

Instituto de Computación

Facultad de Ingeniería

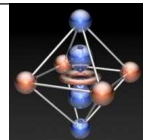
Universidad de la República

Montevideo, Uruguay

Factorización de dos modelos de proceso

Informe final

Factor.uy



$$M_{\text{Java}} + M_{\text{Gx}} = M_{\text{F}} \times (M_{\text{OO}} + M_{\text{Gx}})$$

Proyecto de Grado de Ingeniería en Computación

Estudiante: Yudith Casals

Tutor: Ing. Jorge Triñanes

Junio de 2005

Resumen

El presente documento informa los resultados del trabajo realizado en el marco de la asignatura Proyecto de Grado de la carrera de Ingeniería en Computación de la Facultad de Ingeniería de la Universidad de la República. El proyecto se desarrollo en el contexto del Grupo de Ingeniería de Software (Gris) del Instituto de Computación (InCo) de dicha institución y responde a la necesidad de estudiar el desarrollo que en los últimos años han tenido las distintas áreas que involucran la construcción de software y los procesos de desarrollo que sirven de guía para la construcción de software de calidad.

El Gris cuenta con cuatro modelos de proceso para el desarrollo de software obtenidos como resultado de trabajos anteriores. Estos modelos fueron usados en el curso Proyecto de Ingeniería de Software, dictado por el mencionado grupo, para desarrollos de software utilizando las tecnologías Java o GeneXus. El objetivo de este trabajo es unificar dos de los modelos existentes (Java y MoDSGX), factorizando la parte común a ambos para hacerla independiente del lenguaje utilizado y obtener extensiones de la factorización para desarrollos específicos con las tecnologías antes mencionadas.

La factorización consiste en construir un modelo de proceso abstracto en el que se especifiquen los roles, las actividades y los entregables en forma independiente de las tecnologías de desarrollo. El objetivo es contar con un modelo de proceso que se pueda utilizar para desarrollos sin importar la herramienta utilizada, o en su defecto, que sea fácilmente extensible para lograr un modelo de proceso orientado al desarrollo con una herramienta específica.

Tanto el modelo de proceso Factor como las extensiones para desarrollo Orientado a Objetos y GeneXus fueron creados en el transcurso del primer semestre y fueron puestos en práctica en el segundo semestre con todos los grupos que cursaron la asignatura Proyecto de Ingeniería de Software. Luego de finalizada la prueba del modelo, se construyó una nueva extensión al modelo de proceso Factor correspondiente al modelo de proceso Integrado.

Durante el transcurso de la prueba del modelo y una vez finalizada la misma se realizaron ajustes al modelo, resultantes de las sugerencias de los estudiantes y de los docentes del curso. Se realizaron, auditorías a los grupos que siguieron el proceso y se llevó adelante un seguimiento utilizando el newsgroup del curso. Se respondieron dudas del modelo planteadas por los estudiantes lo que sirvió como fuente de mejoras al mismo.

El presente informe resume el trabajo realizado, el modelo de proceso y las extensiones creadas, la experiencia y los resultados de la puesta en práctica, los ajustes realizados y por último las conclusiones del trabajo.

Palabras clave: Ingeniería de Software, Modelo de Proceso, Proceso de Desarrollo de Software, Calidad.

