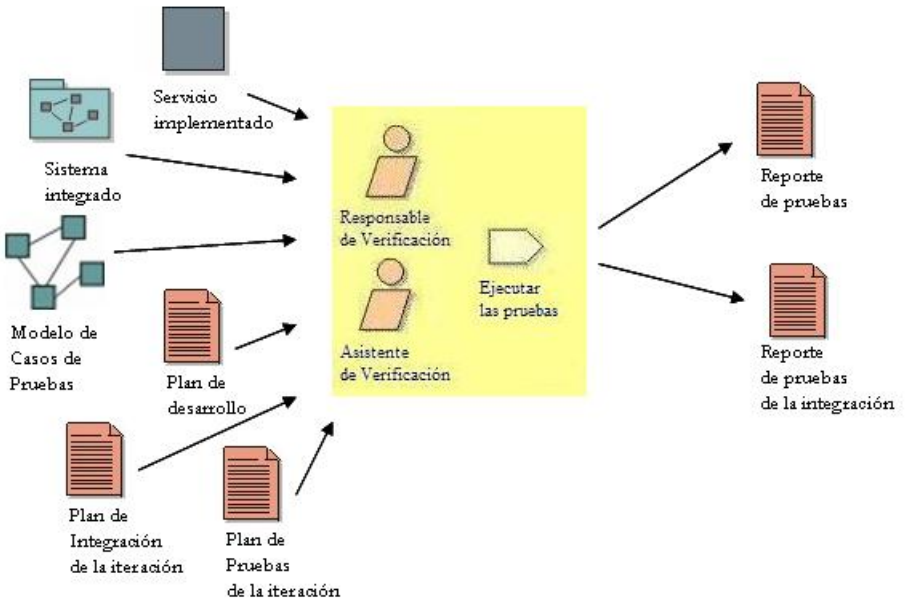


Figura 7.9: Modelado de la actividad Ejecutar las Pruebas (V7) en la Extensión SOA mejorada

Fundamentación: Esta actividad se agrega para dar soporte a la verificación de servicios basada en las definiciones realizadas en el Modelo de Casos de Prueba, enfatizando la orientación a servicios de los casos de prueba definidos para realizar dicha verificación. Esta actividad solamente agrega soporte a la ejecución de las pruebas sobre los servicios, para verificar el cumplimiento de la especificación de servicios realizada en los contratos de los mismos, para detectar fallas antes de ser aprobado para integrar la línea base del producto en desarrollo.

7.1.2.3. Disciplina Gestión de Configuración

El propósito de la disciplina de Gestión de Configuración en la metodología SOA propuesta agrega los siguientes objetivos a los definidos en el proceso base: utilizar como unidad de configuración los servicios definiendo una estructura de configuración adecuada que permita el fácil acceso e integración de servicios en distintos proyectos.

En esta Disciplina se agrega el rol definido en el proceso base Responsable de Gestión de Configuración que tendrá a su cargo la gestión administrativa y técnica de la configuración. Para cumplir con estos objetivos definidos se modifica la actividad Definir ambiente controlado (C4) para guíar la definición de la estructura del repositorio. El resto de las actividades existentes en la Disciplina Gestión de Configuración no se modifican ya que de la forma en que están definidas son adecuadas para el enfoque SOA. A continuación se describe la actividad modificada y los elementos correspondientes, en la forma de presentación de la metodología SOA propuesta.

Definir ambiente controlado (C4)

Objetivo: Esta actividad tiene como objetivo definir y generar la estructura para el ambiente controlado que se utilizará para el manejo de las versiones de los elementos de configuración.

Descripción: En esta actividad se realiza la definición y construcción del ambiente controlado donde se almacenarán, modificarán y consultarán los elementos de la línea base. Para la estructura del repositorio de configuración se recomienda mantener los servicios en diferentes proyectos, ya que estos serán compartidos por varias aplicaciones, por lo que es importante mantener su independencia de versiones también. Para las bibliotecas y otros elementos que serán compartidos por distintos proyectos se recomienda también colocarlas agrupadas en proyectos distintos. Luego de definido el repositorio para la gestión de configuración y los procedimientos de uso asociados, se transmitirá al equipo de proyecto como se debe utilizar.

Roles: El rol responsable de esta actividad es el Responsable de la Gestión de Configuración.

Entregables de entrada: Esta actividad tiene como entrada el Plan de Gestión de la Configuración y el Modelo de Servicios, así como el manual técnico de la herramienta de gestión de configuración que se utilice.

Entregables de salida: Esta actividad tiene como salida el ambiente controlado definido y generado, y el documento de Manejo del ambiente controlado.

Modelado de la actividad: en la figura 7.10 se muestra el modelado de la actividad utilizando los elementos de modelado provistos por el RUP.

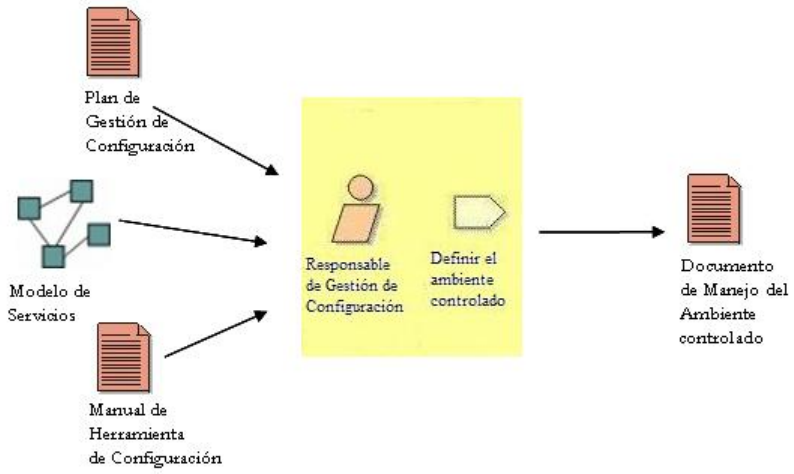


Figura 7.10: Modelado de la actividad Definir el ambiente controlado (C4) en la Extensión SOA mejorada

Fundamentación: La definición de la estructura para la Gestión de la Configuración sobre la implementación, puede realizarse de la forma que el proyecto considere más adecuada, sin embargo, la recomendación realizada en esta actividad aporta a mejorar dicha gestión, permitiendo que la unidad definida para realizarla sean los servicios. Esto trae aparejado varios beneficios como la disponibilidad de las distintas versiones asociadas a los servicios en forma ordenada y fácil de obtener, tanto para el sistema en desarrollo como para otros proyectos en la Organización que pudieran necesitar incluirlos.

7.1.2.4. Disciplina Gestión del Proyecto

El propósito de la disciplina de Gestión del Proyecto en la metodología SOA propuesta agrega los siguientes objetivos a los definidos en el proceso base: utilizar como unidad de planificación del desarrollo los servicios definiendo para cada iteración que conjunto de componentes asignados a servicios serán implementados, integrados y liberados en la versión de la iteración.

En esta Disciplina se agrega el rol definido en el proceso base Coordinador de desarrollo que tendrá a su cargo la gestión administrativa y técnica del desarrollo. Para cumplir con estos objetivos definidos se modifica la actividad Ajustar y controlar el desarrollo (G9) para guíar la planificación del desarrollo basada en servicios. El resto de las actividades existentes en la Disciplina Gestión del Proyecto no se modifican ya que de la forma en que están definidas son adecuadas para el enfoque SOA. A continuación se describe la actividad modificada y los elementos correspondientes, siguiendo la forma de presentación de la metodología SOA propuesta.

Ajustar y controlar el desarrollo (G9)

Objetivo: Esta actividad tiene como objetivo realizar la planificación detallada del desarrollo de la iteración que se está por comenzar, al culminar la iteración actual, con base en los servicios que se van a desarrollar. La planificación utilizará como unidad el servicio, definiendo que componentes asignados a servicios serán implementados teniendo en cuenta las guías de planificación general del proyecto y las dependencias entre los distintos componentes que corresponden a los servicios a implementar.

Descripción: Se debe realizar una reunión del equipo de desarrollo con el Coordinador del desarrollo para tratar los temas relativos a la planificación del desarrollo. En este cronograma se incluirán las actividades a realizar, responsables de las mismas, actividades críticas, hitos, nuevas fechas de ajuste de cronograma. El alcance definido constituye la guía básica de los elementos a desarrollar en cada iteración y el Plan de la iteración que contiene la planificación macro de actividades específicas para la iteración, servirá de guía para definir el detalle del desarrollo. Para planificar el desarrollo se utilizarán los servicios como unidad de planificación, teniendo en cuenta la correspondencia de los mismos con los Casos de Uso y del Negocio que se desarrollarán en la iteración. Los Modelos de Diseño, Servicios e Implementación servirán para tener en cuenta las dependencias entre los componentes asignados a servicios que se desarrollarán.

Roles: El rol responsable de esta actividad es el Coordinador de Desarrollo y participan también los Implementadores que realizan el desarrollo.

Entregables de entrada: Esta actividad tiene como entrada el Modelo de Casos de Uso del Sistema, el Modelo de Casos de Uso del Negocio, el Modelo de Diseño, el Modelo de Servicios, el Modelo de Implementación, el Alcance del Sistema, el Plan de la iteración y el Plan de desarrollo actual, para el caso que se esté realizando un ajuste.

Entregables de salida: Esta actividad tiene como salida el Plan de desarrollo.

Modelado de la actividad: en la figura 7.11 se muestra el modelado de la actividad utilizando los elementos de modelado provistos por el RUP.

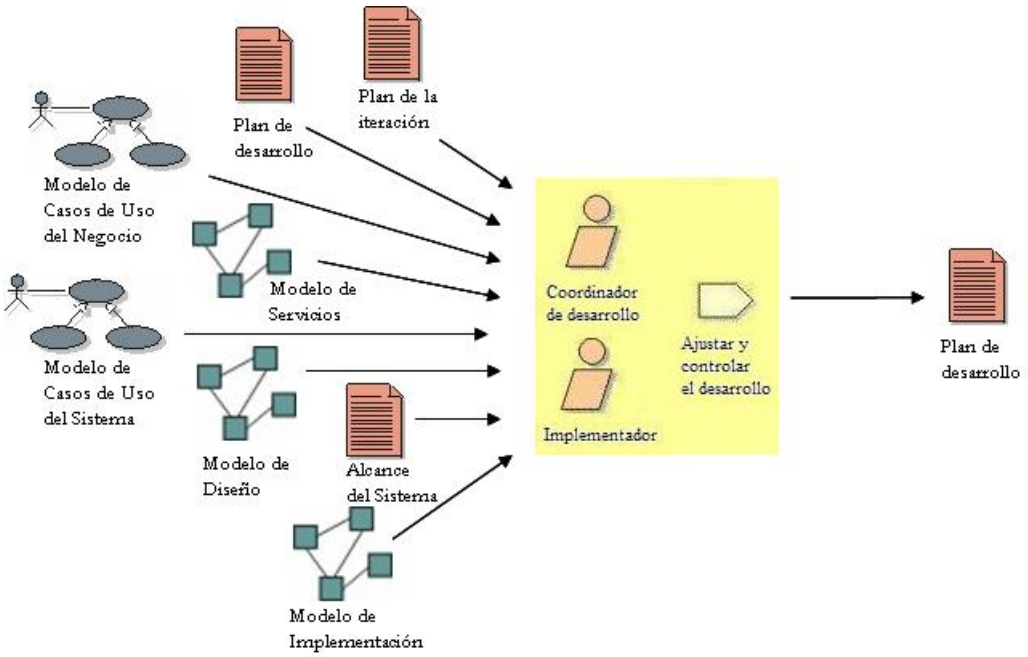


Figura 7.11: Modelado de la actividad Ajustar y controlar el desarrollo en la Extensión SOA mejorada

Fundamentación: Esta actividad se agrega para dar soporte a la gestión del desarrollo basado en servicios, donde la planificación del desarrollo, integración y verificación de la iteración, se realiza referenciando los componentes asignados a servicios que se obtendrán como resultado de la misma. Es una actividad de soporte ya que lo que agrega es el foco en la unidad de gestión que son los servicios, pero mantiene el concepto definido para la actividad que es realizar la planificación del desarrollo para la iteración, de forma de trabajar con un cronograma de desarrollo, integración y verificación definido para la iteración y coordinado entre las distintas áreas.

7.1.2.5. Disciplina Gestión de Calidad

El propósito de la disciplina de Gestión de Calidad en la metodología SOA propuesta agrega los siguientes objetivos a los definidos en el proceso base: identificar propiedades de calidad que deben cumplir los servicios definidos e implementados en el desarrollo del sistema, de forma que sean tenidas en cuenta en el diseño e implementación de los mismos y que se pueda realizar la verificación del cumplimiento de dichas propiedades.

En esta Disciplina se agrega el rol definido en el proceso base Responsable de Gestión de Calidad que tiene a su cargo la identificación de las propiedades de calidad para los servicios y la evaluación del cumplimiento de las mismas. Para cumplir con estos objetivos definidos se modifica la actividad Identificar las propiedades de calidad (Q1) para enfocar dicha identificación sobre servicios. El resto de las actividades existentes en la Disciplina Gestión de Calidad no se modifican ya que de la forma en que están definidas son adecuadas para el enfoque SOA. A continuación se describe la actividad modificada y los elementos correspondientes, siguiendo la forma de presentación de la metodología SOA propuesta.

Identificar las propiedades de calidad (Q1)

Objetivo: Esta actividad tiene como objetivo definir aquellas propiedades que permitan evaluar la calidad, se deben identificar los productos que deben ser evaluados para la calidad, además

de definir los criterios para evaluar la calidad de cada producto. Especificamente se deben identificar las propiedades de calidad sobre los procesos del Negocio identificados y los servicios que se deben desarrollar para cumplirlos.

Descripción: En esta actividad se deben encontrar propiedades de calidad, algunas pueden ser, usabilidad, eficiencia, mantenibilidad, seguridad, portabilidad, etc. como específicos del sistema a construir. Específicamente para los servicios se deben establecer, si los hay, requerimientos de performance, nivel de servicios, seguridad, entre otros. Es importante que estas propiedades queden claramente establecidas de forma de poder realizar la verificación de las mismas.

Roles: El rol responsable de esta actividad es el Responsable de la Gestión de Calidad, y participa también el Arquitecto como referencia de los servicios identificados.

Entregables de entrada: Esta actividad tiene como entrada las Actas de Requerimientos, el Modelo de Casos de Uso del Negocio, el Modelo de Casos de Uso del sistema, el Modelo de Servicios y el Plan de Calidad.

Entregables de salida: Esta actividad tiene como salida el Plan de Calidad, con la sección correspondiente a las propiedades de calidad informada.

Modelado de la actividad: en la figura 7.12 se muestra el modelado de la actividad utilizando los elementos de modelado provistos por el RUP.

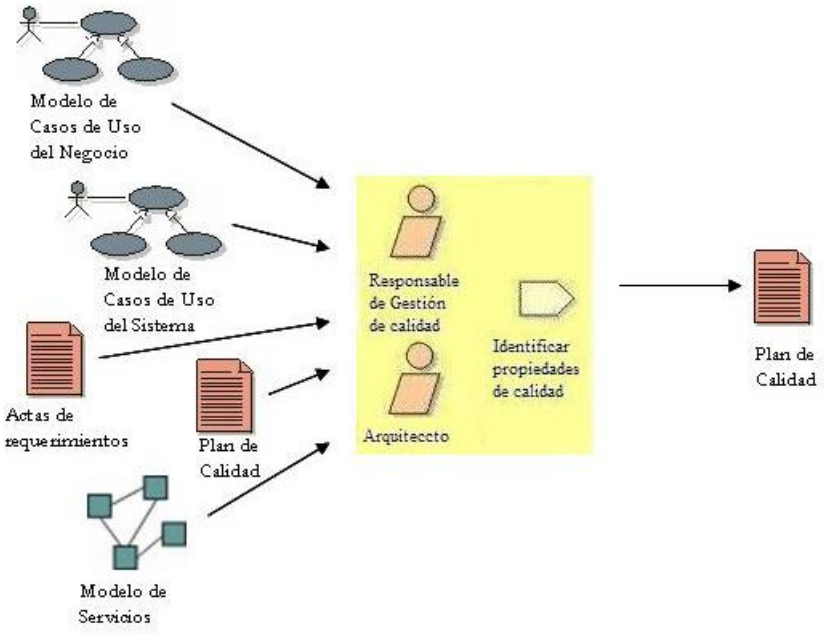


Figura 7.12: Modelado de la actividad Identificar propiedades de calidad (Q1) en la Extensión SOA mejorada

Fundamentación: esta actividad se propone para hacer énfasis en la necesidad de identificar en forma explícita las propiedades de calidad para los servicios que serán desarrollados. Esta identificación se realiza traduciendo los requerimientos de calidad existentes sobre los Casos de Uso del Negocio y del Sistema, a los servicios definidos en el Modelo de Servicios. Si bien la identificación es realizada por el Responsable de la Gestión de Calidad, participa también el Arquitecto dado su conocimiento y visión sobre los servicios definidos. El plan de calidad resultante deberá ser luego validado con el Cliente para aprobar las propiedades de calidad identificadas y/o modificar o agregar las que considere.

7.1.3. Generalización de la Metodología SOA propuesta

El alcance de la metodología SOA propuesta está limitado a la ejecución de un proyecto de desarrollo de software específico, y por lo tanto, no prescribe otras acciones a realizar por fuera de ese proyecto. Por lo que, el alcance de la generalización propuesta está limitado también a la realización de un proyecto específico. Esto restringe la visión general de desarrollo y no define entonces otros elementos organizacionales que apoyen la realización de dicho proyecto en la Organización. Como se presentó en el Capítulo 4 cuando Organización decide encarar un proyecto SOA a nivel organizacional, se deben tener en cuenta otros aspectos para su implantación, que impactan en la Organización como un todo.

Por ejemplo, como se plantea en la visión de SOA de [KRAF05], para encarar un proyecto SOA a nivel organizacional es imprescindible contar con apoyo político de la gerencia, e instaurar un comité SOA para controlar y promover el uso de servicios en toda la Organización, tanto para gerenciar los proyectos como para sincronizar la definición y desarrollo de servicios que son compartidos por varios proyectos y subproyectos. Mediante los contratos de servicio se establecen las interfaces entre las distintas capas de servicios y los datos, permitiendo el desarrollo en paralelo y la estabilidad del proyecto alrededor de las decisiones y compromisos tomados en la definición de estos contratos.

En la vision de SOA de[ERLT05] se incluye una Fase para la Administración de servicios, que se realiza luego de el despliegue de los mismos en el ambiente de producción. La administración de servicios corresponde entonces a las actividades que se realizan luego de la Implantación del software en el ambiente de producción, y puede asimilarse a la infraestructura de administración del software que se tenga en la Organización, incluyendo una visión específica para la administración de los servicios de la Organización.

Para la generalización de la metodología SOA propuesta con el alcance definido, se estudió entonces como incluir las actividades, entregables y roles definidos en el modelo de procesos que plantea el Marco Metodológico COMPETISOFT [COMP06]. En la reunión de coordinación del proyecto realizada en diciembre de 2006 al finalizar el primer año del mismo en el marco del Congreso FIBECYT [FIBE06] en Buenos Aires, Argentina, además de otras definiciones, se vió la oportunidad de agregarle extensiones al modelo de procesos definido para desarrollo SOA y MDA, en la forma de perfiles.

Estos perfiles, tienen el mismo enfoque que el utilizado para realizar las extensiones del proceso base adaptación del RUP del Gris, los que modifican/agregan/eliminan actividades, entregables y roles del modelo de proceso definido. Por lo que, para realizar un proceso de desarrollo con enfoque SOA utilizando COMPETISOFT, se realiza el proceso de desarrollo definido en la categoría Operación más el perfil SOA agregado.

Se tenía familiaridad con las actividades, entregables, roles y demás elementos del proceso de desarrollo de COMPETISOFT por haber integrado el grupo de trabajo de las Fases de Análisis y Diseño de ese proceso, particularmente como parte del grupo de trabajo de Diseño y Arquitectura de Software que revisó la consistencia de las actividades, entregables y roles previstos, realizando guías específicas de implantación para las distintas actividades y templates para algunos de los entregables.

El principal trabajo en esta generalización estuvo dado por la definición del "encastre" de los elementos definidos en la metodología SOA propuesta con los elementos del proceso de desarrollo, y con otros procesos afectados por las modificaciones, más el modelado del perfil SOA utilizando el patrón de procesos que provee COMPETISOFT. Los principales elementos del Perfil SOA (PSOA) realizado que fue enviado como propuesta a los integrantes de COMPETISOFT, se adjunta en el Anexo C.

Los elementos agregados como parte del Perfil SOA definido son los incluidos en las Disciplinas de Modelado del Negocio, Diseño, Implementación y Verificación de la metodología SOA propuesta, que corresponden a las Fases Modelado del Negocio (agregada), Diseño, Construcción e Integración, y Pruebas. Como resultado de esta generalización realizada para la definición del

perfil SOA de desarrollo en COMPETISOFT, se considera que la metodología SOA propuesta contiene varios aspectos clave a tener en cuenta en este tipo de desarrollo, que son fácilmente adaptables a los modelos y procesos de desarrollo que se definan en la Organización.

La adaptación de algunos de los elementos agregados en la mejora de la metodología SOA a las Disciplinas de Gestión resulta más difícil de realizar, ya que las actividades, entregables y roles que se definen por ejemplo para las Disciplinas de Gestión de Proyecto y Gestión de Configuración, cruzan el proceso de desarrollo en el modelo de procesos de COMPETISOFT, por lo que integrarlas en una visión más global no resulta trivial. Por lo tanto se decidió no incluir en el Perfil SOA las actividades de gestión agregadas en la mejora de la metodología SOA.