Tabla de Contenidos

.....INTRODUCCIÓN	
.....7	
1.1	OBJETIVOS7
1.2	RESULTADOS ESPERADOS8
1.3	ESTRUCTURA DEL INFORME8
.....CONTEXTO DEL TRABAJO	
.....9	
2.1	OBJETIVOS Y MOTIVACIONES9
2.2	METODOLOGÍA DE TRABAJO.....10
2.2.1	<i>Estudio de antecedentes en modelos de proceso</i>10
2.2.2	<i>Agilizar el Proceso</i>14
2.2.3	<i>Mejorar el proceso</i>17
2.2.4	<i>Análisis de Impacto</i>19
2.3	MODELOS DE PROCESO EXISTENTES21
2.3.1	<i>Modelo de proceso Java</i>22
2.3.2	<i>Modelo de proceso MoDSGX</i>22
2.3.3	<i>Modelo de proceso Integrado</i>23
2.3.4	<i>Modelo de proceso GXP</i>23
..... MODELO DE PROCESO PARA DESARROLLO DE SOFTWARE	
.....25	
3.1	DIMENSIÓN DEL MODELO25
3.1.1	<i>Equipo de trabajo y Roles</i>25
3.1.2	<i>Actividades y Artefactos</i>26
3.2	DIMENSIÓN TIEMPO28
3.2.1	<i>Fases</i>28
3.2.2	<i>Iteraciones y Semanas</i>29
3.3	ESTRUCTURA DEL MODELO DE PROCESO29
..... MODELO DE PROCESO FACTOR Y EXTENSIONES	
.....31	
4.1	ENFOQUE DEL MODELO31
4.2	CREACIÓN DEL MODELO31
4.2.1	<i>Consideraciones generales</i>31
4.2.2	<i>Modelo de proceso Factor (F)</i>34
4.2.3	<i>Extensión para desarrollo Orientado a Objetos (OO)</i>41
4.2.4	<i>Extensión para desarrollo con GeneXus (Gx)</i>42
4.2.5	<i>Extensión para desarrollo Integrado (I)</i>42
4.2.6	<i>Extensión para desarrollo Integrado Orientado a Objetos (IOO)</i>42
4.3	DEFINICIÓN DEL MODELO DE PROCESO42
4.3.1	<i>Sitio Web del modelo</i>43
.....PRUEBA DEL MODELO DE PROCESO	
.....47	
5.1	PRESENTACIÓN A LOS INVOLUCRADOS47
5.2	AUDITORÍAS48
5.2.1	<i>Auditoría de Fase Inicial</i>48

5.2.2	<i>Auditoría de Fase de Elaboración</i>	49
5.3	INDICADORES	49
5.3.1	<i>Esfuerzo</i>	49
5.3.2	<i>Tamaño del Producto</i>	50
5.4	RESULTADOS.....	51
5.4.1	<i>Satisfacción de los grupos</i>	51
5.4.2	<i>Comparación con experiencias anteriores</i>	56
5.4.3	<i>Satisfacción del cliente</i>	62
5.5	AJUSTES REALIZADOS.....	63
..... CONCLUSIONES Y TRABAJOS FUTUROS		
.....		66
..... CONTENIDO DEL CD		
.....		68
..... REFERENCIAS BIBLIOGRÁFICAS		
.....		70

Capítulo 1

Introducción

El Grupo de Ingeniería de Software [\[1\]](#) del Instituto de Computación [\[2\]](#) de la Facultad de Ingeniería de la Universidad de la República cuenta con cuatro modelos de proceso para desarrollo de software elaborados con el objetivo de ser utilizados en el curso Proyecto de Ingeniería de Software [\[3\]](#) para desarrollar software de mayor calidad utilizando las tecnologías Java y GeneXus.

Tres de los modelos existentes fueron creados basándose fuertemente en Rational Unified Process [\[4\]](#) (RUP), lo que hace que tengan una fuerte base común en la mayoría de sus actividades, roles y entregables mientras que el cuarto de los modelos fue creado inspirándose en Extreme Programming [\[5\]](#) (XP). Los modelos son ajustados de acuerdo a las experiencias obtenidas en la puesta en práctica de los mismos. Cada modelo de proceso se presenta tanto en formato texto como en formato web lo que duplica el material existente.

Motiva este proyecto la necesidad de construir un único modelo de proceso independiente de la tecnología de desarrollo de manera que facilite y reduzca el trabajo necesario para realizar el mantenimiento y los ajustes requeridos. Se busca además obtener extensiones del modelo que permitan el desarrollo en contextos más específicos. Una motivación adicional es la de actualizar el modelo de proceso Java atendiendo a las novedades introducidas con la nueva versión de RUP recientemente disponible.