En el caso de COMPETISOFT no se indica la utilización de una metodología de desarrollo específica en el proceso definido, por lo que hubo que abstraer de las actividades y entregables los elementos inherentes a la utilización del RUP, como ser los distintos modelos incluyendo el Modelo de Casos de Uso, Diseño, Implementación y Servicios definidos, cuya especificación fue transformada, siguiendo la terminología definida en COMPETISOFT, a Documento de Especificación del Sistema, Documento de Diseño, y agregando el Documento de servicios, para seguir los lineamientos del proceso definido.

Para la realización de las guías de adaptación y templates asociados si se incluyeron los elementos específicos al RUP, ya que tanto las guías como los templates sirven para brindar alternativas sobre de las distintas técnicas, herramientas y metodologías que es posible utilizar para realizar las distintas actividades y entregables definidos en el modelo de procesos.

7.1.4. Descripción de nuevos enfoques metodológicos

El enfoque metodológico más importante a ser comentado en esta sección lo constituye el plug-in para desarrollo SOA [RSOA] incorporado al RUP en el Rational Method Composer (RMC) [RMC] liberado por primera vez a mediados del año 2005, el cual constituye una plataforma de ingeniería de procesos para definir y publicar procesos de desarrollo y contenidos asociados, de la misma forma que su equivalente liberado en Eclipse, el framework Eclipse Process Framework (EPF) [EPF]. Ambos frameworks se basan en el metamodelo Unified Modeling Architecture (UMA) [RMCK], [EPF] que define un esquema y terminología para representar procesos de desarrollo de software como en SPEM [SPEM05] basado en SPEM 1.1 . Entre los cambios de terminología que se incorporan respecto a versiones anteriores del RUP, se pueden mencionar el de actividad por tarea y el de artefactos por productos de trabajo, que como se mencionó en el Capítulo 2 son de SPEM.

La primera liberación incluía un plug-in para desarrollo SOA que no fue posible evaluar en ese momento, ya que además la investigación inicial de referencias para la definición de la metodología SOA propuesta ya había sido realizada, y la metodología ya estaba definida y su prueba por comenzar. El plug-in SOA del RMC fue evolucionando durante el año 2006 liberandose dos versiones más, siendo en noviembre de 2006 la liberación de la versión actual que conjunta la visión inicial de SOA planteada en el RMC con el enfoque de los trabajos presentados en 4.2.4.1 específicamente el enfoque Service Oriented Modeling Architecture (SOMA). Utilizando un trial del RMC y publicando el template que viene ya preparado denominado “Classic RUP for SOMA”, se obtuvo la definición del proceso para desarrollo SOA que plantea el RMC, correspondiente a “Version 2.4 Based on: RUP Version: 7.0.1 Business Modeling Version: 3.0” que se comenta brevemente a continuación.

Por una lado el plug-in RUP SOMA presenta como se indica en [RSOA] “un conjunto de actividades neutrales a cualquier método, requeridas por cualquier proceso para el desarrollo de soluciones orientadas a servicios. Este proceso conceptual era común tanto al plug-in SOA del RUP como al método SOMA, y permitió una integración relativamente simple del contenido de ambos modelos.” Esta visión presenta cuatro fases para el desarrollo de servicios, que no son las del RUP, denominadas: Identificación, Especificación, Realización y Despliegue de servicios.

Cuando se mira el proceso desde la vista del RUP SOMA, se muestran esas fases y sus tareas asociadas, que luego se muestran también asignadas a las Disciplinas de Modelado del Negocio, Análisis&Diseño e Implementación. Este hecho coincide con la visión de la metodología SOA propuesta en cuanto a la identificación de las Disciplinas del conjunto núcleo inicial definidas para el desarrollo SOA.

A continuación se presenta la definición de tareas por Disciplina en el plug-in RUP SOMA, incluyendo una selección de sus principales objetivos para dar una idea de que trata la actividad:

- Modelado del Negocio:
  - Identificar objetivos del Negocio e indicadores de desempeño claves (Key performance indicators - KPI): identificar objetivos con los cuales el negocio puede ser planificado y gestionado mediante los indicadores de desempeño, proveer las bases para la medición y mejora de las actividades del negocio.
  - Análisis de áreas funcionales: tomar ideas iniciales de partición del negocio y refinarlas con base en la identificación de conjuntos cohesivos de funciones del negocio, para obtener áreas funcionales que se asignan a sistemas del negocio (conjunto de roles y recursos con un propósito específico para el negocio).
  - Refinar Caso de Uso del Negocio: esta tarea se realiza cuando hay Casos de Uso del Negocio que tienen solo una definición en alto nivel, que es necesario refinar para poder realizarlos.
- Análisis&Diseño
  - Identificar aspectos comunes y variables: esta tarea es aplicable a un número de elementos de análisis y diseño donde a partir de la factorización de aspectos variables y comunes en esos elementos se obtienen resultados más robustos y flexibles a la vez.
  - Identificar patrones de seguridad: durante la definición de la Arquitectura de Software inicial del sistema el Arquitecto de Seguridad es responsable por la identificación y selección de patrones clave de seguridad para asegurar el nivel de seguridad requerido por el sistema.
  - Especificación de Componentes (SOA): esta tarea especifica los detalles de los componentes de servicios que realizan los subsistemas de diseño. Los servicios utilizados en una solución basada en SOA son realizados mediante componentes de servicios que pertenecen a un subsistema alineado con una función específica del negocio.
  - Construir Prueba de Concepto de la Arquitectura (SOA): esta tarea extiende la misma tarea que existe en el RUP para cualquier desarrollo, definiendo como desarrollar una prueba de concepto de la arquitectura para una solución SOA, basada en requerimientos arquitectónicos existentes y perfil de riesgos.
  - Identificar y asociar servicios a objetivos: la asociación de servicios a objetivos vincula servicios con objetivos del negocio e identifica servicios analizando los objetivos de una empresa o unidad del negocio.
  - Analizar los procesos del Negocio: esta tarea identifica los elementos de diseño de una solución orientada a servicios en términos de servicios y particiones, y documenta la especificación inicial de esos servicios.
  - Analizar el modelo de datos: la mayoría de los proyectos utilizan como tecnología para persistir sus datos bases de datos relacionales. El diseñador de la base de datos debe definir los detalles de este diseño, incluyendo tablas, índices, vistas, constraints, triggers, procedimientos almacenados y otras construcciones necesarias para almacenar, recuperar, consultar y eliminar objetos persistentes.

- Analizar activos existentes: se realiza una evaluación del valor para el negocio y calidad técnica de aplicaciones existentes para identificar servicios que se puedan derivar de éstas. Para esto se realiza un mapeo de granularidad gruesa entre los procesos del negocio y las funcionalidades provistas por las aplicaciones existentes, dando lugar a una definición de servicios candidatos que se incluye en el portfolio de servicios de la empresa.
- Analizar reglas del Negocio: se analizan las reglas del negocio definidas en los procesos del Negocio que deben ser tenidas en cuenta en la definición de servicios y relaciones existentes.
- Analizar Casos de Uso del Negocio (SOA): esta tarea utiliza la realización de Casos de Uso del Negocio como entrada e identifica un conjunto de servicios candidatos que se incluyen en el portfolio de servicios del proyecto. Estos servicios candidatos pueden requerir refinamiento adicional, pero es posible obtener en esta tarea un conjunto inicial de especificación de servicios.
- Diseñar mensajes: esta tarea describe las acciones requeridas para desarrollar un modelo de diseño de mensajes, teniendo en cuenta los patrones de intercambio de mensajes y la relación del modelo de dominio con el modelo de mensajes.
- Aplicar tests de filtro: esta tarea describe el requerimiento para calificar servicios candidatos de forma de asegurar que los que se desarrollan cumplen con las necesidades del negocio. La idea es evitar el "síndrome de proliferación de servicios" donde todo se cataloga como servicio sin importar su valor ni granularidad.
- Especificar Servicios: esta tarea define y especifica los servicios y la estructura de una solución orientada a servicios en términos de las colaboraciones entre los subsistemas de diseño que contiene y subsistemas/interfaces externos, se analizan los servicios por aspectos comunes y variables, se documenta la especificación de servicios y se determinan las dependencias y comunicación entre servicios. En esta tarea también se categorizan servicios y se definen orquestaciones y coreografías entre servicios cuando estos se compongan de otros servicios.
- Diseñar subsistemas (SOA): esta tarea extiende el diseño de subsistemas del RUP con detalles específicos de una solución SOA, especialmente donde los subsistemas fueron identificados a partir de modelos de análisis del negocio. Una vez que se realizó la transición del dominio del negocio al dominio de TI, se mapean las áreas funcionales definidas en el negocio a subsistemas, su contraparte en el dominio de TI.

- Implementación

  - Documentar decisiones de realización de servicios: esta tarea se enfoca en la realización de un modelo de servicios en términos de componentes de software que ejecutarán en los ambientes de middleware existentes.

Se puede observar que la cantidad de tareas definidas supera ampliamente las provistas por la metodología SOA propuesta, si bien hay coincidencia en la mayoría de las definiciones realizadas, por ejemplo en la definición, categorización y especificación de servcios, el modelado de procesos del Negocio como Casos de Uso del Negocio, donde esto último era esperable dado que la referencia para definir las actividades de la Disciplina Modelado del Negocio fue el RUP.

También se coincide en la definición de una tarea para especificar los componentes que implementarán los servicios definidos, pero, una diferencia es que los componentes que implementan los servicios, en la metodología SOA propuesta se implementan en una actividad específica definida en Implementación, mientras que en este plug-in se incluye en la actividad del RUP Implementar elementos de Diseño. Otra divergencia se encuentra en que los servicios son derivados directamente de elementos del Negocio sin pasar por la Disciplina de Requerimientos, especificamente no se pudo encontrar en que situación queda el Modelo de Casos de Uso y la

Definición de la Arquitectura y el SAD con respecto a la inclusión del enfoque de servicios y los Modelos de Casos de Uso del Negocio y de Servicios.

Se definen como principales productos de trabajo generados y utilizados como entrada en las tareas descritas, en la Disciplina de Modelado del Negocio los siguientes: dominio, objetivos, Caso de Uso, Modelo de Casos de Uso, realización de Caso de Uso, Modelo de Análisis del Negocio y en la Disciplina de Análisis&Diseño los siguientes: Modelo de objetivos-servicios, servicio componente y Modelo de Servicios. En la Disciplina Implementación no se definen productos de trabajo nuevos, ya que la salida de la actividad que se define, se incluye en el Modelo de Diseño existente del RUP.

El Modelo de Servicios contiene a su vez los siguientes productos de trabajo: servicios, mensajes, canales de servicios, contratos de servicios, gateway de servicios, partición de servicios, proveedor de servicios y especificación de servicios, donde el elemento que es el software implementado está dado por el proveedor de servicios. Si bien en la metodología SOA propuesta se define también un Modelo de Servicios, el contenido no es coincidente en su totalidad, pero si en su concepción: agrupar en un único producto de trabajo toda la información relativa a los servicios definidos y sus características. Además en la breve evaluación que se pudo realizar de este plug-in RUP SOMA, no se detectó que se incluyera en el Documento de la Arquitectura de Software (SAD) la vista correspondiente al Modelo de Servicios con la información de los servicios definidos.

Una idea interesante que se plantea en este enfoque es la creación de un portfolio de servicios empresarial, donde se encontrará la lista de servicios candidatos y su correspondencia con las aplicaciones que los proveen. Si bien en la metodología SOA propuesta se define una actividad para investigar servicios existentes, similar a la que se define en este plug-in de evaluación de activos existentes, no se prescribió la realización de un documento que listara los servicios existentes y que fuera creciendo a medida que los proyectos van desarrollando nuevos servicios e incorporandolos al conjunto de servicios de que dispone la Organización. Esto resulta una idea interesante a incluir en la metodología SOA propuesta, ya que refuerza el enfoque horizontal de creación y reutilización de servicios en toda la Organización.

Como evaluación final del plug-in RUP SOMA descrito se puede concluir que hay una importante coincidencia en el enfoque de elementos que se deben agregar al RUP para desarrollo con enfoque SOA, pero se cree que si bien las tareas definidas en el mismo aportan al desarrollo SOA, la granularidad de las mismas en algunos casos es demasiado fina lo que hace que sea una cantidad bastante importante de actividades a incluir, donde en algunos casos no se definen o no se logran ver las relaciones con el resto de los elementos del RUP, por ejemplo con los Casos de Uso del Sistema, o el Documento de la Arquitectura de Software (SAD), elementos principales de la definición del RUP clásico.

Capítulo 8

Conclusiones y trabajo futuro

8.1. Introducción

En este capítulo se presentan las conclusiones del trabajo de tesis realizado, y el trabajo a futuro en las líneas de investigación que han quedado abiertas o se han abierto a partir de la realización de este trabajo. En la sección 8.2 se presenta un resumen de los objetivos y el trabajo realizado en esta tesis, y las conclusiones que se obtienen de cada uno. En la sección 8.3 se presentan algunas líneas de investigación que constituyen posible trabajo a futuro a realizar a partir de lo obtenido en esta tesis.

8.2. Conclusiones

En este trabajo de tesis se realizó una investigación y conceptualización del enfoque de desarrollo Service Oriented Architecture (SOA) para el desarrollo de aplicaciones distribuidas con orientación a servicios, identificando las principales características que debe cumplir un producto SOA y su relación con otros enfoques como Business Process Management (BPM) para el tratamiento de procesos del Negocio y Model Driven Architecture (MDA) para el desarrollo basado en modelos. Cabe aclarar que los enfoques SOA y MDA fueron presentados en su sentido más alto de abstracción sin atarlos a la tecnología, pero existe actualmente la tendencia a considerarlos plataformas tecnológicas que permiten llevar a cabo los correspondientes paradigmas de Service Oriented Computing (SOC) y Model-Driven Development (MDD). La metodología propuesta aplica totalmente si se considera el paradigma SOC de desarrollo en el lugar de SOA.

Se presentó el estado actual del conocimiento en procesos y metodologías de desarrollo y en las disciplinas de Diseño y Arquitectura de Software, que aportaron elementos base de gran importancia para la metodología SOA propuesta, como es el proceso de desarrollo Rational Unified Process (RUP) y los conceptos asociados a la definición e importancia de la Arquitectura de Software en el proceso de desarrollo. La evolución en dichas áreas hacia la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo de software, como es la definición de la Ingeniería de Software, aportan a la madurez del área sentando las bases para el surgimiento de nuevos paradigmas como es el enfoque SOA de desarrollo.

Se definió una metodología SOA para desarrollo de aplicaciones con enfoque SOA, que se incorporó al proceso base adaptación del RUP con que cuenta el Grupo de Ingeniería de Software (Gris) [GRIS06], que representa un paso en el camino para la adopción de dicho paradigma en forma disciplinada, basada en la conceptualización y estandarización de las prácticas asociadas con el enfoque y no en una herramienta o interpretación en particular, maximizando el valor de

la comprensión y aplicación del paradigma al desarrollo de software. La definición de la metodología SOA contiene Disciplinas, Actividades, Entregables y Roles para realizar el desarrollo, así como plantillas para los entregables y guías para los aspectos más importantes.

La metodología SOA propuesta incluye nuevos elementos a utilizar en el desarrollo como son los procesos del Negocio modelados como Casos de Uso del Negocio, y el Modelo de Servicios donde se identifican, definen, especifican y reutilizan los servicios que hacen a la Organización en general y al sistema en desarrollo en particular. Tanto las actividades como los entregables definidos en la propuesta siguen la filosofía planteada por el RUP y son totalmente compatibles con el RUP básico que no incluye Modelado del Negocio ni orientación a Servicios, y cuya relación fue presentada en el capítulo 5 para cada Disciplina y Actividad descrita, incluyendo ejemplos prácticos.

Para la aplicación de la metodología SOA propuesta en el curso "Proyecto de Ingeniería de Software" se definió un conjunto núcleo de elementos a incorporar en el proceso base adaptación del RUP, en lo que se denominó Extensión SOA a dicho proceso. Para hacer accesible la información asociada a la Extensión SOA durante la prueba en el curso, se realizó un Sitio Web [WTAD05] que permite navegar fácilmente y ubicar los distintos elementos definidos para el desarrollo SOA, así como la provisión de material y links de interés para el estudio del enfoque SOA y su aplicación mediante la metodología SOA propuesta.

La aplicación de la metodología en el desarrollo del sistema de Help-Desk para ser integrado en la plataforma SOA del proyecto Link-all, demostró que constituye una guía importante para obtener un producto que cumpla con las características definidas por el enfoque SOA, apoyando dicho desarrollo en la realización de actividades específicas para el modelado de los procesos del Negocio de la Organización, y para el diseño e implementación de servicios que los realicen. Como aspectos principales a tener en cuenta se observó que es imprescindible realizar las actividades previstas en la Disciplina de Formación y Entrenamiento, además de en las herramientas a utilizar, también sobre el enfoque SOA en si mismo, de forma de incorporar los conceptos a utilizar en el desarrollo y en la aplicación de la metodología SOA.

Luego de su aplicación la metodología SOA fue ajustada y mejorada, en base a los resultados obtenidos en la prueba de la misma y agregando elementos al conjunto núcleo definido inicialmente. Principalmente se agregaron elementos de gestión y verificación con enfoque a servicios en las Disciplinas de Implementación, Verificación, Gestión de Configuración y Gestión del Proyecto. La metodología SOA ajustada y mejorada está documentada para poder ser utilizada en un nuevo proyecto de desarrollo, sin embargo, se cree que el conjunto núcleo definido inicialmente constituye una base importante para incursionar en desarrollos de este tipo, ya que provee orientación para el desarrollo sobre los dos conceptos más importantes que en mi forma de ver, se encuentran involucrados en un desarrollo SOA: especificación de procesos del Negocio, e identificación, definición y especificación de Servicios, para la implementación del Software.

El año siguiente a la prueba de la metodología SOA en el curso "Proyecto de Ingeniería de Software", fue utilizada nuevamente en el proyecto de grado "Enfoque SOA (Service Oriented Architecture) con integración de aplicaciones móviles para LIMS (Laboratory Information Management System)" [MSOA06], en el cual se desarrolló un prototipo con la suite de herramientas IBM modelando e implementando servicios con los enfoques Service Component Architecture (SCA)[SCAS] y Service Data Object (SDO)[SDOS] presentados en el Capítulo 4. Si bien no fue posible incluir la evaluación y análisis de resultados obtenidos con la metodología en el capítulo 6 de aplicación de la metodología, ya que al momento de escribir esta tesis el proyecto estaba aún en pleno desarrollo, habiendo finalizado el mismo se puede mencionar como aspecto importante evaluado en el informe entregado por los estudiantes el hecho de la compatibilidad entre la metodología SOA propuesta y los enfoques SCA y SDO, como se muestra a continuación.

"La capa de componentes de servicios (SCA) agrega un nuevo nivel de abstracción. En este nivel se definen los componentes de servicios a los que se asignará cada uno de estos, se identifican todos los servicios que serán ofrecidos por cada componente y las relaciones entre los diferentes componentes. Además, se define qué componentes de software implementarán cada uno de los

componentes de servicios. Los lineamientos propuestos por la metodología son aplicables a este tipo de arquitecturas, ya que en la definición de la actividad D9 - Asignación de servicios a componentes, se propone la asignación de los servicios a los componentes que los contendrán, en este caso componentes de servicios y la relación que tendrán estos componentes.

Entonces, siguiendo los pasos propuestos por la metodología, iniciando por la definición del negocio y la organización y el estudio y definición de los servicios para luego llegar a la definición de los componentes, en una clara estructura de razonamiento top-dowx, es posible generar una arquitectura de componentes de servicio con una división claramente enfocada al negocio. Otro aspecto interesante a mencionar son los SDO (service data objects), objetos para el intercambio de mensajes en una arquitectura de servicios basada en una SCA. Cabe destacar que la utilización de este tipo de objetos no modifica la forma en la que esta actividad se realiza. Los SDO son simplemente una de las formas de intercambiar datos, por lo que su utilización o no, no determina la definición de los componentes; sin embargo, son un elemento importante dentro de una SCA y su utilización resulta natural en una arquitectura de este tipo."

Finalmente, se evaluaron oportunidades de aplicación de la metodología SOA propuesta en la industria de Software o como parte de otros procesos de desarrollo definidos para su aplicación en la industria de Software, realizando con ese objetivo una generalización de la metodología SOA para el modelo de procesos del proyecto COMPETISOFT [COMP06] llamada Perfil SOA (PSOA), que fue remitida a los integrantes del proyecto para su consideración. Dicha generalización permite confirmar la fácil integración de los conceptos más importantes que existen detrás de la definición de actividades y entregables realizada, con las adaptaciones necesarias, con otros procesos de desarrollo como el descrito en COMPETISOFT manteniendo la base del planteo que se realiza en la metodología SOA propuesta.

A la luz del surgimiento de otros enfoques metodológicos para desarrollo SOA durante el período de realización de este trabajo, y aunque no estaba previsto entre los objetivos del mismo, se presentó una breve descripción del plug-in para RUP por resultar el más interesante debido a la similitud del enfoque con el de la Extensión SOA, mencionando puntos de convergencia y divergencia con el planteado por la metodología SOA propuesta, que permite confirmar la adecuación de los aspectos más importantes que se plantean en la misma. Un análisis comparativo de resultados obtenidos con la utilización de ambas "extensiones" al RUP, si bien interesante, está fuera del alcance de este trabajo ya que no es posible contar con datos históricos suficientes dado lo reciente del surgimiento del plug-in y de la metodología SOA propuesta.

Las conclusiones presentadas sobre los objetivos planteados para este trabajo de tesis, permiten concluir que los mismos fueron cumplidos en su totalidad en forma satisfactoria, agregando incluso una última parte de comparación con otras metodologías surgidas recientemente, que no estaba prevista.

8.3. Trabajo a futuro

Como trabajo a futuro han quedado varias líneas abiertas que permiten la continuación de la investigación en el área, en primer lugar la evaluación de la experiencia con la metodología SOA propuesta del proyecto de grado "Enfoque SOA (Service Oriented Architecture) con integración de aplicaciones móviles para LIMS (Laboratory Information Management System)" [SOAM06] que se espera presente resultados interesantes sobre la aplicación de la metodología SOA con el enfoque de implementación SCA y SDO, además de la evaluación de la suite de herramientas de IBM para modelado del Negocio en forma gráfica y con BPEL y su ejecución en el motor de procesos que se provee.

Otra prueba interesante a realizar sobre la metodología SOA es una nueva aplicación de la metodología SOA pero ajustada y mejorada, que incluye más elementos que los definidos inicialmente, que se espera aporten a la mejora de la gestión de servicios en el proceso de desarrollo y de aspectos de la calidad del producto a obtener. En forma ambiciosa se podría plantear que

esta aplicación fuera realizada en alguna empresa de desarrollo de software real, que podría ser del medio o iberoamericana si se pudiera probar el Perfil SOA realizado para el modelo de procesos de COMPETISOFT [COMP06].

Se podría investigar también la integración de los conceptos en la metodología SOA propuesta con procesos ágiles como eXtreme Programing [BECK99], para lo cual sería necesario, como principales aspectos a destacar para evaluar, la relación entre la definición y modelización de procesos del Negocio con la escritura de historias realizada por el cliente, así como la relación entre la definición de la metáfora que guía el diseño y el desarrollo, a similitud de la definición de la Arquitectura de Software, con la identificación, definición y especificación de servicios.

Otro aspecto interesante a probar sería realizar el desarrollo de un mismo producto por más de un proyecto, donde uno aplique la metodología SOA ajustada y mejorada, y el otro aplique alguno de los enfoques metodológicos de reciente surgimiento mencionados. Esto podría dar elementos para la evaluación de resultados de la aplicación de ambos enfoques en el contexto de un proyecto con los mismos requerimientos, restricciones, tecnologías, etc. lo que aportaría a la identificación de las buenas prácticas en cada uno y a la comparación de la adecuación de los elementos divergentes que se plantean.

Otra línea abierta de trabajo interesante y que está siendo tratada por la comunidad del software en general, es la conjunción del enfoque SOA con los enfoques presentados en el Capítulo 4, Business Process Management (BPM) y Model Driven Architecture(MDA). Investigar la conjunción de la metodología SOA ajustada y mejorada con dichos enfoques, permitiría contar con un proceso definido que establezca en forma clara por un lado, las Actividades, Entregables, Roles, por otro lado las Fases y elementos de ejecución del desarrollo, y por otro la integración de herramientas de desarrollo a utilizar.

Para la conjunción con BPM, si bien en la metodología SOA propuesta existen actividades definidas que tienen en cuenta la utilización de ese enfoque, será necesario investigar herramientas que permitan realizar el modelado de los procesos del Negocio, tanto en forma gráfica para definir los diagramas de flujo y elementos asociados, como en BPEL para permitir la ejecución de procesos en un motor de ejecución de procesos. Existen herramientas que proveen esas tres características y se cuenta con una lista de herramientas de este estilo recomendadas por la OMG, pero cuya evaluación excedía el alcance definido para este trabajo.

Para el enfoque MDA se cuenta con una primera propuesta que también está basada en el proceso base adaptación del RUP, nombrada Extensión MDA, definida y probada por parte de un proyecto de grado realizado en el año 2006 bajo mi tutoría, que fue propuesto como parte de las investigaciones en el marco de la realización de este trabajo. Sería interesante investigar la oportunidad de conjuntar la Extensión MDA obtenida con la Extensión SOA para obtener la Extensión SOA-MDA.

Este trabajo implica revisar todas las actividades y elementos definidos en ambas extensiones, más los ajustes que éstas realizan en las Disciplinas, actividades, entregables y roles del proceso base, para definir las actividades y resto de los elementos para esta integración. Se deberá investigar la consistencia entre las actividades que se agregan, así como el uso de herramientas definidas para la Extensión MDA, y ver como se compatibilizan con el enfoque de la Extensión SOA, además de investigar otras herramientas que permitan generación de servicios y componentes asociados, o mejoras a las ya relevadas.

Otro aspecto a tener en cuenta en la conjunción de BPM-SOA-MDA es como armonizar los enfoques y el uso de herramientas que los soportan, para obtener una suite de trabajo integral para BPM-SOA-MDA, donde el ideal sería una herramienta BPM que permita el modelado de procesos del negocio en forma gráfica, la asignación de servicios a estos procesos, tanto a nivel del proceso entero en servicios centrados en procesos, como en las invocaciones a servicios en el flujo de los mismos, la definición de los componentes que implementarán estos servicios diseñados con enfoque SOA, y la generación por parte de una herramienta de MDA de los componentes definidos, que puedan ser invocados en la ejecución del flujo definido en BPEL por el motor de ejecución de procesos de la herramienta BPMS.

Capítulo 9

Anexo A - Sitio Web de la Extensión SOA

9.1. Introducción

Para la aplicación de la "Metodología de desarrollo para aplicaciones con enfoque SOA (Service Oriented Architecture)" en el curso "Proyecto de Ingeniería de Software" en el segundo semestre de 2005 se hizo necesario poner la información accesible para los estudiantes, de la misma forma en que se encuentra accesible la información del proceso de desarrollo base del curso, para lo cual se realizó el Sitio Web que se describe en este documento.

En el Sitio Web del curso se presenta una estructura en la cual en el esqueleto base se definen las Dimensiones del Tiempo y las Disciplinas, donde se muestran los elementos que son comunes a las distintas extensiones, y en cada extensión se presentan las modificaciones, agregados o eliminaciones de Disciplinas, Actividades, Entregables y Roles que corresponden a la ejecución de dicha extensión. El sitio original está basado en el Proyecto Web que Rational Unified Process (RUP) brindaba en el año 2000 en forma libre para adaptar al proceso de la Organización. Se puede acceder a los distintos contenidos del sitio mediante un TreeBrowser que es un applet que despliega una estructura de directorios similar a la mostrada en el Explorador de Windows, que fue adaptado como parte del trabajo realizado en [ADBP00].

Para la edición del curso 2005 en que se realiza la prueba de la Extensión SOA el proceso base definido se denomina "Modificable, Unificado y Medible (MUM)" sobre el cual se estaba trabajando por parte de un proyecto de grado para la mejora de la calidad y métricas del proceso. Se cuenta también con las extensiones Orientación a Objetos (OO) y Genexus (GX) definidas en ediciones anteriores, para desarrollo con dichos enfoques. La ejecución de la Extensión SOA consiste en la ejecución de los elementos definidos en el proceso base, más los definidos en la Extensión OO, más los definidos en la Extensión SOA. En la figura 9.1 se muestra la página inicial del curso "Proyecto de Ingeniería de Software" donde se puede apreciar la estructura de árbol mencionada para la edición 2005 del curso:

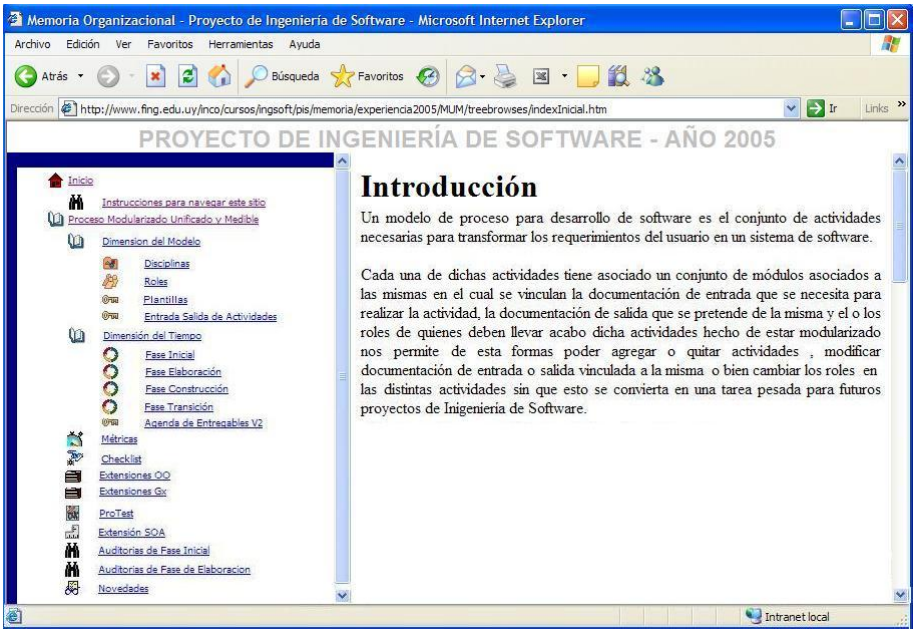


Figura 9.1: Página inicial del Sitio Web "Proyecto de Ingeniería de Software" edición 2005

9.2. Sitio Web de la Extensión SOA

Una vez en la Extensión SOA se pueden visualizar y navegar sus componentes en el primer nivel: Inicio, Extensión SOA y Bajar el Sitio. Dentro del nivel Extensión SOA se presentan la Introducción, Disciplinas, Roles, Plantillas SOA, Glosario y Material, desplegando abierto el nivel de las Disciplinas que constituyen el núcleo de la metodología: Modelado del Negocio, Diseño e Implementación. En la figura 9.2se muestra la página de presentación de la Extensión SOA:

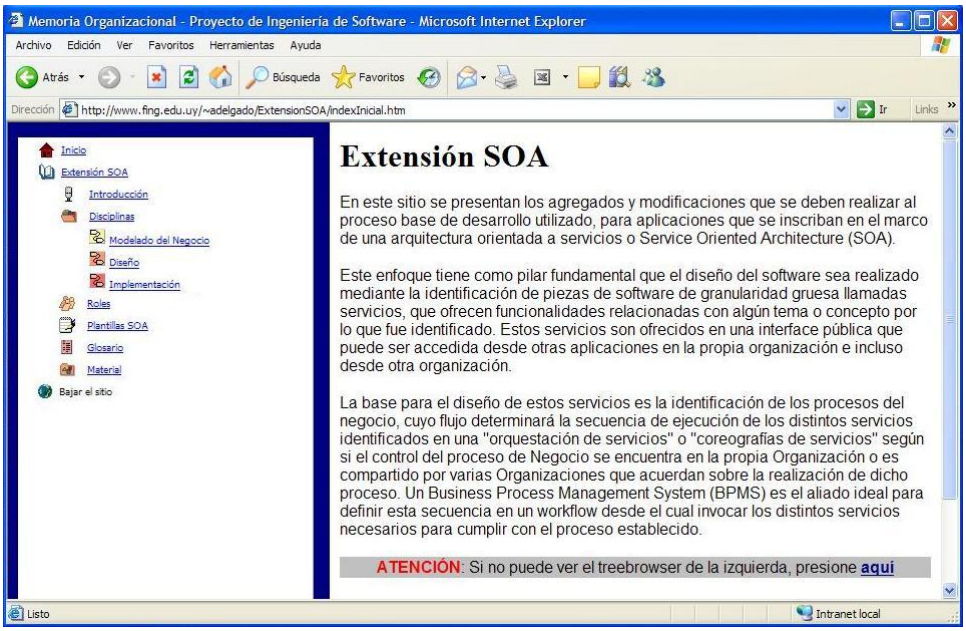


Figura 9.2: Página inicial del Sitio Web de la Extensión SOA edición 2005

9.2.1. Introducción

Al seguir el link del nivel de Introducción se muestra una página en la cual se brinda una definición global del enfoque y los principales conceptos asociados, que constituyen la base de la definición de la metodología SOA realizada. No se muestra la figura correspondiente a dicha página ya que las definiciones y conceptos que allí se presentan son los mismos que se detallaron en el capítulo 4.

9.2.2. Disciplina Modelado del Negocio

Al abrir el nivel de la Disciplina de Modelado del Negocio se presentan sobre la izquierda las actividades incluídas en esta disciplina, y sobre la derecha la descripción de la Disciplina con sus objetivos. Debajo de la descripción se brinda el link Entradas - Salidas - Roles Involucrados que lleva a una página donde se muestra dicha información para las actividades definidas en la Disciplina. En la figura 9.3 se muestra la página inicial correspondiente a esta disciplina:

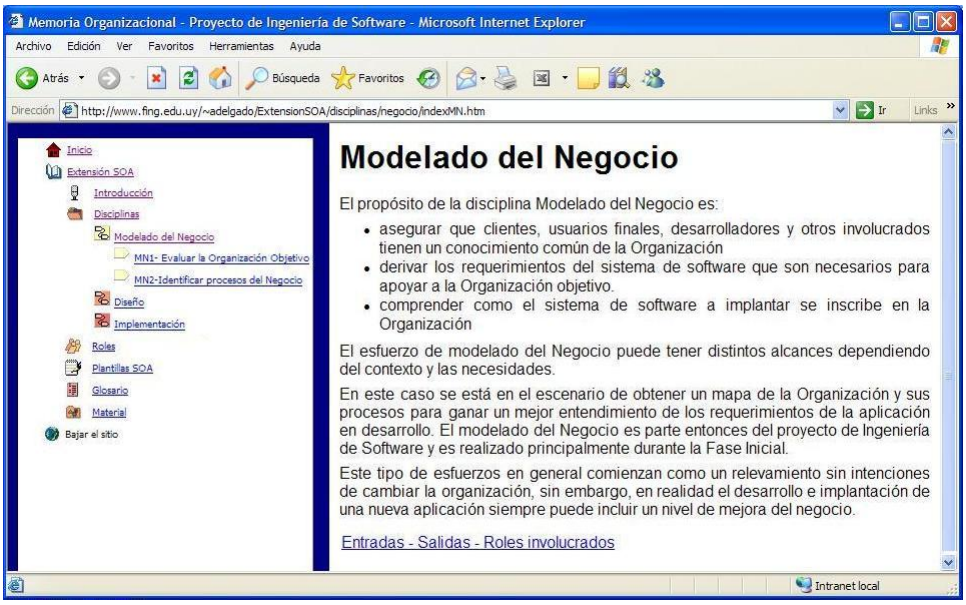


Figura 9.3: Página de la Disciplina Modelado del Negocio en la Extensión SOA

Al seguir el link de Entradas - Salidas - Roles involucrados, se presenta una página que tiene dos grillas, en la primera se muestra para cada una de las actividades definidas en la Disciplina, los entregables de entrada y salida asociados, el rol responsable de la realización de la actividad y los entregables, los roles que participan en cada actividad, y una indicación por semana de cuando debe ser realizada la misma. Esta información se presenta también en forma de link, por lo que las tablas son navegables, accediendo desde la misma a la definición de la actividad, a los entregables asociados y a la descripción de los roles involucrados. En la figura 9.4 se muestra página con la grilla correspondiente a la información de las actividades:

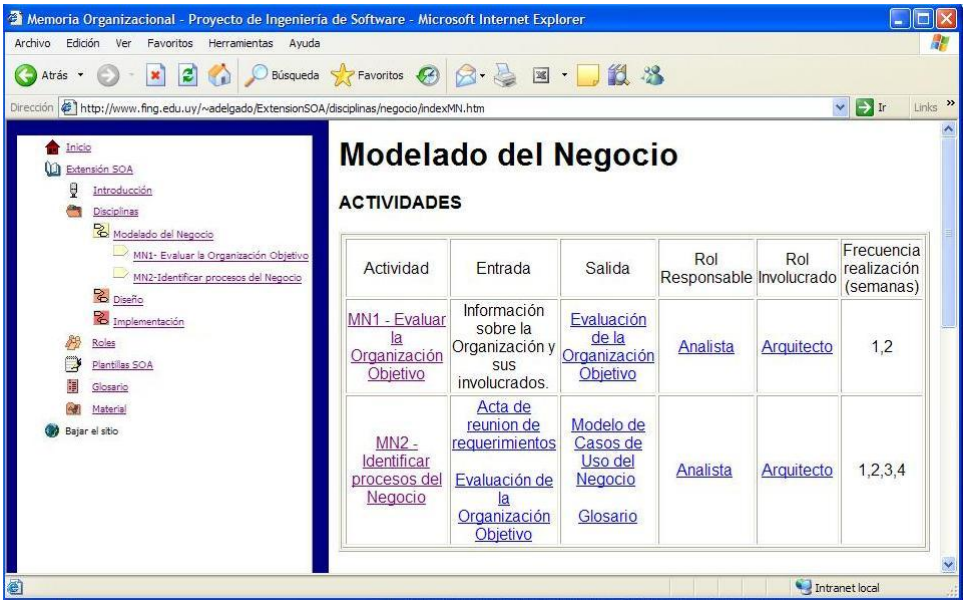


Figura 9.4: Página con grilla de Actividades para la Disciplina Modelado del Negocio

Bajando por el scroll en esa misma página se encuentra la grilla correspondiente a los entregables de la Disciplina, mostrando las actividades para las cuales es entrada y las actividades de las que es salida, más el rol responsable, roles involucrados y frecuencia de la entrega, la que se muestra en la figura 9.5:

Figura 9.5: Página con grilla de Entregables para la Disciplina Modelado del Negocio

La primer actividad en la Disciplina Modelado del Negocio corresponde a MN1 - Evaluar la Organización Objetivo, que se realiza para obtener información sobre el contexto del Negocio en que se realiza el desarrollo. Al seleccionarla se presentan en el marco de la derecha, los objetivos y la descripción correspondientes a la actividad, así como el link a Entradas - Salidas - Roles involucrados correspondientes a la misma. En la figura 9.6 se muestra la página correspondiente a dicha actividad:

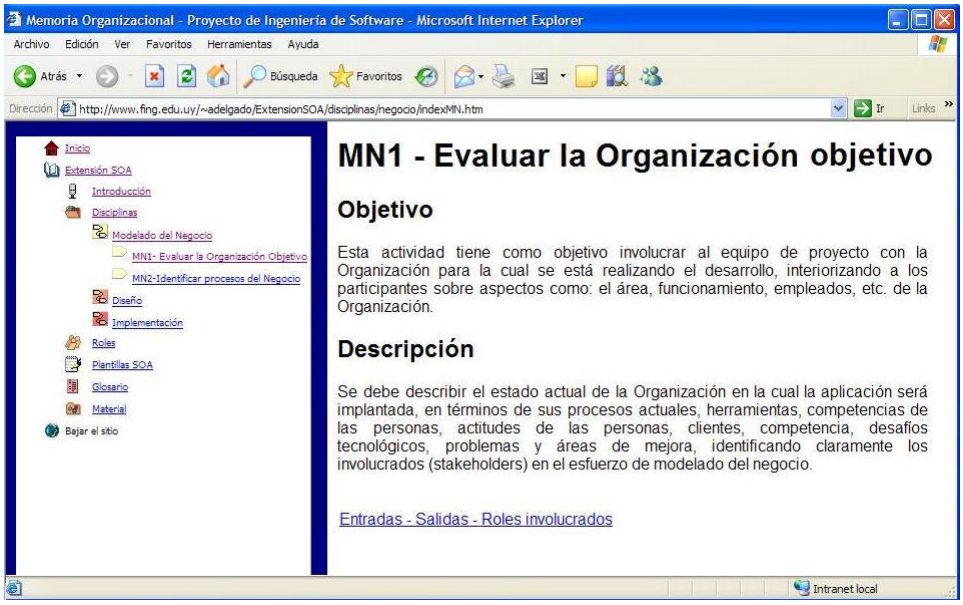


Figura 9.6: Página de la actividad MN1 - Evaluar la Organización Objetivo

Al seguir el link correspondiente a Entradas - Salidas - Roles involucrados se muestra la grilla de la misma forma que la anterior pero solamente para esta actividad, con los links asociados para navegar los elementos que la componen. En la figura 9.7 se muestra la página correspondiente a la grilla:

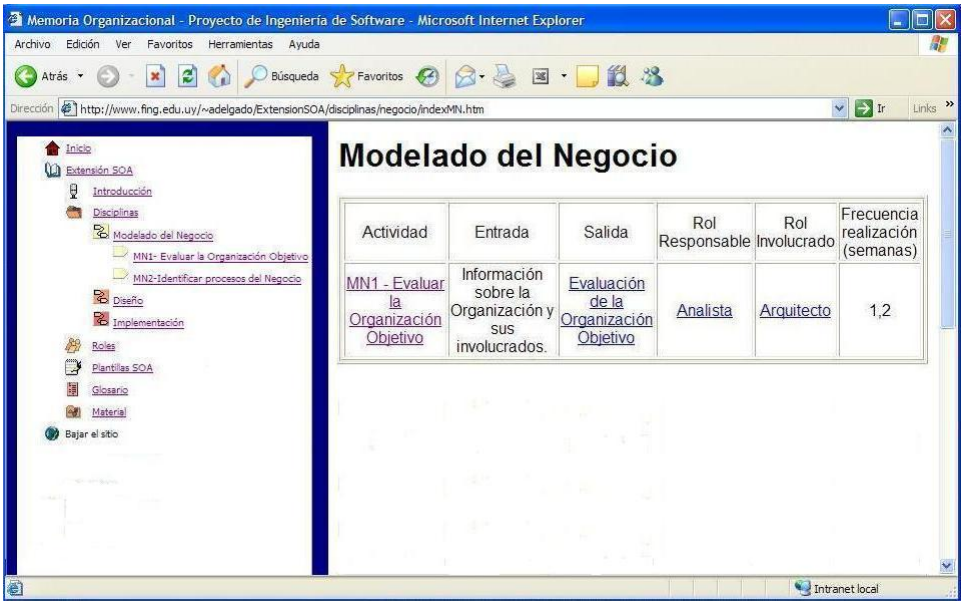


Figura 9.7: Página con la grilla correspondiente a la actividad MN1 - Evaluar la Organización Objetivo