1.1 Objetivos

Dos de los modelos de proceso con que cuenta el Gris, Java [\[6\]](#) e Integrado [\[7,8\]](#), fueron pensados para trabajar con Java como herramienta de desarrollo mientras que los otros dos, MoDSGX [\[9\]](#) y GXP [\[10\]](#), fueron pensados para trabajar con GeneXus. En este contexto los objetivos del presente proyecto son:

- Disponer de un modelo de proceso independiente de las herramientas de desarrollo que incorpore lo mejor del modelo Java y del modelo MoDSGX.
- Disponer de agregados al modelo de proceso anterior que permitan su particularización para contextos de uso específicos.

En otras palabras, el objetivo del proyecto es unificar los modelos de proceso Java y MoDSGX, en uso en el curso Proyecto de Ingeniería de Software, generando un único modelo de proceso. Asimismo, este modelo debe permitir la creación de extensiones para obtener modelos de proceso particulares, tanto para el desarrollo Orientado a Objetos como para el desarrollo con GeneXus. El modelo de proceso junto a sus extensiones se debe poner a prueba durante el segundo semestre de 2004 con grupos de proyectos de dicho curso, evaluar la experiencia y ajustar el proceso en consecuencia. Paralelamente, se

busca incorporar mejoras al proceso en términos de: descripción de actividades, estructura y contenido de plantillas, agilidad del desarrollo, verificación, entre otras.

Un objetivo adicional del proyecto en su planteo original era modelar el proceso para aquellos casos en los que se desea extender o modificar un producto ya existente. Este objetivo fue eliminado del alcance de este proyecto ya que este punto implicaba la creación de un modelo de proceso completamente nuevo que requería de un tiempo de estudio e investigación bastante grande que sumado al tiempo necesario para crear el modelo de proceso Factor y las extensiones, resultaba demasiado extenso para la duración de este proyecto y los recursos humanos con los que contamos ya que originalmente el proyecto estaba pensado para ser llevado a cabo por dos estudiantes.

1.2 Resultados esperados

En pro de los objetivos del proyecto presentados anteriormente, los resultados esperados se enumeran a continuación.

- Estudio de antecedentes análogos en modelos de proceso.
- Comparación de los modelos Java y MoDSGX e identificación en este último de los elementos que son específicos a GeneXus.
- Factorización de los elementos comunes a ambos modelos.
- Posibilidad de derivar mediante agregados modelos específicos para Java, GeneXus y desarrollo Integrado.
- Evaluación de la puesta en práctica de esos modelos.
- Ajustes adicionales.

1.3 Estructura del informe

El presente informe tiene la siguiente estructura. El capítulo 2 presenta el contexto del trabajo realizado. Se profundiza en los objetivos y motivaciones del proyecto, se habla de la metodología de trabajo utilizada en este proyecto y se describen resumidamente los modelos de proceso existentes en el Gris. El capítulo 3 explica qué es un modelo de proceso y cómo se especifica mientras que el capítulo 4 presenta el modelo de proceso Factor, su creación y los aspectos generales del mismo, las distintas áreas que hacen al modelo, su definición y el sitio web creado. Se describen además las distintas extensiones creadas para desarrollo con Java, GeneXus e Integrado. Luego el capítulo 5 presenta la prueba del modelo de proceso, es decir, la puesta en práctica del mismo. Se expone la presentación realizada, el seguimiento, las auditorías, los indicadores, los resultados, y los ajustes. El capítulo 6 resume las conclusiones y sugiere trabajos futuros.

Capítulo 2

Contexto del trabajo

Este capítulo describe el contexto en que se realiza el presente trabajo, describiendo mayormente los antecedentes en modelos de proceso, así como aquellos que han sido elaborados en el Grupo de Ingeniería de Software, su posible mejora y agilización. La adaptación y factorización de estos procesos conforman los principales objetivos de este trabajo.

Este capítulo se estructura de la siguiente manera. La sección 2.1 profundiza en los objetivos y motivaciones del proyecto. Luego, la sección 2.2 describe la metodología de trabajo utilizada. Presenta antecedentes en modelos de proceso, comparaciones y adaptaciones para su mejora y agilización, analizando por último su impacto en este trabajo. Finalmente, la sección 2.3 detalla los modelos de proceso existentes en el Gris.