La siguiente actividad en la Disciplina Modelado del Negocio corresponde a MN2 - Identificar los procesos del Negocio, que se realiza para identificar los procesos que se siguen en la Organización y describirlos como Casos de Uso del Negocio. Al seleccionarla se presentan en el marco de la derecha, los objetivos y la descripción correspondientes a la actividad, así como el link a Entradas

- Salidas - Roles involucrados correspondientes a la misma. En la figura 9.8 se muestra la página correspondiente a dicha actividad:

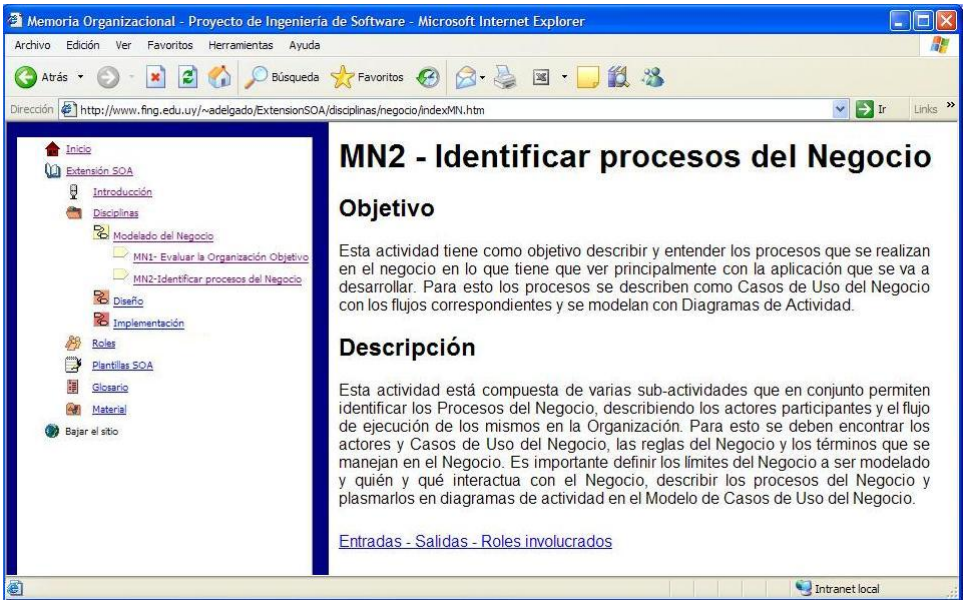


Figura 9.8: Página de la actividad MN2 - Identificar los procesos del Negocio

Al seguir el link de Entradas - Salidas - Roles involucrados, se muestra una página que contiene la grilla de entregables entrada y salida, roles involucrados y semanas en que se realiza dicha actividad, que no se presenta para esta actividad ni para el resto de las actividades, ya que es del mismo tipo que la grilla presentada en la figura 9.7 específica para actividad que se está describiendo.

9.2.3. Disciplina Diseño

Al abrir el nivel de la Disciplina Diseño se presentan sobre la izquierda las actividades incluídas en esta disciplina, y sobre la derecha la descripción de la Disciplina con sus objetivos, y un link de Entradas - Salidas - Roles Involucrados que lleva a una página donde se muestra dicha información para las actividades de la Disciplina. En la figura 9.9 se muestra la página inicial correspondiente a esta disciplina:

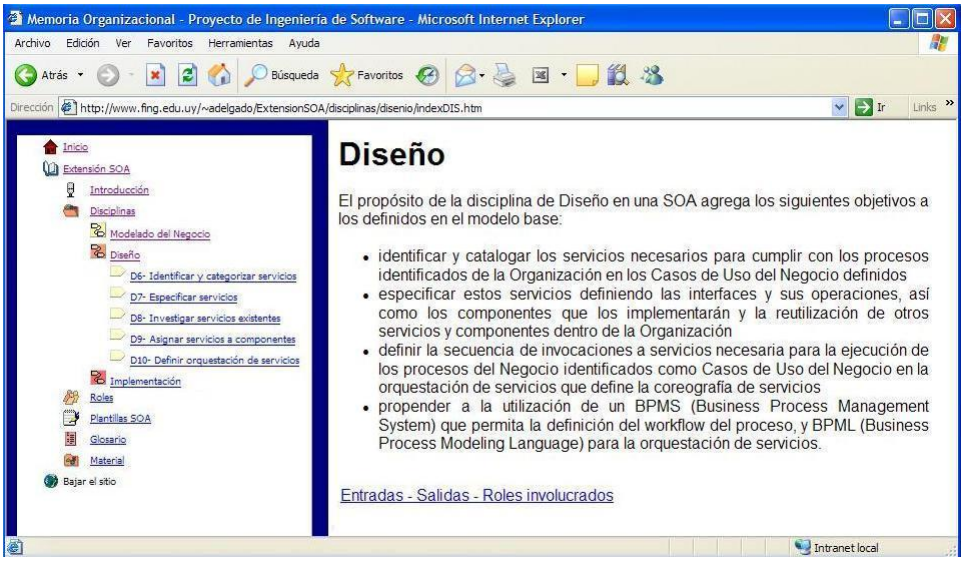


Figura 9.9: Página de la Disciplina Diseño en la Extensión SOA

Al seguir el link de Entradas - Salidas - Roles involucrados, se presenta una página con dos grillas, la primera con los entregables de entrada y salida para cada actividad definida, el rol responsable de la actividad, los entregables y roles que participan en cada una, y una indicación por semana de cuando debe realizarse. En la figura 9.10 se muestra la página con la grilla correspondiente a la información de las actividades:

Figura 9.10: Página con grilla de Actividades para la Disciplina Diseño

Bajando por el scroll en esa misma página se encuentra la grilla correspondiente a los entregables de la Disciplina, mostrando las actividades para las cuales es entrada y las actividades de las que es salida, más el rol responsable, roles involucrados y frecuencia de la entrega, la que se muestra en la figura 9.11:

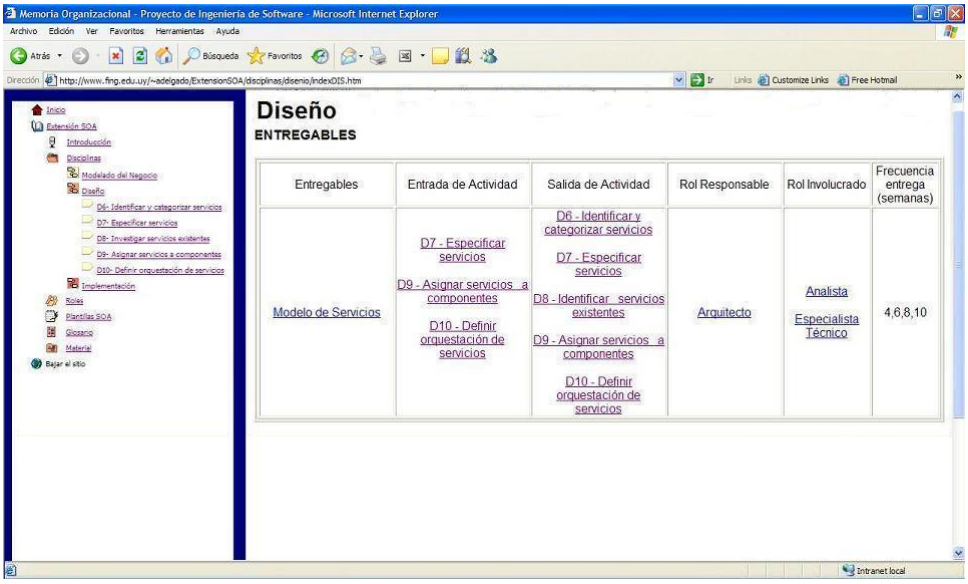


Figura 9.11: Página con grilla de Entregables para la Disciplina Diseño

La primer actividad en la Disciplina Diseño es D6 - Identificar y categorizar servicios, cuyo objetivo es realizar la definición de los servicios que compondrán el software, clasificandolos para su mejor comprensión. Al seleccionarla se presentan en el marco de la derecha, los objetivos y la descripción correspondientes a la actividad, y el link a Entradas - Salidas - Roles involucrados correspondientes. En la figura 9.12 se muestra la página correspondiente a dicha actividad:

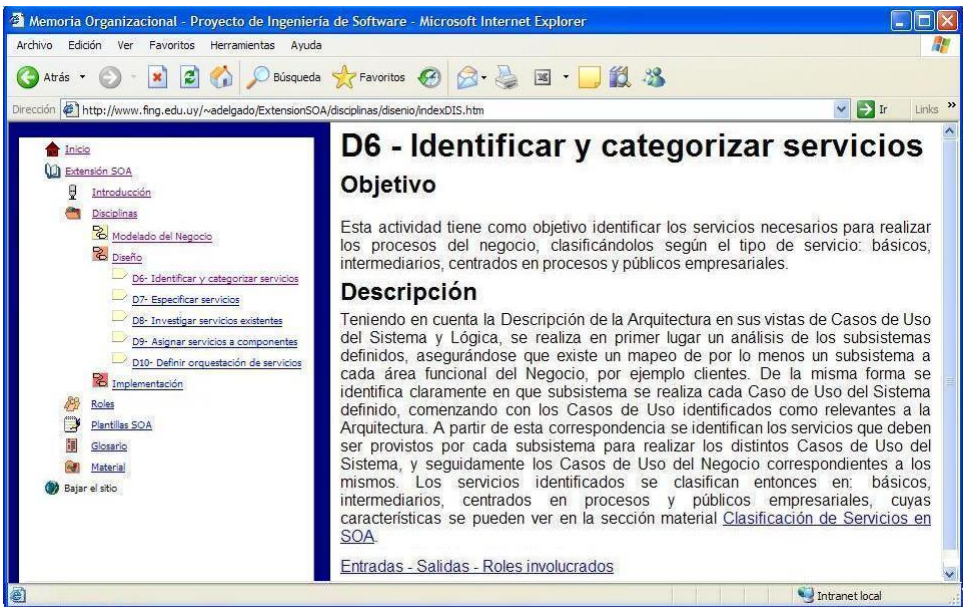


Figura 9.12: Página de la actividad D6- Identificar y categorizar servicios

La siguiente actividad en la Disciplina Diseño corresponde a D7 - Especificar servicios, cuyo objetivo es realizar la especificación de los servicios definidos, identificando los contratos de servicio para cada uno, incluyendo interfaces brindadas, operaciones, parámetros, entre otros. Al seleccionarla se presentan en el marco de la derecha, los objetivos y la descripción correspondientes a la actividad, y el link a Entradas - Salidas - Roles involucrados correspondientes. En la figura 9.13 se muestra la página correspondiente a dicha actividad:

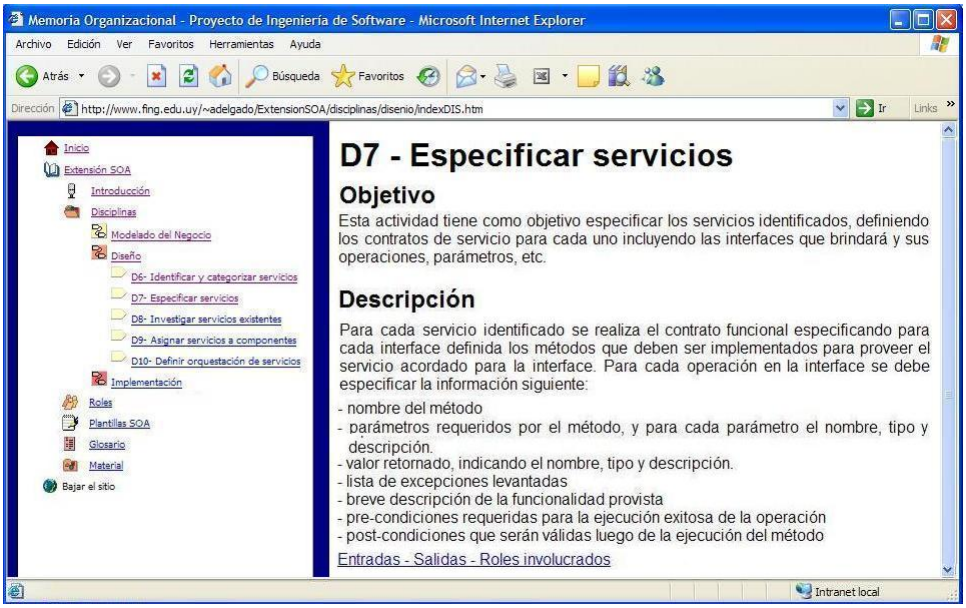


Figura 9.13: Página de la actividad D7 - Especificar servicios

La siguiente actividad D8 - Investigar servicios existentes, tiene como objetivo encontrar servicios que ya estén implementados en la Organización para ser reutilizados. Al seleccionarla se presentan en el marco de la derecha, los objetivos y la descripción correspondientes a la actividad, y el link a Entradas - Salidas - Roles involucrados correspondientes. En la figura 9.14 se muestra la página correspondiente a dicha actividad:

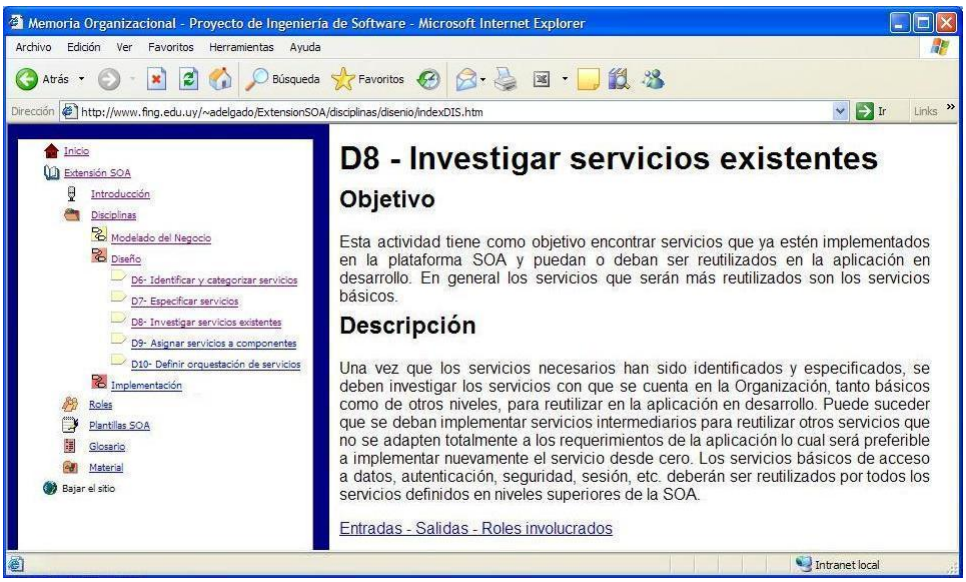


Figura 9.14: Página de la actividad D8 - Investigar servicios existentes

La siguiente actividad D9 - Asignar servicios a componentes, tiene como objetivo definir los componentes que serán implementados para proveer los servicios especificados. Al seleccionarla se presentan en el marco de la derecha, los objetivos y la descripción de la actividad, y el link a Entradas - Salidas - Roles involucrados correspondientes. En la figura 9.15 se muestra la página de dicha actividad:

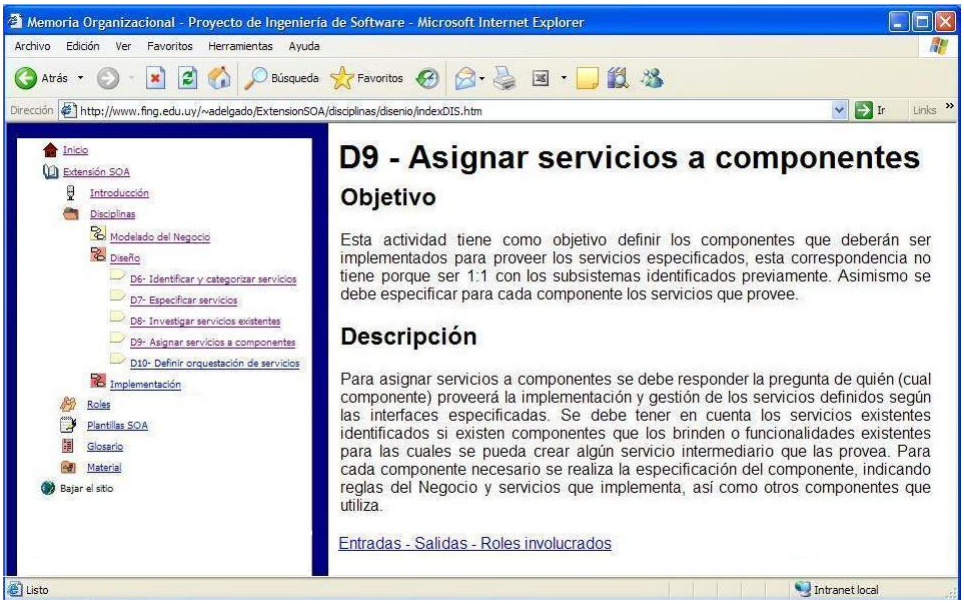


Figura 9.15: Página de la actividad D9 - Asignar servicios a componentes

La siguiente actividad es D10 - Definir orquestación de servicios, cuyo objetivo es definir la secuencia de invocación a servicios definidos y existentes, necesaria para realizar los procesos del Negocio identificados. Al seleccionarla se presentan en el marco de la derecha, los objetivos y la descripción de la actividad, y el link a Entradas - Salidas - Roles involucrados correspondientes. En la figura 9.16 se muestra la página de dicha actividad:

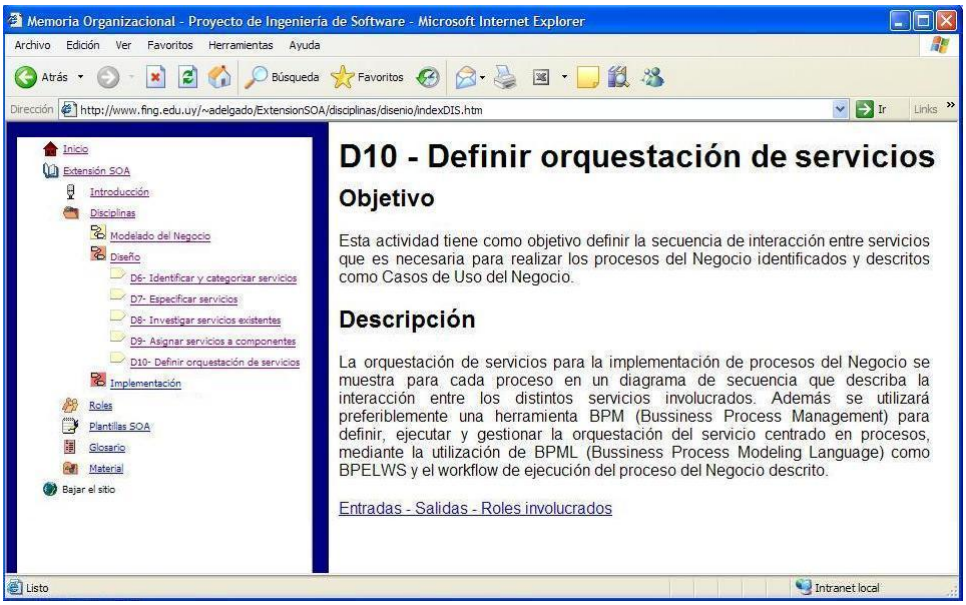


Figura 9.16: Página de la actividad D10 - Definir orquestación de servicios

9.2.4. Disciplina Implementación

Al abrir el nivel de la Disciplina Implementación se presentan sobre la izquierda las actividades incluídas en esta disciplina, y sobre la derecha la descripción, sus objetivos y el link Entradas - Salidas - Roles Involucrados que lleva a una página donde se muestra dicha información para las actividades definidas. En la figura 9.17 se muestra la página inicial de la disciplina:

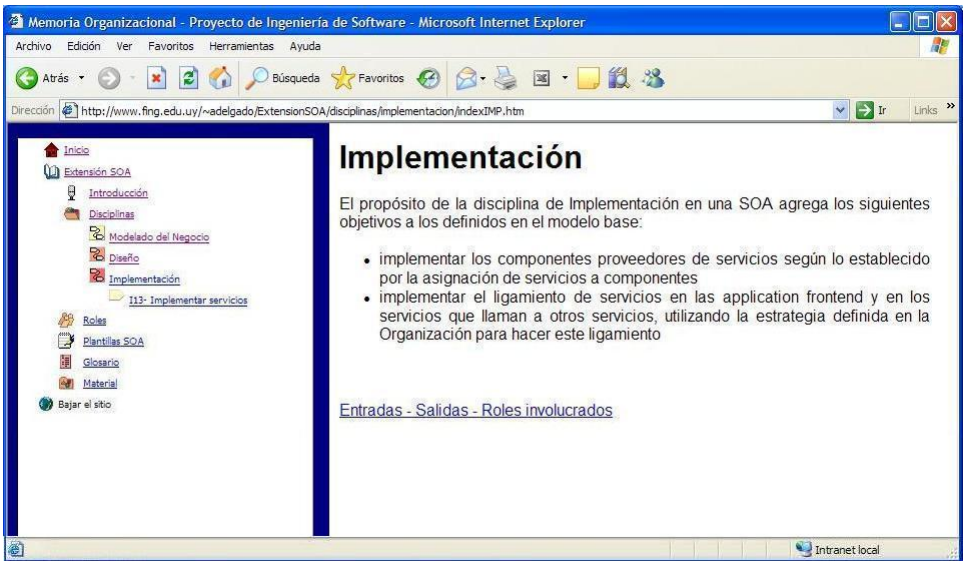


Figura 9.17: Página de la Disciplina Implementación en la Extensión SOA

Al seguir el link de Entradas - Salidas - Roles involucrados, se presenta una página que tiene dos grillas, en la primera se muestra para cada una de las actividades, los entregables de entrada y salida asociados, el rol responsable de la realización de la actividad, los entregables y roles que participan en una, y una indicación por semana de cuando debe realizarse. En la segunda grilla se muestran los entregables indicando las actividades de las que es entrada y salida, el rol responsable, roles involucrados y frecuencia de la entrega. En la figura 9.18 se muestra la página con las grillas de la información sobre actividades y entregables:

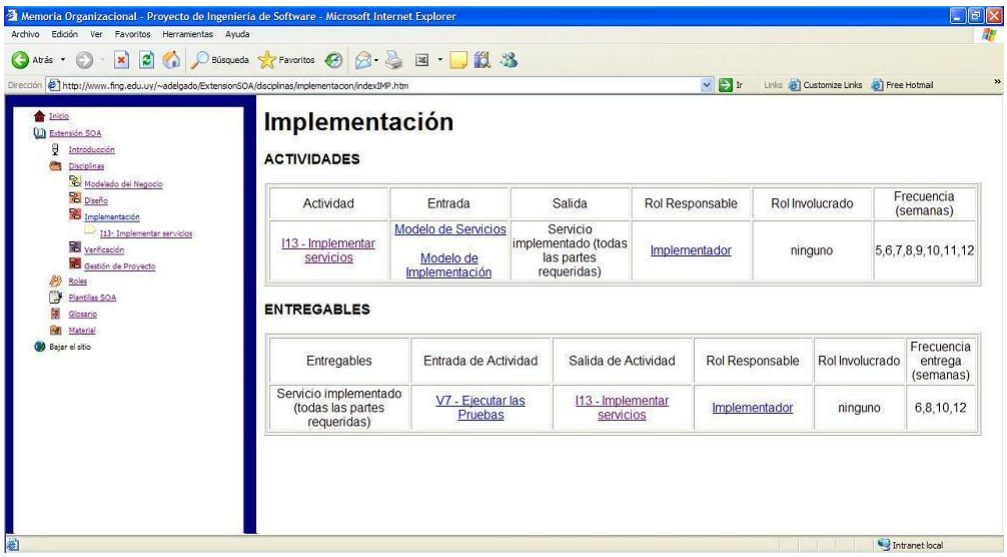


Figura 9.18: Página con grilla de Actividades y Entregables para la Disciplina Implementación

9.2.5. Roles

Siguiendo el link de roles se muestra una página en la que se indica que los roles del proceso base no se modifican en la Extensión SOA, siendo los principales involucrados en las actividades y entregables planteados los roles de Arquitecto, Analistas, Especialistas Técnicos e Implementadores, quienes tienen participación y responsabilidad en los mismos.

9.2.6. Plantillas

El link de plantillas presenta una página en la que se presentan las plantillas definidas para los entregables planteados, cada una se muestra con un link para acceder a su visualización o bajar el archivo correspondiente, que se brinda como .dot para ser utilizado en procesadores de texto como Word u Open Office, definiendo las secciones a realizar así como brindando guía del contenido para cada una. En la figura 9.19 se muestra la página correspondiente a las plantillas de la Extensión SOA.

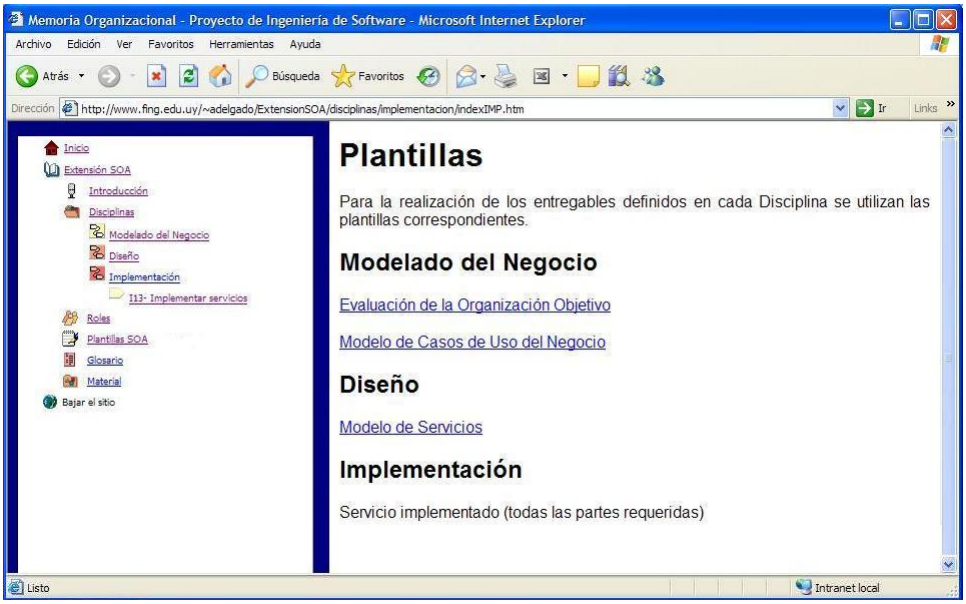


Figura 9.19: Página de acceso a las plantillas de la Extensión SOA

Los documentos correspondientes a las plantillas se adjuntan en el Capítulo 10 - Anexo B.

9.2.7. Glosario

Siguiendo el link de glosario se presenta una página que contiene las definiciones de los términos más importantes involucrados en la Extensión SOA y el enfoque de desarrollo, que son utilizados en toda la metodología SOA planteada. Un glosario sirve para sentar acuerdos sobre los términos utilizados entre todos los involucrados, lo que ayuda a la comprensión y la comunicación de los conceptos asociados. En la figura se muestran los términos definidos en la Extensión SOA.

9.2.8. Material

El link de material presenta una página en la cual se muestra la principal bibliografía utilizada para la definición de la Extensión SOA y recomendaciones para profundizar sobre los diversos

aspectos del enfoque. La bibliografía disponible en internet se presenta con un link para poder accederla directamente, la bibliografía correspondiente a libros se muestra con datos asociados como ISBN y año de edición, para facilitar su búsqueda. En la figura 9.20 se muestra la página asociada al material.

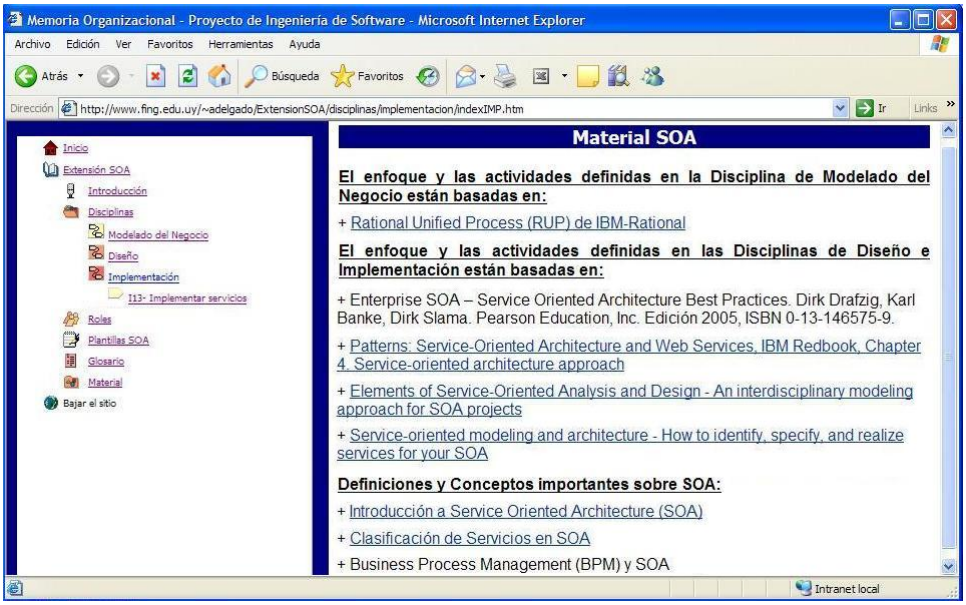


Figura 9.20: Página de acceso al material en la Extensión SOA

9.2.9. Bajar el sitio

El link Bajar el sitio permite bajar un archivo .zip que contiene el sitio presentado, para su instalación en cualquier máquina, de forma de permitir la navegación off line de la Extensión SOA para su consulta. Al presionar el link aparece el diálogo para bajar el archivo, por lo que no se muestra ninguna página asociada a dicho link.

Capítulo 10

Anexo B - Plantillas de entregables de la Extensión SOA

10.1. Introducción

Las plantillas de entregables son parte fundamental de la comprensión de los entregables a realizar al aplicar la metodología SOA propuesta. Además de la descripción de las actividades definidas, proveer una guía para la realización de los entregables que se deben obtener como salida de las mismas resulta un elemento de apoyo importante para su comprensión. A continuación se presentan los entregables definidos para las actividades y Disciplinas correspondientes; en la sección Disciplina Modelado del Negocio y sección Disciplina Diseño.

10.2. Plantillas de la Disciplina Modelado del Negocio

En esta sección se presentan las plantillas definidas para los entregables de la Disciplina Modelado del Negocio, que se obtienen como salida de las actividades MN1 - Evaluar la Organización Objetivo y MN2 - Identificar los procesos del Negocio.

10.2.1. Plantilla del entregable Evaluación de la Organización Objetivo

[Nombre del proyecto]

Evaluación de la Organización Objetivo

Versión [1.0]

[Este documento es la plantilla base para elaborar el documento Modelo de Casos de Uso. Los textos que aparecen entre paréntesis rectos son explicaciones de que debe contener cada sección. Dichos textos se deben seleccionar y sustituir por el contenido que corresponda. Para actualizar la tabla de Contenido, haga clic con el botón derecho del ratón sobre cualquier línea del contenido de la misma y seleccione Actualizar campos, en el cuadro que aparece seleccione Actualizar toda la tabla y haga clic en el botón Aceptar.]

[Este documento se realiza para describir el estado actual de la Organización en la cual la aplicación será implantada, en términos de sus procesos del Negocio, herramientas, competencias de las personas y actitudes, clientes, competidores, tendencias tecnológicas, problemas y áreas de mejora posibles. Es importante también identificar quienes serán las personas representantes del cliente para llevar adelante el esfuerzo de Modelado del Negocio.

No es el propósito de este documento describir en profundidad la Organización en forma completa y detallada, sino lo suficiente para permitir al equipo priorizar el trabajo en las partes más importantes de la Organización afectadas por el desarrollo actual. Sin embargo, las razones del Modelado del Negocio deben guiar la amplitud y profundidad de esta evaluación, teniendo siempre como objetivo las áreas afectadas por el desarrollo actual.]

Historia de revisiones

Fecha	Versión	Descripción	Autor
[dd/mm/aaaa]	[x.x]	[detalles]	[nombre]

INDICE

1 Contexto del Negocio

[En esta sección se debe proveer una breve introducción del dominio del Negocio en que la Organización trabaja.]

2 Factores externos

[En esta sección se describen los factores externos que influyen en la Organización.]

2.1 Clientes

[Aquí se presenta una lista de los clientes de la Organización y sus expectativas sobre los productos. Se debería incluir un resumen de las investigaciones realizadas para comprender la demanda de los clientes en el Negocio.]

2.2 Competidores

[Aquí se presenta una lista de los competidores, si los hay, en el Negocio de la Organización.]

2.3 Otros involucrados

[Aquí se presenta una lista de otros involucrados en el Negocio de la Organización, como proveedores y socios de Negocio.]

3 Factores internos

[En esta sección se presentan los factores internos que inciden en la Organización y su Negocio.]

3.1 Procesos del Negocio

[Aquí se presenta una breve descripción de los procesos del Negocio actuales, principalmente los que están relacionados con el desarrollo a realizar y otros con los cuales existan importantes interrelaciones a tener en cuenta.]

3.2 Herramientas de soporte

[Aquí se presenta una lista de las herramientas de soporte que son actualmente utilizadas en la Organización.]

3.3 Organización interna

[Aquí se describe brevemente la organización interna de la Organización, que roles y equipos tiene actualmente y cual es su orden jerárquico.]

3.4 Competencias, habilidades y actitudes

[Aquí se presenta un inventario de las competencias, habilidades y actitudes de los individuos en la Organización.]

3.5 Capacidad de cambio en la Organización

[Aquí se presenta una lista de la capacidad de cambio percibida en la Organización.]

4 Conclusiones de la evaluación

[En esta sección se presentan las conclusiones de la evaluación realizada sobre la Organización Objetivo, presentando una lista de las áreas de mayores problemas y oportunidades, sin importar la categoría a la que pertenecen.]

4.1 Áreas problemáticas

[Aquí se presenta un resumen del análisis de las actividades en los procesos del Negocio existentes actualmente.]

4.2 Nuevas tecnologías aplicables

[Aquí se presenta un resumen de las soluciones disponibles que son aplicables a la Organización.]

10.2.2. Plantilla del entregable Modelo de Casos de Uso del Negocio

[Nombre del proyecto]

Modelo de Casos de Uso del Negocio

Versión [1.0]

[Este documento es la plantilla base para elaborar el documento Modelo de Casos de Uso. Los textos que aparecen entre paréntesis rectos son explicaciones de que debe contener cada sección. Dichos textos se deben seleccionar y sustituir por el contenido que corresponda. Para actualizar la tabla de Contenido, haga clic con el botón derecho del ratón sobre cualquier línea del contenido de la misma y seleccione Actualizar campos, en el cuadro que aparece seleccione Actualizar toda la tabla y haga clic en el botón Aceptar.]

[Este documento se realiza en forma previa al Modelo de Casos de Uso del Sistema y constituye una entrada fundamental para realizar el mismo. El propósito principal de modelar actores y CU del Negocio es describir como se utiliza el negocio por parte de sus clientes y socios. Estos Casos de Uso del Negocio constituyen los procesos del Negocio que dan valor a los actores involucrados. Estos procesos son el vehículo con el que el Negocio hace cosas. Las estrategias del Negocio se traducen en Objetivos del Negocio que guían las actividades realizadas en los procesos del Negocio, por lo que cada CU del Negocio debe soportar por lo menos un objetivo del Negocio, y esta es una guía para identificar la granularidad correcta para la definición de los CU del Negocio.

Cuando se está realizando el modelado del Negocio solo para tener una visión primaria para un proyecto de Ingeniería de Software, se debe delimitar con cuidado el esfuerzo de modelado del Negocio. En este caso en general no vale la pena hacer el modelado del Negocio entero, y es conveniente considerar la parte del Negocio con la que se está tratando, como si fuera el Negocio entero, la parte considerada será la que utilice directamente el sistema en desarrollo. Las partes de la Organización que se dejen fuera de los límites del Negocio tomado en cuenta, pueden ser modeladas como actores del Negocio.]

Historia de revisiones

Fecha	Versión	Descripción	Autor
[dd/mm/aaaa]	[x.x]	[detalles]	[nombre]

INDICE

1 Actores del Negocio

[En esta sección se debe describir a cada uno de los actores del Negocio que participan en los Casos de Uso del Negocio, un actor del Negocio es un usuario del Negocio o cualquier entidad que interactúa con el Negocio.]

1.1 [Actor del Negocio 1]

[Descripción del Actor 1 del Negocio]

1.2 [Actor del Negocio 2]

[Descripción del Actor 2 del Negocio]

2 Casos de Uso del Negocio

2.1 Diagrama de Casos De Uso del Negocio

[Aquí se presenta el Diagrama de Casos de Uso del Negocio, para mostrar la interacción entre los Actores y los Casos de Uso del Negocio definidos según los procesos del negocio identificados.]

2.2 [Caso de Uso del Negocio 1]

2.2.1 Descripción

[Explicar brevemente el propósito del caso de uso del Negocio]

2.2.2 Flujo de eventos principal

[En esta sección se realiza una descripción textual del flujo del Negocio que representa el Caso de Uso. El flujo debe describir que hace el Negocio para proveer de valor a un actor del Negocio, pero no como hace el Negocio para resolver sus problemas. Las alternativas simples se pueden presentar dentro del texto del Caso de Uso. Si el flujo alternativo es más complejo, use una sección separada para describirlo.]

2.2.3 Flujos de eventos alternativos

[En esta sección se describen las alternativas más complejas a las cuales se hacía referencia en el Flujo de eventos principal. Estos flujos alternativos representan la conducta alternativa normalmente debida a excepciones que ocurren en el flujo principal. Ellos pueden ser necesarios para describir los eventos asociados con la conducta alternativa. Cuando finaliza el flujo alternativo, se retoman los eventos del flujo de eventos principal a menos que se establezca otra cosa.]

2.2.3.1. [Flujo alternativo 1]

[Los flujos alternativos pueden dividirse en subsecciones si esto aporta claridad]

[Sub-flujo alternativo 1]

[Sub-flujo alternativo 2]

...

2.2.3.2. [Flujo alternativo 2]

...

2.2.4 Diagrama de Actividad

[En esta sección incluye un Diagrama de Actividad que modele en forma gráfica los flujos del Caso de Uso del Negocio descritos previamente. Es importante que se muestren los distintos condicionales que existen en el proceso que determinan que se siga un camino u otro en la ejecución del proceso, y cuales son todas las opciones de terminación para los flujos modelados.]

2.2.5 Requerimientos especiales

[En esta sección se especifican los requerimientos especiales que son específicos a un caso de uso del Negocio, que son las características del caso de uso del Negocio no cubiertas por los flujos descritos.]

[Requerimiento especial 1]

...

2.3 [Caso de Uso del Negocio 2]

...

3 Reglas del Negocio

[En esta sección se especifican las Reglas del Negocio que se deben tener en cuenta en la ejecución de los distintos procesos definidos. Las Reglas del Negocio son tipos de requerimientos sobre como el Negocio, incluyendo sus herramientas, debe funcionar.

Pueden ser leyes y regulaciones impuestas al Negocio entre otras, y pueden ser clasificadas de distinta forma según el Negocio, por ejemplo pueden estar agrupadas por dominio, usuario o grupo de productos, aunque una clasificación más formal las define como Restricciones y Derivaciones.

Las Reglas del Negocio deben ser expresadas con un nivel de formalismo que permita su automatización, una forma puede ser utilizando el Object Constraint Language (OCL) especificado en UML. De todas formas es útil mostrar las Reglas de Negocio como notas de texto en los elementos de los distintos diagramas y modelos, por ejemplo en los Diagramas de Actividad.]

3.1 [Regla del Negocio 1]

...

10.3. Plantillas de la Disciplina Diseño

En esta sección se presenta la plantilla definida para el entregable Modelo de Servicios de la Disciplina Diseño, que se obtiene como salida de las actividades D6 - Identificar y categorizar servicios, D7 - Especificar servicios, D8 - Investigar servicios existentes, D9 - Asignar servicios a componentes y D10 - Definir orquestación de servicios.

10.3.1. Plantilla del entregable Modelo de Servicios

[Nombre del proyecto]

Modelo de Servicios

Versión [1.0]

[Este documento es la plantilla base para elaborar el documento Modelo de Servicios. Los textos que aparecen entre paréntesis rectos son explicaciones de que debe contener cada sección. Dichos textos se deben seleccionar y sustituir por el contenido que corresponda. Para actualizar la tabla de Contenido, haga clic con el botón derecho del ratón sobre cualquier línea del contenido de la misma y seleccione Actualizar campos, en el cuadro que aparece seleccione Actualizar toda la tabla y haga clic en el botón Aceptar.]

[Este documento se realiza en forma coordinada con el Documento de Arquitectura de Software (SAD) en el cual además se incluye la Vista del Modelo de Servicios con los servicios más importantes definidos a partir de los Casos de Uso relevantes a la Arquitectura identificados. El modelado de servicios es el pilar del enfoque SOA donde se diseña el sistema como un conjunto de servicios que proveen las funcionalidades requeridas. La identificación y categorización de servicios permiten definirlos y ubicarlos conceptualmente según su propósito principal. La especificación de servicios permite determinar claramente las operaciones que proveen, junto a los parámetros y excepciones que manejan, y las pre y post condiciones que aplican en su ejecución.

La investigación de servicios existentes en la Organización permite reutilizar conocimiento, diseño e implementación en la misma, así como contribuir al crecimiento de servicios comunes entre aplicaciones de distintas secciones de la Organización. La orquestación de servicios define la secuencia de invocaciones a servicios definidos necesaria para realizar los procesos del Negocio que fueron especificados como Casos de Uso del Negocio, cuando el control del proceso corresponde a la Organización. En el caso de coordinación de procesos entre Organizaciones distintas, esta secuencia definida de invocaciones se conoce como coreografía y el control del proceso se encuentra repartido entre las Organizaciones participantes.]

Historia de revisiones

Fecha	Versión	Descripción	Autor
[dd/mm/aaaa]	[x.x]	[detalles]	[nombre]

INDICE

1 Identificación y categorización de servicios

[En esta sección se listan los servicios identificados indicando el tipo de servicio correspondiente. Los servicios pueden clasificarse como básicos, intermediarios, centrados en procesos y públicos empresariales. Se debe describir para cada servicio, en lenguaje natural de forma clara y sin ambigüedades, la justificación de porqué se define ese servicio, que funcionalidades provee indicando a que Caso de Uso corresponden, y que tipo de servicio es, indicando también que subsistema debe proveer este servicio.]

1.1 [Servicio1]

1.1.1 Funcionalidades provistas

[Se describe en lenguaje natural las funcionalidades generales provistas por el servicio1 identificado.]

1.1.2 Categorización

[Se indica que tipo de servicio es el servicio1 identificado, de acuerdo a la clasificación de servicios indicada en el documento GuíaServicios.pdf provisto en la sección Material.]

1.1.3 Caso/s de Uso asociado/s

[Se identifican los Casos de Uso que definen las funcionalidades que provee el servicio1 identificado.]

1.1.4 Subsistema de Diseño asociado

[Se identifica el Subsistema de Diseño que provee el servicio1 identificado.]

1.2 [Servicio2]

.....

2 Especificación de servicios

[Para cada servicio identificado se realiza su contrato funcional, especificando para cada interface definida los métodos que deben ser implementados para proveer el servicio acordado para la interface.]

2.1 [Servicio1]

2.1.1 Contrato funcional

[Para cada operación definida en la interface que compone el contrato funcional, se debe especificar la información que se indica a continuación.]

2.1.1.1. [Método1]

Método	[nombre del método]		
Parámetros	[nombre del parámetro]	[tipo del parámetro]	[descripción del parámetro]
Valor de Retorno	[nombre del valor retornado]	[tipo del valor retornado]	[descripción del valor retornado]
Excepciones	[Excepción 1 que levanta el método]		
	[Excepción 2 que levanta el método]		
Descripción	[Breve descripción en lenguaje natural de lo que realiza el método]		
Precondiciones	[Breve descripción en lenguaje natural de lo que se asume al invocar al método]		
Postcondiciones	[Breve descripción en lenguaje natural de lo que se asume una vez ejecutado el método]		

2.1.1.2. [Método2]

....

2.2 [Servicio 2]

...

3 Servicios existentes

[En esta sección se listan los servicios existentes en la Organización identificados, tanto básicos como de otros niveles, que sirvan para reutilizar en la aplicación en desarrollo. No todos los servicios existentes en la Organización serán reutilizados en la aplicación en desarrollo, los que no se usen no se listarán en esta sección. Puede suceder que se deban implementar servicios intermediarios para reutilizar otros servicios que no se adapten totalmente a los requerimientos de la aplicación lo cual será preferible a implementar nuevamente el servicio desde cero. Los servicios básicos de acceso a datos, autenticación, seguridad, sesión, etc. deberán ser reutilizados por todos los servicios definidos en niveles superiores de la SOA.]

3.1 [Servicio1]

3.1.1 Funcionalidades provistas

[Se describe en lenguaje natural las funcionalidades generales provistas por el servicio1 existente identificado.]

3.1.2 Categorización

[Se indica que tipo de servicio es el servicio1 existente identificado, de acuerdo a la clasificación de servicios indicada en el documento GuíaServicios.pdf provisto en la sección Material.]

3.1.3 Contrato funcional

[Para cada operación definida en la interface que compone el contrato funcional, se debe especificar la información que se indica a continuación.]

3.1.3.1. [Método1]

Método	[nombre del método]		
Parámetros	[nombre del parámetro]	[tipo del parámetro]	[descripción del parámetro]
Valor de Retorno	[nombre del valor retornado]	[tipo del valor retornado]	[descripción del valor retornado]
Excepciones	[Excepción 1 que levanta el método]		
	[Excepción 2 que levanta el método]		
Descripción	[Breve descripción en lenguaje natural de lo que realiza el método]		
Precondiciones	[Breve descripción en lenguaje natural de lo que se asume al invocar al método]		
Postcondiciones	[Breve descripción en lenguaje natural de lo que se asume una vez ejecutado el método]		

3.1.3.2. [Método2]

....

3.2 [Servicio2]

....

Capítulo 11

Anexo C - Perfil SOA (PSOA) para COMPETISOFT

11.1. Introducción

En este capítulo se presentan los aspectos más importantes de la generalización realizada de la metodología SOA propuesta, como Perfil SOA (PSOA) para COMPETISOFT. El agregado y/o modificación de actividades se realiza sobre el proceso de Desarrollo perteneciente a la categoría de Operaciones, por lo que las actividades que se generalizan son las correspondientes al núcleo de la metodología SOA definido y probado en el año 2005, agregando las mejoras realizadas para las Disciplinas de verificación (pruebas en COMPETISOFT) e integración.

En la sección 11.2 se presenta parte del documento enviado al proyecto COMPETISOFT con la propuesta de perfil SOA (PSOA) realizada, que además fue acompañado de plantillas para los entregables generalizadas en base a las presentadas en el Capítulo 10 - Anexo B, y de guías de implantación que corresponden a la generalización de las descripciones de las actividades presentadas en el Capítulo 5, por lo que no se presentan en esta sección.

11.2. Perfil SOA (PSOA) para COMPETISOFT

INDICE

1 Introducción

En este documento se presentan los elementos definidos en el patrón de procesos para COMPETISOFT, correspondientes al Perfil SOA para el proceso de Desarrollo de Software. Siguiendo la forma del CMMI-DEV v1.2, el proceso se nombra Desarrollo de Software + PSOA.

En primer lugar se definen los distintos alcances previstos para el Perfil SOA definido, desde las tres visiones principales del desarrollo de Software: cuando la Organización es Proveedor de software, cuando la Organización es Adquirente de software, y cuando la Organización desarrolla software para la misma Organización.

En segundo lugar se transcribe el patrón de procesos para el proceso de Desarrollo de Software, indicando como el perfil SOA afecta a cada elemento definido en el mismo. Para esto se definen los siguientes estados para las actividades, roles y entregables: se mantiene, se elimina, se sustituye, se agrega, con el siguiente significado:

- Se mantiene: el elemento del proceso no se ve afectado por el Perfil SOA
- Se elimina: el elemento no se incluye en el proceso al realizar el Perfil SOA
- Se sustituye: el elemento del proceso se sustituye en forma total o parcial, por el/los nuevo/s elemento/s que agrega el Perfil SOA
- Se agrega: este elemento no existía en el proceso y se agrega al realizar el Perfil SOA

2 Distintos Alcances del Perfil SOA (PSOA)

En esta sección se describen los distintos alcances definidos para el Perfil SOA (PSOA) agregado. Estos alcances pueden ser para una empresa como proveedora de software, como adquirente de software o una Organización con desarrollo de software propio interno a la Organización.

2.1 Empresa proveedora de software para otra Organización (Proveedor)

Aplica el perfil SOA agregado como está definido

En este caso, el Perfil SOA guiará las Fases y actividades del proceso de desarrollo que es necesario seguir para encarar un proyecto de desarrollo de este tipo.

2.2 Organización consumidora de una empresa de software (Adquirente)

Aplica el perfil SOA agregado como está definido.

En este caso, el Perfil SOA se convierte en una guía para validar las actividades del proceso de desarrollo que presente la empresa de desarrollo de software a la que se encargará el desarrollo.

2.3 Organización que desarrolla software para si misma (Área de TI)

Aplica el perfil SOA como está definido pero, para que el enfoque sea incorporado en toda la Organización debería ser extendido.

En este caso se deben tomar otras acciones a nivel organizacional para encarar un proyecto de inclusión de SOA en todos los niveles y proyectos de la Organización.

- Se debe aprobar por la gerencia la realización de dicho proyecto, definiendo los recursos, etc. asignados para el mismo.
- Se debe crear a nivel organizacional un Comité SOA integrado por personas clave en la estructura de TI en la Organización, con los objetivos de

- controlar y promover el uso de servicios en toda la Organización, tanto para gerenciar los proyectos como para sincronizar la definición y desarrollo de servicios que son compartidos por varios proyectos
- permitir el desarrollo en paralelo y la estabilidad de los distintos proyectos alrededor de las decisiones y compromisos tomados en la definición de los contratos de servicio en los que se establecen las interfaces entre las distintas capas de servicios y los datos.

- Se deben incluir también actividades específicas para la administración de los servicios de la Organización, luego que éstos son implantados en producción. Esto implica la gestión y mantenimiento de los servicios existentes, así como la de nuevos servicios que se vayan integrando como resultado de los distintos proyectos de desarrollo.

3 Desarrollo de Software + Perfil SOA (PSOA)

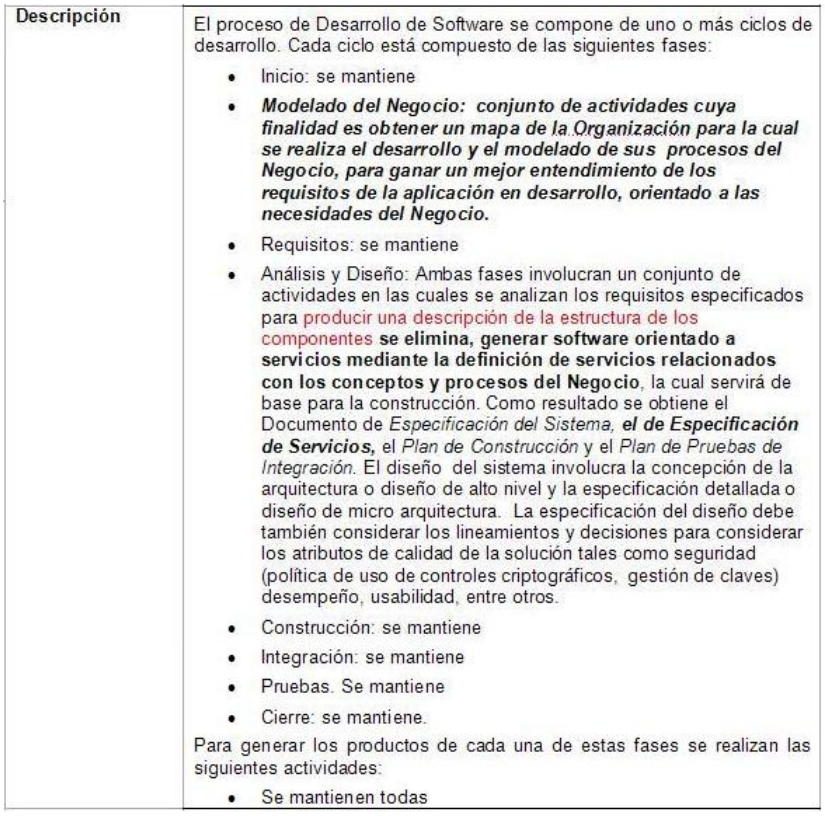
Definición general del proceso

Se toma como base la definición del proceso existente en COMPETISOFT especificado mediante el patrón de procesos, agregando y/o modificando las definiciones según el agregado y/o modificación de actividades planteadas para el Perfil SOA. El patrón de procesos se compone de varias secciones donde se incluyen las definiciones de los elementos que componen el proceso que está siendo descrito. Estas secciones se pueden dividir en grupos relacionados, donde hay una primera que constituye la presentación del proceso y se compone de: identificación del proceso que se describe, categoría a la que pertenece, propósito del proceso, descripción general del proceso, objetivos definidos para la ejecución del proceso, indicadores para la medición de los objetivos, metas cuantitativas a alcanzar, responsabilidad y autoridad y procesos relacionados.

Luego se incluye una sección que especifica las entradas y salidas asociadas con la ejecución del proceso, y productos intermedios que se obtienen durante su realización. En la sección de prácticas se incluye la definición de roles y capacidades asociadas, así como la definición y descripcion de todas las actividades que componen el proceso, para cada fase definida en la descripción del mismo, incluyendo al final de cada una un diagrama de actividad en UML que muestra la realización de las mismas. En la sección de verificación y validaciones se definen las actividades de verificación y validación que se realizan sobre las actividades y entregables que componen el proceso para asegurar la correcta realización de los mismos. Luego se incluye una sección de recursos de infraestructura necesarios para realizar las actividades definidas en el proceso, y finalmente una sección de mediciones asociadas a los indicadores definidos. En la figura 11.1 se muestra la definición general del proceso para el Perfil SOA propuesto y en la figura 11.2 la descripción asociada.

Proceso	**OPE.2 Desarrollo de Software + PSOA**
Categoría	Operación (OPE)
Propósito	El propósito del proceso de Desarrollo de Software + PSOA es la realización sistemática de las actividades de ***modelado del Negocio,*** análisis, diseño, construcción, integración y pruebas de productos de software nuevos cumpliendo con los requisitos especificados y los procesos del Negocio identificados en la Organización.

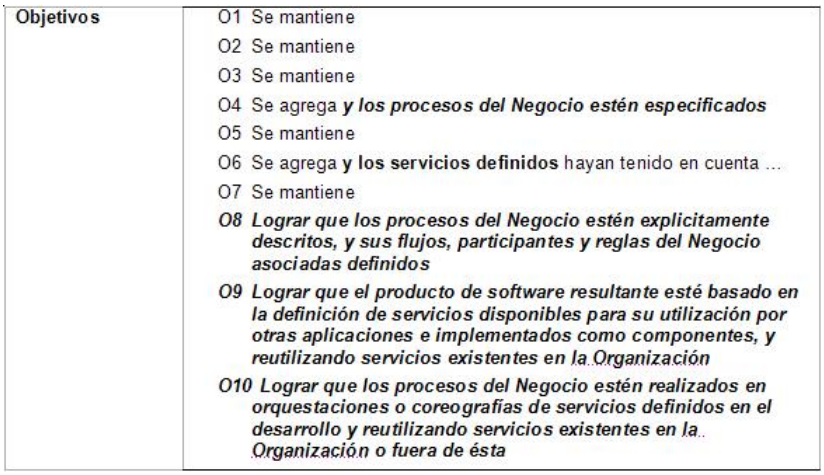
Figura 11.1: Definición general del proceso de desarrollo + PSOA



Descripción	El proceso de Desarrollo de Software se compone de uno o más ciclos de desarrollo. Cada ciclo está compuesto de las siguientes fases: • Inicio: se mantiene • ***Modelado del Negocio: conjunto de actividades cuya finalidad es obtener un mapa de la Organización para la cual se realiza el desarrollo y el modelado de sus procesos del Negocio, para ganar un mejor entendimiento de los requisitos de la aplicación en desarrollo, orientado a las necesidades del Negocio.*** • Requisitos: se mantiene • Análisis y Diseño: Ambas fases involucran un conjunto de actividades en las cuales se analizan los requisitos especificados para producir una descripción de la estructura de los componentes **se elimina, generar software orientado a servicios mediante la definición de servicios relacionados con los conceptos y procesos del Negocio**, la cual servirá de base para la construcción. Como resultado se obtiene el Documento de *Especificación del Sistema*, ***el de Especificación de Servicios,*** el *Plan de Construcción* y el *Plan de Pruebas de Integración*. El diseño del sistema involucra la concepción de la arquitectura o diseño de alto nivel y la especificación detallada o diseño de micro arquitectura. La especificación del diseño debe también considerar los lineamientos y decisiones para considerar los atributos de calidad de la solución tales como seguridad (política de uso de controles criptográficos, gestión de claves) desempeño, usabilidad, entre otros. • Construcción: se mantiene • Integración: se mantiene • Pruebas. Se mantiene • Cierre: se mantiene. Para generar los productos de cada una de estas fases se realizan las siguientes actividades: • Se mantienen todas

Figura 11.2: Descripción general del proceso de desarrollo + PSOA

Siguiendo el patrón de procesos se agregan los objetivos e indicadores correspondientes a las actividades que agrega el Perfil SOA, así como las metas cuantitativas definidas. En la figura 11.3 se presentan los objetivos definidos y en la figura 11.4 los indicadores y metas cuantitativas correspondientes.



Objetivos	O1 Se mantiene O2 Se mantiene O3 Se mantiene O4 Se agrega ***y los procesos del Negocio estén especificados*** O5 Se mantiene O6 Se agrega **y los servicios definidos** hayan tenido en cuenta ... O7 Se mantiene ***O8 Lograr que los procesos del Negocio estén explicitamente descritos, y sus flujos, participantes y reglas del Negocio asociadas definidos*** ***O9 Lograr que el producto de software resultante esté basado en la definición de servicios disponibles para su utilización por otras aplicaciones e implementados como componentes, y reutilizando servicios existentes en la Organización*** ***O10 Lograr que los procesos del Negocio estén realizados en orquestaciones o coreografías de servicios definidos en el desarrollo y reutilizando servicios existentes en la Organización o fuera de ésta***

Figura 11.3: Objetivos agregados por el PSOA

Indicadores	I1 (O1) se mantiene I2 (O2) se mantiene I3 (O3) se mantiene I4 (O4) se agrega ***y la realización de procesos del Negocio mediante los servicios definidos está prevista*** I5 (O5) se mantiene I6 (O6) se mantiene I7 (O7) se mantiene I8 (O8) ***el Documento de Especificación de Procesos del Negocio contenga la información y descripcion de los procesos del Negocio identificados en la Organización como está establecido*** I9 (O9) ***el Documento de Especificación de Servicios contenga la información y descripcion de los servicios definidos como está establecido*** I10 (O10) ***el Documento de Especificación de Servicios contiene la información de orquestaciones y coreografías de servicios definidas para realizar los procesos del Negocio , indicando claramente que servicios intervienen en las mismas***
Metas cuantitativas	Valor numérico o rango de satisfacción por indicador: Se agregan: **M5 que el 100% de los procesos del Negocio de la Organización definidos para la aplicación en desarrollo estén modelados en la Especificación de Procesos del Negocio.** ***M6 que el 80% del grupo técnico haya revisado y aprobado el documento de Especificación de Servicios.***

Figura 11.4: Indicadores y metas cuantitativas agregados por el PSOA

En el patrón de procesos se describen a continuación las entradas y salidas del proceso, las principales salidas que agrega el Perfil SOA se presentan en la figura 11.5 las correspondientes a la fase agregada de Modelado del Negocio y en la figura 11.6 las correspondientes a las actividades de diseño agregadas, donde se muestran también las actividades de verificación y validación asociadas. Adicionalmente se modifica la descripción de otras salidas del proceso de desarrollo como ser Especificación del Sistema, manteniendose igual el resto de las salidas definidas, las que no se presentan.

Salidas

Nombre	Descripción	Destino	Plantilla Soporte	Forma de aprobación
Evaluación de Organización Objetivo	Este documento describe el estado actual de la Organización en la cual la aplicación será implantada, en términos de sus procesos del Negocio, herramientas, competencias de las personas y actitudes, clientes, competidores, tendencias tecnológicas, problemas y áreas de mejora posibles.	Administración de un Proyecto Específico	Tiene plantilla	Ver14
Especificación de los Procesos del Negocio	Este documento describe como se utiliza el negocio por parte de sus clientes y socios. Los procesos del Negocio dan valor a los actores involucrados, y constituyen el vehículo con el que el Negocio funciona. Las estrategias del Negocio se traducen en Objetivos del Negocio que guían las actividades realizadas en los procesos del Negocio, por lo que cada proceso en general soporta por lo menos un objetivo del Negocio.	Administración de un Proyecto Específico	Tiene plantilla	Ver15, Val15

Figura 11.5: Salidas agregadas por la fase Modelado del Negocio el PSOA

Servicio	Un servicio consiste en una implementación que provee lógica del Negocio y datos, un contrato de servicio que especifica la funcionalidad, el uso y las restricciones para un cliente del servicio, y una interface del servicio que expone fisicamente la funcionalidad.	Administración de un Proyecto Específico	No tiene plantilla	Ver16, Val16
Especificación de Servicios	En este documento se especifican los servicios identificados, indicando el tipo de servicio correspondiente. Se compone de las siguientes secciones: Identificación y categorización: Se debe describir para cada servicio, en lenguaje natural de forma clara y sin ambigüedades, la justificación de porqué se define ese servicio, que funcionalidades provee indicando a que funcionalidad y proceso del Negocio corresponden, de que tipo es el servicio (según la clasificación elegida), indicando también que subsistema debe proveer este servicio. Especificación: Para cada servicio identificado se realiza su contrato funcional, especificando para cada interface definida los métodos que deben ser implementados para proveer el servicio acordado para la interface Servicios existentes: En esta sección se listan los servicios existentes en la Organización identificados, que sirvan para reutilizar en la aplicación en desarrollo.	Administración de un Proyecto Específico	Tiene plantilla	Ver16, Val16

Figura 11.6: Salidas agregadas por las actividades incluidas en la fase de Diseño

A continuación en el patrón de procesos se detallan las prácticas definidas para el proceso de desarrollo, que incluye los roles participantes y sus competencias, así como las actividades para cada una de las fases que lo componen: Inicio, Requisitos, Análisis, Diseño, Construcción, Integración y Pruebas.

En lo que sigue se presentan las principales actividades agregadas y/o modificadas por el Perfil SOA para cada una de las fases definidas, comenzando en la figura 11.7 por las fases de Inicio y de Modelado del Negocio que se agrega completa.

Actividades

Se asocian a los objetivos y describen las tareas y roles responsables.

Se agrega la Fase de Modelado del Negocio de la Organización para la cual se realiza el desarrollo, se agregan actividades en las Fases de Análisis, Diseño, Construcción, Integración y Pruebas.