2.1 Objetivos y Motivaciones

En esta sección se exponen los elementos que motivaron este proyecto relacionándolos con los objetivos planteados en la introducción.

El curso Proyecto de Ingeniería de Software pertenece al octavo semestre de la carrera y pretende afirmar y profundizar los conocimientos de Ingeniería de Software, contrastarlos con su aplicación práctica e integrarlos con conocimientos de otras materias. La metodología de trabajo está basada en un proyecto guiado y controlado, sometido a restricciones análogas a las que son comunes en la industria lo que posibilita una reflexión sobre los conocimientos teóricos a la luz de su aplicación práctica. Al comienzo del curso se forman equipos de trabajo que varían entre 10 y 13 personas dependiendo de las características de los proyectos que se van a llevar adelante, de los procesos que se van a utilizar, de la cantidad de estudiantes inscriptas al curso y de otras posibles variables. A estos equipos de trabajo se les asigna un proyecto, con un cliente y un director determinado y se les asigna también el proceso que deben seguir.

Los modelos de proceso Java y MoDSGX fueron creados en el año 2000 y 2002 respectivamente. Ambos basaron sus principios en RUP y han agregado prácticas específicas atendiendo al contexto de desarrollo. Estos modelos fueron desarrollados pensando en el uso de determinadas herramientas de desarrollo, Java y GeneXus respectivamente. Además fueron pensados para ser aplicados en el curso Proyecto de Ingeniería de software por lo que son fijos ciertos aspectos como el tiempo y el número de integrantes de cada equipo.

Como resultado de trabajos de investigación adicionales de algunos integrantes del Gris surgieron el modelo de proceso GXP y el Integrado en el año 2003 y fueron puestos en práctica ese mismo año. Estos modelos tienen fijados los mismos parámetros que los anteriores pero el contexto de uso es sensiblemente diferente ya que GXP es una

adaptación de Extreme Programming para desarrollo con GeneXus mientras que el Integrado fue pensado como una extensión del modelo Java para permitir desarrollos integrados de múltiples grupos de trabajo.

Cada uno de estos modelos se presenta en un sitio web que describe en detalle todos sus aspectos, roles, actividades, artefactos, entre otros. Además, algunos de ellos también se encuentran descritos en documentos de texto, que aunque puede resultar más cómodo de leer, multiplica el material existente y por lo tanto multiplica también la posibilidad de inconsistencias y el trabajo de mantenimiento.

Lo que se busca entonces es unificar los procesos Java y MoDSGX en un modelo único e independiente del lenguaje de desarrollo que redunde en mejoras para el proceso incorporando en las actividades, roles, y entregables las mejores prácticas de cada uno de ellos. Un objetivo buscado y consecuencia adicional de la integración de los modelos es la reducción del material existente y por lo tanto se espera con esto reducir el trabajo de mantenimiento.

La integración de los modelos debe resultar en una mejora de la calidad del proceso por incorporar las mejores prácticas de cada uno. Además de lo anterior se realizarán tareas que redundarán en una mejora de la calidad del proceso como ser la incorporación de actividades, la mejora en las especificaciones de otras ya existentes, la incorporación de algunas modificaciones introducidas por RUP, la mejora de las especificaciones y descripciones de las responsabilidades de los roles del proceso, la mejora de la agenda propuesta y una mejora de las plantillas existentes. Como objetivo adicional se busca la posibilidad de lograr extensiones que resulten en modelos de proceso para desarrollos con Java, GeneXus e Integrado.

2.2 Metodología de Trabajo

La metodología de trabajo seguida busca obtener un modelo de proceso mejorado con respecto a los ya existentes en el Gris. Para ello, se estudiaron antecedentes de modelos de proceso existentes fuera del grupo, incluyendo el propio RUP, en el que fueron basados los modelos de proceso Java y MoDSGX. Además se estudiaron materiales en la búsqueda de lograr un proceso más ágil y mejorar algunas áreas específicas concernientes a la integración y a la especificación y descripción de la liberación.