Rol	Descripción
A1. Realización de la Fase de Inicio (O3)	
Entradas	Plan de Desarrollo, Definición del Proyecto
ET	A1.1. **Identificar los usuario claves para la definición de los procesos del Negocio a modelar en el desarrollo.** Se agrega
RD	A1.2. se mantiene.
Salidas	*Reporte de Actividades*
PSOA - A1. *Realización de la Fase de Modelado del Negocio* (O1, O3, O4,O8) SE AGREGA COMPLETA	
Entradas	Plan de Desarrollo, Definición del Proyecto
RD AN AR	PSOA – A1.1. Distribuir tareas a los miembros del equipo de trabajo según su rol, de acuerdo al *Plan de Desarrollo* actual.
AN AR	**PSOA – A1.2 Evaluar la Organización Objetivo** • Identificar y comprender aspectos del negocio, funcionamiento, competencia, necesidades, problemas y áreas de mejora de la Organización para la cual se realiza el desarrollo • Identificar tecnologías que se utilizan, informatización existente, personas y sus competencias, actitudes y roles en la Organización
RE	**PSOA – A1.3 Verificar la *Evaluación de la Organización Objetivo* (Ver14).**
AN	**PSOA – A1.4** Corregir los defectos encontrados en la *Evaluación de la Organización Objetivo* con base en el *Reporte de Verificación* y obtener la aprobación de las correcciones.
AN AR	**PSOA – A1.5 Identificar los procesos del Negocio** • Definir los límites del Negocio a modelar en cuanto a procesos del Negocio, involucrados en su realización (secciones, roles de la misma u otras Organizaciones) • Realizar reuniones de trabajo con los usuarios seleccionados para definir los procesos del Negocio a modelar • Definir los procesos del Negocio incluyendo el flujo de ejecucion (en forma textual y/o gráfica), los involucrados, las reglas del Negocio que aplican (incluyendo cálculos, time-outs de los pasos o subprocesos involucrados, entre otros) • Identificar y definir en conjunto con el cliente los términos que se manejan en los procesos del Negocio y el negocio en general
RE	**PSOA – A1.6 Verificar la *Especificación de Procesos del Negocio* (Ver15).**
AN	**PSOA – A1.7** Corregir los defectos encontrados en la *Especificación de Procesos del Negocio* con base en el *Reporte de Verificación* y obtener la aprobación de las correcciones.
CL US RPU	**PSOA – A1.8** Validar la *Especificación de Procesos del Negocio* **(Val5).**
AN	**PSOA – A1.9** Corregir los defectos encontrados en la *Especificación de Procesos del Negocio* con base en el *Reporte de Validación* y obtener la aprobación de las correcciones.
RD	**PSOA – A1.10** Incorporar la *Evaluación de la Organización Objetivo*, la *Especificación de Procesos del Negocio y el Glosario de términos del Negocio* a la *Configuración de Software*.
RD	**PSOA – A1.11** Elaborar el *Reporte de Actividades* registrando las actividades realizadas, fechas de inicio y fin, responsable por actividad y mediciones requeridas.
Salidas	Evaluación de la Organización Objetivo Especificacion de Procesos del Negocio Glosario de términos del Negocio

Figura 11.7: Fase de Modelado del Negocio agregada por PSOA

En la fase de Requisitos se agregan conceptos a la descripción de algunas de las actividades definidas, en la fase de Análisis se mantienen en general las descripciones previstas, con modificaciones en algunas que permiten la introducción al diseño orientado a servicios. En la figura

11.8 se presenta a modo de ejemplo las modificaciones realizadas en la descripción de la fase de Requisitos, las actividades intermedias que no se presentan se mantienen como estaban definidas.

A2. Realización de la Fase de Requisitos (O1, O3, O7)	
Entradas	Se agregan ***Evaluación de la Organización Objetivo, Especificacion de Procesos del Negocio, Glosario de términos del Negocio***
RD AN	A2.1. se mantiene
AN CL US ES **AR**	A2.2. **Se agregan** • Utilizar también para identificar los requisitos, la información de la Organización en que se realiza el desarrollo provista por el documento de Evaluación de la Organización Objetivo • Utilizar también para identificar los requisitos, la especificación de los Procesos del Negocio definidos y modelados en el documento de Especificación de Procesos del Negocio • Analizar la consistencia de los requisitos identificados con los procesos del Negocio definidos y modelados en el documento de Especificación de Procesos del Negocio
RPU AN	A2.10. se agrega ***incluyendo la prueba de los Procesos del Negocio definidos en el documento de Especificación de Procesos del Negocio***
RE	A2.11. se mantiene

Figura 11.8: Modificaciones en la Fase de Requisitos del PSOA

En la fase de Diseño se agregan las actividades principales al proceso de desarrollo que plantea la Metodología SOA propuesta en el Perfil SOA, como se presentó en el Capítulo estas actividades y sus correspondientes entregables salida guían el diseño orientado a servicios, a partir de los procesos del Negocio identificados y los requisitos establecidos. En la figura 11.9 se presentan las actividades agregadas a la fase Diseño.

A4. Realización de la Fase de Diseño (O1, O3, O5, O6, O8, O9)	
AN, Di, DU, AR ST	**PSOA – A4.1 Identificar y categorizar servicios** se agrega • identificar los servicios necesarios para realizar los procesos del negocio, clasificándolos según el tipo de servicio. A partir de la *Especificación del Sistema*, analizar los subsistemas definidos, asegurándose que existe un mapeo de por lo menos un subsistema a cada área funcional del Negocio. • identificar en que subsistema se realiza cada funcionalidad, comenzando con las más relevantes para la implementación. Identificar los servicios que deben ser provistos por cada subsistema para realizar las distintas funcionalidades y los Procesos del Negocio asociados a las mismas. • Actualizar la *Especificación de Servicios* en la sección corespondiente, para que incluya los servicios identificados y categorizados.
RAPE, RD	**PSOA – A4.2 Especificar servicios** se agrega • Especificar los servicios identificados, definiendo los contratos de servicio para cada uno incluyendo las interfaces que brindará y sus operaciones, parámetros, etc. • Actualizar la *Especificación de Servicios* en la sección correspondiente, para que incluya la especificación de servicios realizada.
RAPE, RD	**PSOA – A4.3 Investigar servicios existentes** se agrega • Encontrar servicios que ya estén implementados en la Organización y puedan o deban ser reutilizados en la aplicación en desarrollo, y que puedan incluso ser brindados por otra Organización. • Actualizar la *Especificación de Servicios* en la sección corespondiente, para que incluya la investigación de servicios existentes realizada.

Figura 11.9: Actividades agregadas en la fase Diseño por el PSOA

A4. Realización de la Fase de Diseño (O1, O3, O5, O6, O8, O9)	
RAPE, RD	**PSOA – A4.4 Asignar servicios a componentes** se agrega • Definir los componentes que deberán ser implementados para proveer los servicios especificados, esta correspondencia no tiene porque ser 1:1 con los subsistemas identificados previamente. Asimismo se debe especificar para cada componente los servicios que provee • Para cada componente necesario se realiza la especificación del componente, indicando reglas del Negocio y servicios que implementa, así como otros componentes que utiliza. • Actualizar la *Especificación de Servicios* en la sección correspondiente, para que incluya la asignación de servicios a componentes definida.
RAPE, RD	**PSOA – A4.5 Definir Orquestación de servicios** se agrega • Definir la secuencia de interacción entre servicios que es necesaria para realizar los procesos del Negocio identificados en la *Especificación de procesos del Negocio*. La orquestación de servicios para la implementación de procesos del Negocio se puede mostrar gráficamente en el diagrama que se considere más apropiado. • Actualizar la *Especificación de Servicios* en la sección correspondiente, para que incluya las orquestaciones de servicios definidas.

Figura 11.10: Actividades agregadas en la fase Diseño por el PSOA - continuación

Se agregan también actividades de verificación y validación sobre los servicios definidos y especificados mediante las actividades presentadas. En la figura 11.11 se presenta el resto de las actividades agregadas a la fase de Diseño.

A4. Realización de la Fase de Diseño (O1, O3, O5, O6, O8, O9)	
RE	**PSOA – A4.6** se agrega Verificar la *Especificación de servicios* y el *Registro de Rastreo* (**Ver16**).
AN AR	**PSOA – A4.7** se agrega Corregir los defectos encontrados en la *Especificación de servicios*, y en el *Registro de Rastreo* con base en el *Reporte de Verificación* y obtener la aprobación de las correcciones.
CL RPU	**PSOA – A4.8** se agrega Validar la *Especificación de servicios* (**Val6**).
AN AR	**PSOA – A4.9** se agrega Corregir los defectos encontrados en la *Especificación de servicios* con base en el *Reporte de Validación* y obtener la aprobación de las correcciones.
RAPE	A4.6. Realizar la estimación de construcción de cada unidad funcional y **de los servicios asociados.**
RAPE, RD	A4.7. Definir el *Plan de Construcción* ***basado en la implementación de los servicios definidos***
RE	A4.8. se mantiene.
AN, DI	A4.9. se mantiene.
RD	A4.10. Incorporar la *Especificación del Sistema*, ***Especificación de Servicios***, *Registro de Rastreo* y el *Plan de Construcción* como líneas base a la *Configuración de Software*.
RD	A4.11. se mantiene idem.
RD	A4.12. se mantiene
Salidas	*Especificación del Sistema* ***Especificación de Servicios*** *Registro de Rastreo* *Plan de Construcción* *Configuración del Software* *Reporte de Actividades*

Figura 11.11: Actividades agregadas en la fase de Diseño por el PSOA - continuación

En las fases de Construcción, Integración y Pruebas se modifican algunas descripciones para incluir los componentes asignados a servicios definidos, y para tener en cuenta los procesos

del Negocio identificados, los servicios definidos y componentes asignados a servicios para la definición de los Casos de Prueba, y para la ejecución de las pruebas sobre los componentes definidos. En la figura 11.12 se presentan las modificaciones realizadas a la fase de Construcción, en la figura 11.13 las modificaciones realizadas a la fase de Integración y en la figura 11.14 las modificaciones realizadas a la fase de Pruebas.

A5. Realización de la Fase de Construcción (O1, O3)	
Entradas	*Se agrega **Especificación de Servicios***
RDM	A5.1. se mantiene.
PR	A5.2. Construir o modificar el(los) *Componente*(s) de software: • Implementar o modificar *Componente*(s) con base a la parte detallada de la *Especificación del Sistema*. **Se sustituye por:** en la especificación de servicios y asignación a componentes definida • Se mantiene
	A5.3 Realizar pruebas unitarias • Definir y aplicar pruebas unitarias para verificar que el funcionamiento de cada componente esté acorde con la parte detallada de la *Especificación del Sistema*. **Se sustituye por:** especificación de servicios y asignación a componentes realizada. • El resto se mantiene
RE	A5.3. se mantiene
PR	A5.4. se mantiene
RDM	A5.5. se mantiene idem
Salidas	***Componente(s) asignados a servicios implementados*** *Registro de Rastreo* *Configuración del Software*

Figura 11.12: Modificaciones en la fase de Construcción del PSOA

A6. Realización de la Fase de Integración (O1, O3)	
Entradas	*Se agrega **Especificación de Servicios***
RD	A6.1. se mantiene.
PR RPU	A6.2. Realizar integración y pruebas. • Verificar que todas las unidades funcionales están listas **Se sustituye por:** todos los componentes asignados a servicios están listos para su integración • Se mantiene • Integrar todas las unidades funcionales **Se sustituye por:** todos los componentes asignados a servicios de acuerdo al procedimiento definido • El resto se mantiene.
RD	A6.3. se mantiene, *idem*
RD	A6.4. se mantiene idem.
Salidas	*Software* *Reporte de Pruebas de Integración* *Sistema de Seguimiento de Defectos* *Registro de Rastreo* *Configuración del Software* *Reporte de Actividades*

Figura 11.13: Modificaciones en la fase de Integración del PSOA

A7. Realización de la Fase de Pruebas (O1, O3, O7)	
Entradas	*Se agregan* ***Especificación de procesos del Negocio, Especificación de Servicios***
RD	A7.1. *se mantiene.*
RPU, ES	A7.2 Diseñar los *Casos de Prueba del Sistema*, en base a al *Plan de Pruebas del Sistema* y el *Plan de Pruebas de Seguridad y a la Especificación de Procesos del Negocio y la Especificación de Servicios* • el resto se mantiene
RPU, ES	A7.3 se mantiene El resto se mantiene
RPU, RS	A7.4 se mantiene
RPU CL, RS	A7.5 se mantiene
RPU	A7.6 se mantiene.
	A7.7 se mantiene El resto se mantiene
PR,RPU, CL	A7.8 se mantiene

Figura 11.14: Modificaciones en la fase de Pruebas del PSOA

Se definen también las actividades de verificación y validación que se realizan sobre las actividades y entregables agregados para asegurar la correcta realización de los mismos. En la figura 11.15 se presentan las actividades de verificación y validación agregadas por el PSOA para la fase de Modelado del Negocio agregada, y en la figura se presentan las correspondientes a la fase de Diseño.

Verificación o Validación	Actividad	Producto	Rol	Lineamientos de Verificación o Validación
Ver14		*Evaluación de Organización Objetivo*	RE, ES	Verificar claridad de la documentación de la *Evaluación de la Organización Objetivo*, que contiene toda la información requerida y con el estándar de documentación requerido en el *Proceso Específico*. Los defectos encontrados se documentan en un *Reporte de Verificación*.
Ver15		*Especificación de Procesos del Negocio*	RE, ES	Verificar la claridad de redacción de la *Especificación de Procesos del Negocio* y su consistencia con la *Evaluación de la Organización Objetivo* y la *Descripción del Producto* y con el estándar de documentación requerido en el *Proceso Específico*. Adicionalmente revisar que los pasos y las opciones en los procesos del Negocio descritos sean completos y no ambiguos o contradictorios. Los defectos encontrados se documentan en un *Reporte de Verificación*.
Val5		*Especificación de Procesos del Negocio*	CL, RPU, ES	Validar que la *Especificación de Procesos del Negocio* cumple con las necesidades y expectativas acordadas. Los defectos encontrados se documentan en un *Reporte de Validación*.

Figura 11.15: Definición de actividades de verificación y validación para la fase Modelado del Negocio del PSOA

Verificación o Validación	Actividad	Producto	Rol	Lineamientos de Verificación o Validación
Ver16		*Especificación de Servicios*	RE, ES	Verificar claridad de la documentación de la *Especificación de Servicios*, su factibilidad y la consistencia con la *Especificación de Procesos del Negocio*, *Especificación de Requisitos* y *Especificación del Sistema*, y con el estándar de documentación requerido en el *Proceso Específico*. Verificar que el *Registro de Rastreo* contenga las relaciones adecuadas entre los procesos del Negocio, los requisitos, los elementos de la *Especificación del Sistema* y de la *Especificación de Servicios*. Los defectos encontrados se documentan en un *Reporte de Verificación*.
Val6		*Especificación de Servicios*	CL, RPU, ES	Validar que la *Especificación de Servicios* está de acuerdo con las necesidades y expectativas acordadas con el cliente según los documentos relacionados de *Especificación de Procesos del Negocio*, *Especificación de Requisitos* y *Especificación del Sistema*. Los defectos encontrados se documentan en un *Reporte de Validación*.

Figura 11.16: Definición de actividades de verificación y validación para la fase de Diseño del PSOA

Finalmente se definen las mediciones asociadas a los indicadores definidos y la forma de realizarlas. En la figura 11.17 se muestra la definición de mediciones agregadas por el Perfil SOA.

Mediciones	Mediciones que se establecen para evaluar los indicadores del proceso. Las mediciones se identifican como M1, M2, etc. y entre paréntesis se especifica la identificación del indicador que le corresponde.			
Medición	**Indicador**	**Objeto de medición**	**Rol**	**Mecanismo de medición**
M8	I8	*Especificación de Procesos del Negocio*	AN, RD	Comprobar que los procesos del Negocio están definidos y contienen la información de los flujos, participantes y reglas del Negocio correspondientes
M9	I9	*Especificación de Servicios*	AR	Comprobar que en la definición de servicios se han tenido en cuenta los requisitos planteados por el usuario
M10	I10	*Especificación de Servicios*	AR	Comprobar que la definición de servicios ha tenido en cuenta la orquestación y coreografía de servicios de acuerdo a los procesos del Negocio identificados

Figura 11.17: Mediciones asociadas a indicadores agregadas por PSOA

4 Referencias

Delgado, A.: "Metodología de desarrollo para aplicaciones con enfoque SOA (Service Oriented Architecture)", Tesis de Maestría en Informática, PEDECIBA Informática, Instituto de computación, Facultad de Ingeniería, Universidad de la Republica, Uruguay, ISSN: 0797-6410, 2007.

Bibliografía

[ADBP00] Delgado A., Pérez B., Modelado de proceso de software, Informe final Proyecto Taller V, Instituto de computación, Facultad de Ingeniería, Universidad de la República, 2000.

[ADBP06] Delgado A., Pérez, B., Modelo de Desarrollo de Software OO, Experimentación en un curso de Ingeniería de Software, en V Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento (JIISIC'06), Actas, pp. 39-46, ISBN 970-94770-0-5, Puebla, México, Febrero de 2006.

[AGMA01] Beck, K., Cockburn, A., Fowler, M., Mellor, S., et al., Manifesto for Agile Software Development, <http://agilemanifesto.org/>[consulta 31 de julio de 2007]

[ANDR] AndroMDA, <http://www.andromda.org>[consulta 31 de julio de 2007]

[ARSA04] Arsanjani A., Service-oriented modeling and architecture, How to identify, specify, and realize services for your SOA, <http://www.ibm.com/developerworks/webservices/library/ws-soa-design1/, noviembre 2004>[consulta 31 de julio de 2007]

[ASCA05] Beisiege M. et al, Service Component Architecture (SCA), Building Your First Application - Simplified BigBank, noviembre 2005, <http://www.osoa.org/download/attachments/28/SCA_BuildingYourFirst Application_V09.pdf?version=1 [consulta 31 de julio de 2007]

[ASPI04] Hurtado, J., Pino, F., Vidal J., Agile Software Process Improvement (SPI), Sistema Integral para el Mejoramiento de los Procesos de Desarrollo de Software en Colombia (SIMEP-SW), Colciencias - Universidad del Cauca, Popayán, Colombia, 2003.

[ATAM04] Clements P., Kazman R., Klein R., Evaluating Software Architectures, Methods and Case Studies, Addison-Wesley Professional, 1st edition, 2002, 2nd. printing, 2004, 368 p., ISBN 0-201-70482-X

[ATAM07] Delgado A., Castro A., Germán M., "Evaluación de Arquitecturas de Software con ATAM (Architecture Tradeoff Analysis Method): un caso de estudio", en VI Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento (JIISIC'07), Actas, pp. 151-159, ISBN 978-9972-2885-1-7, Lima, Perú, Febrero de 2007.

[AJAX] Asynchronous JavaScript and XML (AJAX), <http://en.wikipedia.org/wiki/Ajax_framework>[consulta 31 de julio de 2007]

[BASS03] Bass L., Clements P., Kazman R., Software Architecture in Practice, Addison-Wesley Professional, 2nd. edition, 2003, 560 p., ISBN 0-321-15495-9

[BECK99] Beck K., Extreme Programming Explained: Embrace Change, Addison-Wesley Professional, 224 p., 1999, ISBN 0-201-61641-6

[BPMI] Business Process Management Initiative, <http://www.bpmi.org/>[consulta 31 de julio de 2007]

[BPMN06] Business Process Modeling Notation (BPMN) Specification, Object Management Group (OMG) Final Adopted Specification, febrero 2006, dtc/06-02-0, BPMI 2004 <http://www.omg.org/bpm/>[consulta 31 de julio de 2007]

[C++] Stroustrup, B., The C++ Programming Language, 3rd Edition, 2000, 1030 p., ISBN 0-201-70073-5

[CES] Centro de Ensayos de Software (CES), Montevideo, Uruguay, <http://www.ces.com.uy/index.html>[consulta 31 de julio de 2007]

[CMMI06] Capability Maturity Model Integration (CMMI) for development (CMMI-DEV) v.1.2, Software Engineering Institute (SEI), 2006, <http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr008.pdf>[consulta 31 de julio de 2007]

[COMP06] proyecto COMPETISOFT 506PI287- Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software de Iberoamérica, financiado por CYTED con el código 3789, <http://alarcos.inf-cr.uclm.es/Competisoft/>[consulta 31 de julio de 2007]

[CONA05] Conallen, J., Brown, A., An introduction to model-driven architecture Part III: How MDA affects the iterative development proces, IBM-Rational, The rational Edge, 2005, <http://www.ibm.com/developerworks/rational/library/may05/brown/index.html> [consulta 31 de julio de 2007]

[CORB] Common Object Request Broker Architecture (CORBA), <http://www.corba.org/>[consulta 31 de julio de 2007]

[CYTED] Programa Iberoamericano de Ciencia y Tecnología para el Desarrollo (CYTED), <http://www.cyted.org/>[consulta 31 de julio de 2007]

[DSDM] Dynamic Systems Development Method (DSDM), <http://www.dsdm.org/>[consulta 31 de julio de 2007]

[dX98] Booch G., Martin R., Newkirk J., Object Oriented Analysis and Design with Applications, Addison-Wesley, 2nd. edition, 1998, chapter 4, The process, <http://www.objectmentor.com/resources/articles/RUPvsXP.pdf>[consulta 31 de julio de 2007]

[EJB] Enterprise JavaBeans Technology, <http://java.sun.com/products/ejb/>[consulta 31 de julio de 2007]

[EMDA06] Proyecto de grado "Extensión MDA (Model Driven Architecture) para el Proceso de Desarrollo de Software del curso Proyecto de Ingeniería de Software (PIS)", Tutor Andrea Delgado, Estudiantes Natacha Carballal y Catalina Rapetti, 2006.

[EMDA07] Delgado A., Carballal N., Rapetti C., Extensión MDA (Model Driven Architecture) para proceso basado en RUP (Rational Unified Process), en VI Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento (JIISIC'07), Actas, pp. 247-255, ISBN 978-9972-2885-1-7, Lima, Perú, Febrero de 2007.

[ENDR04] Endrei M., Ang J., Arsanjani A., et. al, Patterns: Service-oriented Architecture and Web Services, IBM Redbook, SG24-6303-00, 2004, <http://www.redbooks.ibm.com/redbooks/pdfs/sg246303.pdf>[consulta 31 de julio de 2007]

[EPF] Eclipse Process Framework (EPF), Composer 1.0 Architecture Overview, <http://www.eclipse.org/epf/composer_architecture/>[consulta 31 de julio de 2007]

[ERLT05] Erl, T., Service-Oriented Architecture: Concepts, Technology, and Design, Prentice Hall, 1st. edition, 2005, 792 p., ISBN: 0-13-185858-0

[ESOA06] Delgado A., González L., Piedrabuena F. , Desarrollo de aplicaciones con enfoque SOA (Service Oriented Architecture), en V Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento (JIISIC'06), Actas, pp. 237-244, ISBN 970-94770-0-5, Puebla, México, Febrero de 2006.

[EVAL05] Oktaba H., Esquivel C., Ramos A., et. al, EvalProSoft, Modelo de evaluación de procesos para la Industria del Software, Secretaría de Economía de México, 2005.

[FIBE06] Foro Iberoamericano de Ciencia, Tecnología, Empresa y Sociedad (FIBECYT), <http://www.cyted.org/fibecyt/>[consulta 31 de julio de 2007]

[GARL03] Clements P., Bass L.., Garlan D., et al., Documenting Software Architectures: Views and Beyond, Addison-Wesley Professional, 1st. edition, 2002, 2nd. printing, 2003, 560 p., ISBN 0-201-70372-6

[GAVR04] Gavras A., Belaunde M., Ferreira L., Almeida J.P., Towards an MDA-based development methodology for distributed applications, <http://modeldrivenarchitecture.esi.es/pdf/paper3-3.pdf>[consulta 31 de julio de 2007]

[GOF95] Gamma, E.,Helm, R.,Johnson, R.,Vlissides, J., Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Professional, 1st. edition, 1995, 365 p., ISBN 0-201-63361-2

[GRIS06] Grupo de Ingeniería de Software, Instituto de Computación, Facultad de Ingeniería, Universidad de la República, <http://www.fing.edu.uy/inco/grupos/gris/>[consulta 31 de julio de 2007]

[GR0405] Proyecto de Ingeniería de Software, edición 2005, Grupos 4 y 5, Instituto de computación, Facultad de Ingeniería, Universidad de la República, <http://www.fing.edu.uy/inco/cursos/ingsoft/pis/>

[GXAR02] Genexus, Artech Consultores, Uruguay, <http://www2.gxtechnical.com/portal/hgxpp001.aspx?15,8,8,O,E,0,,688,>[consulta 31 de julio de 2007]

[IEEE90] IEEE std. 610.12-1990, IEEE Standard Glosary of Software Engineering Terminology, 1990.

[IEEE00] IEEE std. 1471-2000, IEEE Recommended Practice for Architectural Description of Software-Intensive System, 2000.

[INSW68] Bauer, F., Primer conferencia sobre Desarrollo de Software patrocinada por el Comité de Ciencia de la OTAN, Garmisch, Alemania, octubre de 1968.

[ISO95] Norma ISO/IEC 12207, Information technology - Software life cycle processes, 1995, ISO/IEC 12207:1995/Amd 1:2002, ISO/IEC 12207:1995/Amd 2:2004.

[ISO98] Norma ISO/IEC 15504, Software Process Improvement and Capability dEtermination (SPICE), 1998, ISO/IEC 15504-1:2004, ISO/IEC 15504-2:2003, ISO/IEC 15504-4:2004, ISO/IEC 15504-5:2006.

[ISO00] ISO 9000:2000, <http://www.iso.org/iso/>[consulta 31 de julio de 2007]

[ISO05] Norma ISO/IEC 25000, Software product Quality Requirements and Evaluation (SQuaRE), SQuaRE reemplaza las series actuales de normas ISO/IEC 9126 e ISO/IEC 14598 para la Calidad del Software, 2005

[JAVA] Java, <http://java.sun.com/>[consulta 31 de julio de 2007]

[JBPM] JBoss JBPM, <http://www.jboss.com/products/jbpm>[consulta 31 de julio de 2007]

[JCA] J2EE Connector Architecture, <http://java.sun.com/j2ee/connector/>[consulta 31 de julio de 2007]

[JMS] Java Message Service (JMS), <http://java.sun.com/products/jms/>[consulta 31 de julio de 2007]

[J2EE] Java 2 Platform, Enterprise Edition (J2EE), <http://java.sun.com/j2ee/overview.html>[consulta 31 de julio de 2007]

[JMET] Apache JMeter, <http://jakarta.apache.org/jmeter/>[consulta 31 de julio de 2007]

[KRAF05] Krafzig, D. Banke, K. Slama, D., Enterprise SOA, Service Oriented Architecture Best Practices, Prentice Hall, 410 p., 2005, ISBN 0-13-146575-9

[KRUC95] Kruchten P., The 4+1 View Model of Architecture, IEEE Software, Noviembre 1995, Volumen: 12, pags: 42-50, ISSN: 0740-7459

[KITS00] Kitson, D., Capability Maturity Model Integration (CMMI): Current Status and Standardization Aspects, SEI, en 3rd. Annual Systems Engineering & Supportability Conference, National Defense Industrial Association (NDIA), 23 al 26 de Octubre de 2000, <http://www.dtic.mil/ndia/systems/Kitson.pdf>[consulta 31 de julio de 2007]

[KLEP03] Kleppe A.,Warmer J., Bast W., MDA Explained. The Model Driven Architecture: Practice and Promise, Pearson Education, 1st. edition, 2003, 4th. printing 2005, 188 p., ISBN 0-321-19442-X

[LIMS05] Proyecto LIMS, 2005, <http://www.fing.edu.uy/inco/pm/Convenios/LIMS2005> [consulta 31 de julio de 2007]

[LKAL05] Proyecto Link-All, Programa @lis, Unión Europea, <http://www.link-all.org/>[consulta 31 de julio de 2007]

[MDAC02] Carter, K., Supporting MDA with eXecutable UML, Ref: CTN 80 v2.2, Kennedy Carter Ltd., 2002 <http://www.kc.com>[Consulta: 31 de julio de 2007]

[MDAD05] Delgado, A., Model Driven Architecture (MDA): enfoque para el desarrollo de software basado en modelos, TR-05-18, 2005, Biblioteca PEDECIBA, Instituto de Computación, Facultad de Ingeniería, Universidad de la República.

[MERC] Mercury LoadRunner, <http://www.mercury.com/us/products/performance-center/loadrunner/works.html>[Consulta: 31 de julio de 2007]

[MOPR04] Oktaba H., Esquivel C., Ramos A., et. al, MoProSoft, Modelo de procesos para la Industria del Software, Secretaría de Economía de México, 2004

[MDA03] Object Management Group (OMG), Model Driven Architecture (MDA) Guide version 1.0.1, junio 2003. <http://www.omg.org/mda/>[consulta 31 de julio de 2007]

[MELL04] Mellor S., Scott K.,Uhl A.,Weise D., MDA Distilled: Principles of Model-Driven Architecture, Addison Wesley, 1st. edition, 2004, 176 p., ISBN 0-201-78891-8

[MNET] Microsoft .NET Framework 3.0 Programming Model <http://msdn2.microsoft.com/en-us/library/ms687300.aspx>[consulta 31 de julio de 2007]

[MSOA06] Proyecto de grado Enfoque SOA (Service Oriented Architecture) con integración de aplicaciones móviles para LIMS (Laboratory Information Management System), Tutores Raúl Ruggia, Andrea Delgado, Estudiantes Amparo Pittier, Diego Pérez, Alvaro Garepe, Eduardo Restuccia, 2006.

[OMG] Object Management Group (OMG), <http://www.omg.org/>[consulta 31 de julio de 2007]

[OASIS] Organization for the Advancement of Structured Information Standards (OASIS), <http://www.oasis-open.org>[consulta 31 de julio de 2007]

[OPUP] Open Unified Process (OpenUP), Version 1.0, Julio 2007, <http://www.epfwiki.net/wikis/openup/>[consulta 31 de julio de 2007]

[OWEN03] Owen, M., Raj, J., BPMN and Business Process Management, Introduction to the New Business Process Modeling Standard, Popkin Software, 2003, <http://www.popkin.com>[consulta 31 de julio de 2007]

[PCFE02] Delgado, A., Procesamiento de cuestionarios finales a los estudiantes, edición año 2001 del curso Proyecto de Ingeniería de Software, enero 2002.

[PFLE02] Pfleeger, S.L., Ingeniería de Software, teoría y práctica, Prentice Hall, 1era. edición en español, 2002, p., ISBN 987-9460-71-5

[PHP] PHP: Hypertext Preprocessor, <http://www.php.net/>[consulta 31 de julio de 2007]

[PIS06] Proyecto de Ingeniería de Software, Instituto de computación, Facultad de Ingeniería, Universidad de la República, <http://www.fing.edu.uy/inco/cursos/ingsoft/pis/>[consulta 31 de julio de 2007]

[POSA96] Buschmann F., Meunier R., Rohnert H., Sommerlad P., Stal M., Pattern-oriented Software Architecture Volume 1: A system of Patterns, John Wiley&Sons, 1st. edition, 1996, 476 p., ISBN 0-471-95869-7

[POSA00] Schmidt D., Stal M., Rohnert H., Buschmann F., Pattern-Oriented Software Architecture Volume 2: Patterns for Concurrent and Networked Objects, John Wiley&Sons, 1st. edition, 2000, 666 p., ISBN 0-471-60695-2

[PYKE05] Pyke J., BPM in context: now and in the future, WfMC Chair, United Kingdom, 2005, <http://www.e-workflow.org/downloads/Pyke_BPM_in_Context.pdf>[consulta 31 de julio de 2007]

[QAW03] Barbacci, M., Ellison, R., Lattanze, A., Stafford, J., Weinstock, C., Wood, W., Quality Attribute Workshops (QAW), CMU/SEI-2003-TR-016, SEI, 3rd edition, 2003, <http://www.sei.cmu.edu/pub/documents/03.reports/pdf/03tr016.pdf>[consulta 31 de julio de 2007]

[RMC] IBM - Rational Method Composer (RMC), <http://www-306.ibm.com/software/awdtools/rmc/>[consulta 31 de julio de 2007]

[RMCK] Haumer, P., IBM - Rational Method Composer (RMC), Part 1: key concepts, 2005, <http://www-128.ibm.com/developerworks/rational/library/dec05/haumer/>[consulta 31 de julio de 2007]

[RMOD96] ISO/IEC 10746-3; ITU-T X.903 - Arquitectura, RM-ODP Reference Model for Open Distributed Processing, 1996, <http://www.uco.es/~in1rosaj/rmodp/#standards>[consulta 31 de julio de 2007]

[RROB] Rational Robot, <http://www-306.ibm.com/software/awdtools/tester/robot/> [consulta 31 de julio de 2007]

[RSA] Rational Software Architect (RSA), <http://www-128.ibm.com/developerworks/rational/products/rsa/>[consulta 31 de julio de 2007]

[RSOA] IBM - Rational Method Composer (RMC), RUP for Service-Oriented Modeling and Architecture Plug-in V2.4, <http://www.ibm.com/developerworks/rational/downloads/06/rmc_plugin7_1/#12>[consulta 31 de julio de 2007]

[RSQA05] Ulferts K., Why isn't there a RUP workflow for software quality assurance, The Rational Edge, junio 2005, <http://www.ibm.com/developerworks/rational/library/jun05/ulferts/>[consulta 31 de julio de 2007]

[RUP] IBM - Rational Unified Process (RUP), <http://www-306.ibm.com/software/awdtools/rup/>[consulta 31 de julio de 2007]

[SAAM] Kazman R., Bass L., Abowd G., Webb M., Software Architecture Analysis Method (SAAM), A Method for Analyzing the Properties of Software Architectures, SEI, <http://www.sei.cmu.edu/publications/articles/saam-metho-propert-sas.html>[consulta 31 de julio de 2007]

[SADR03] Software Architecture Document (SAD) Template, RUP, <http://www.ibm.com/developerworks/rational/products/rup/>Trial: Rational Method Composer V7.1 Work products Analisis&Design [consulta 31 de julio de 2007]

[SADS04] Software Architecture Document (SAD)Template, SEI, <http://www.sei.cmu.edu/architecture/SAD_template2.dot>actualizado en http://www.sei.cmu.edu/architecture/SAD_template_05Feb2006.dot>[consulta 31 de julio de 2007]

[SCAS] Service Component Architecture (SCA) Specification, Open Service Oriented Architecture (OSOA), <http://www.osoa.org/display/Main/Service+Component+Architecture+Specifications>[consulta 31 de julio de 2007]

[SCA05] Beisiege M. et. al, Service Component Architecture (SCA), Building Systems using a Service Oriented Architecture (SOA), joint whitepaper by BEA, IBM, Interface21, IONA, Oracle, SAP, Siebel, Sybase, noviembre 2005. <http://www.osoa.org/download/attachments/28/SCA_White_Paper1_09.pdf?version=1>[consulta 31 de julio de 2007]

[SCRU] SCRUM, <http://www.controlchaos.com/about/>[consulta 31 de julio de 2007]

[SDOS] Service Data Object (SDO) Specification, Open Service Oriented Architecture (OSOA), <http://www.osoa.org/display/Main/Service+Data+Objects+Specifications>[consulta 31 de julio de 2007]

[SEI] Software Engineering Institute (SEI), Universidad Carnegie Mellon, <http://www.sei-cmu.edu.org/>[consulta 31 de julio de 2007]

[SHFP03] H.Smith, P.Fingar, Business Process Management: The third wave, Meghan-Kieffer Press, 2003, p., ISBN 0-920-65233-9

[SHGA96] Shaw M., Garlan D., Software Architecture: Perspectives on an Emerging Discipline, Prentice Hall, 1st. edition, 1996, 242 p., ISBN 0-131-82957-2

[SHGA93] Shaw, M. and Garlan, D., An Introduction to Software Architecture. In Advances in Software Engineering and Knowledge Engineering, V. Ambriola and G. Tortora (eds.), River Edge, NJ: World Scientific Publishing Company, 1993

[SIEM95] Soni, Nord, Hofmeister, Siemens Four View model for Architecture, 1995

[SING04] Singh I., Brydon S., Murray G., Ramachandran V., Violleau T., Stearns B., Designing Web Services with the J2EE(TM) 1.4 Platform: JAX-RPC, SOAP, and XML Technologies, Java Series, Prentice Hall, 2004, 464 p., ISBN 0-321-20521-9 <https://blueprints.dev.java.net/books.html>[consulta 31 de julio de 2007]

[SOA06] Service Oriented Architecture (SOA), Object Management Group (OMG), <http://www.omg.org/soa/>[consulta 31 de julio de 2007]

[SOAC06] Sun Microsystems Inc., Sun Java Composite Application Platform Suite (CAPS), 2006, <http://www.sun.com/software/javaenterprisesystem/suites/caps_ds.pdf> [consulta 31 de julio de 2007]

[SOAE05] Eclipse Project, Eclipse SOA Tools Platform (STP), 2005, <http://www.eclipse.org/stp/index.php>[consulta 31 de julio de 2007]

[SOAF05] IBM Corporation, IBM SOA Foundation: providing what you need to get started with SOA, 2005, <ftp://ftp.software.ibm.com/software/solutions/pdfs/SOA_g224-7540-00_WP_final.pdf>[consulta 31 de julio de 2007]

[SOAJ06] Red Hat Inc., JBoss Red Hat Service Oriented Architecture Strategy, 2006, <http://www.jboss.com/pdf/rh_soa_stategy_04_07.pdf>[consulta 31 de julio de 2007]

[SOAM06] Microsoft Corporation, Enabling "Real World SOA" through the Microsoft Platform, 2006, <http://download.microsoft.com/download/b/4/d/b4db580a-0361-4907-9a6e-9d2866d8b581/Real%20World%20SOA.doc>[consulta 31 de julio de 2007]

[SOAO06] Oracle Corporation, Oracle SOA Suite, 2006, <http://www.oracle.com/technologies/soa/oracle-soa-suite-datasheet.pdf>[consulta 31 de julio de 2007]

[SOAP] Simple Object Access Protocol (SOAP), <http://www.w3.org/TR/soap/>[consulta 31 de julio de 2007]

[SOMM02] Sommerville I., Ingeniería de Software, Addison Wesley, 6ta. edición, p., 2002, ISBN 970-26-0206-8

[SPEM05] Software Process Engineering Metamodel (SPEM), Object Management Group (OMG), 2005, <www.omg.org/docs/formal/05-01-06.pdf>[consulta 31 de julio de 2007]

[SPRI] Spring Framework, <http://www.springframework.org/>[consulta 31 de julio de 2007]

[SPRO] UML 2.0 Profile for Software Services, <http://www.ibm.com/developerworks/rational/library/05/419_soa/>[consulta 31 de julio de 2007]

[STEV04] Stevens M., McGovern J., Tyagi S., Mathew S., Java Web Services Architecture, Morgan Kaufmann Publishers, The Morgan Kaufmann Series in Data Management Systems, 2003, 831p., ISBN 1-55860-900-8

[SWEB04] Software Body of Knowledge (SWEBOK), versión 2004, <http://www.swebok.org/ironman/pdf/SWEBOK_Guide_2004.pdf>[consulta 31 de julio de 2007]

[TMAD06] Delgado A., Metodología para desarrollo de aplicaciones con enfoque SOA (Service Oriented Architecture), en XXXII Conferencia Latinoamericana de Informática (CLEI'06), Sesión 4, Artículo Nro. 265, Libro de resúmenes pp. 99, Santiago de Chile, Chile, Agosto de 2006.

[UDDI] Universal Description, Discovery and Integration (UDDI), <http://www.oasis-open.org/specs/index.php#uddiv3.0.2>[consulta 31 de julio de 2007]

[UML] Unified Modeling Language Specification, Object Management Group (OMG), <http://www.uml.org/>[consulta 31 de julio de 2007]

[USDP99] Jacobson I., Booch G., Rumbaugh J., The Unified Software Development Process, Addison-Wesley Professional, 463 p., 1999, ISBN 0-201-57169-2

[WTAD05] Delgado A., Metodología para desarrollo de aplicaciones con enfoque SOA (Service Oriented Architecture), Agosto de 2005, <http://www.fing.edu.uy/~adelgado/ExtensionSOA/index.html>[consulta 31 de julio de 2007]

[W3C] Worl Wide Web Consortium (W3C), <http://www.w3.org/>[consulta 31 de julio de 2007]

[WfMC] Workflow Management Coalition (WfMC), <http://www.wfmc.org/>[consulta 31 de julio de 2007]

[WSDL] Web Services Description Language (WSDL), <http://www.w3.org/TR/ws-arch/http://www.w3.org/TR/wsdl>[consulta 31 de julio de 2007]

[WSSP] Web Services Architecture, World Wide Web Consortium (W3C), <http://www.w3.org/TR/ws-arch/>[consulta 31 de julio de 2007]

[WS-BP] Web Services Business Process Execution Language (WS-BPEL), <http://www.oasis-open.org/specs/index.php#wsbpelv2.0>[consulta 31 de julio de 2007]

[WS-CD] Web Services Choreography Description Language (WS-CDL), <http://www.w3.org/TR/ws-cdl-10/>[consulta 31 de julio de 2007]

[WS-I] Web Services Interoperability Organization (WS-I), <http://www.ws-i.org>[consulta 31 de julio de 2007]

[WS-IB] WS-I Basic Profile 1.0, <http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html>[consulta 31 de julio de 2007]

[WS-IA] WS-I Attachments Profile 1.0, <http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>[consulta 31 de julio de 2007]

[WS-PO] Web Services Policy Framework (WS-Policy),<http://www.w3.org/Submission/WS-Policy/>[consulta 31 de julio de 2007]

[WS-SE] Web Services Security (WS-Security), <http://www.oasis-open.org/specs/index.php#wssv1.1>[consulta 31 de julio de 2007]

[xUML01] Carter K., Executable UML (xUML), Supporting MDA with Executable UML, <http://www.kc.com/xuml.php>, OMG Semantics of a Foundational Subset for Executable UML Models RFP, <http://www.omg.org/cgi-bin/doc?ad/2005-4-2>[consulta 31 de julio de 2007]

[XML] eXtensible Markup Language (XML), <http://www.w3.org/TR/xml/>[consulta 31 de julio de 2007]

[XMLS] eXtensible Markup Language (XML) SCHEMA, <http://www.w3.org/XML/Schema>[consulta 31 de julio de 2007]

[XPIN] Introducción a eXtreme Programming (XP), <http://extremeprogramming.org> [consulta 31 de julio de 2007]

[XSLT] XSL Transformations (XSLT), <http://www.w3.org/TR/xslt>[consulta 31 de julio de 2007]

[XQUE] XQuery: An XML Query Language, <http://www.w3.org/TR/xquery/>[consulta 31 de julio de 2007]

[ZIMM04] Zimmermann O., Krogdahl P., Gee C., Elements of Service-Oriented Analysis and Design, An interdisciplinary modeling approach for SOA projects, <http://www.ibm.com/developerworks/webservices/library/ws-soad1/>, junio 2004 [consulta 31 de julio de 2007]