2.2.1 Estudio de antecedentes en modelos de proceso

Se presenta a continuación una descripción de las principales características de algunos de los modelos de proceso disponibles en el mercado, a saber RUP, XP y DSDM. Estos modelos han servido como base para los modelos de proceso desarrollados por el grupo Gris, presentados en la Sección 2.3, y para el modelo de proceso Factor presentado en el Capítulo 4.

Rational Unified Process (RUP)

RUP es un modelo de proceso creado en 1998 por J. Rumbaugh, G. Booch y I. Jacobson para la empresa Rational Corporation. Surge como la fusión de los enfoques de desarrollo particulares de estos autores, Object Modelling Technique, Booch Method y Objectory respectivamente, incorporando prácticas de desarrollo de diferentes compañías de software. Poco tiempo después de la aparición de RUP, los mismos autores presentaron una versión

pública del modelo que fue denominada Unified Process [\[11\]](#) (UP). Esta versión data de 1999, estando desactualizada hoy en día respecto a la última versión de RUP de 2003.

RUP provee un enfoque disciplinado para asignar tareas y responsabilidades en una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que cumpla con las necesidades de los usuarios finales, dentro de un cronograma y presupuesto predecible. Permite un vocabulario común entre distintos equipos de desarrollo y es considerado una base para la creación de nuevos procesos y la creación de procesos para contextos de uso específicos. RUP es un proceso que puede ser utilizado en una amplia gama de proyectos, de diferentes áreas de aplicación, diferentes tipos de organizaciones y diferentes tamaños, esto se debe a que reúne muchas de las buenas prácticas utilizadas en el desarrollo de software de la actualidad.

Tres características definen a RUP y sus especializaciones ya que representan los principios fundamentales de su enfoque. Primero, RUP esta dirigido por casos de uso. Los casos de uso son un mecanismo ampliamente usado para la captura de requerimientos funcionales del sistema mediante la descripción de la interacción actor-sistema. Aparte de ello, y más importante desde la perspectiva del proceso, los casos de uso son el punto de partida de las actividades de otras disciplinas como ser Análisis y Diseño, Implementación, y Testeo. Además, los casos de uso son usados para planificar, estimar, controlar y administrar estas actividades. Segundo, RUP esta centrado en la arquitectura. La arquitectura de software refiere a las decisiones significativas sobre la organización del sistema de software, la elección de los elementos estructurales y sus conexiones, su composición y el estilo general que guía esta organización. Las primeras fases del proceso refieren a la construcción de una arquitectura a partir de los requerimientos no funcionales críticos y de los casos de uso más significativos. Las siguientes fases del proceso refieren a la construcción sobre esta arquitectura del resto de las funcionalidades. Tercero, RUP es iterativo e incremental. El desarrollo se divide en iteraciones que pueden verse cada una como un mini-proyecto. Cada iteración resulta en un incremento que no es necesariamente aditivo sino que puede, por ejemplo, reemplazar un diseño superficial por uno más detallado o sofisticado, lo que ocurre principalmente en las primeras iteraciones. Las iteraciones se refieren a los pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto.

Otras características destacables de RUP son su adaptabilidad a diferentes organizaciones y proyectos y su constante evolución. Asimismo, el proceso es repetible y permite la realización de mediciones y estimaciones de costos, avance y tiempo.

Extreme Programming (XP)

XP fue creado en 1996 por K. Beck como resultado de un proyecto en DaimlerChrysler en el que aplicó nuevos conceptos en el desarrollo de software. La base fueron sus años de experiencia como desarrollador de software orientado a objetos y en particular su trabajo con W. Cunningham utilizando el lenguaje Smalltalk.

XP consiste de un conjunto de valores, principios y prácticas para desarrollar rápidamente software de buena calidad que provea el mayor valor para el cliente en la manera más rápida posible. XP fue creado en respuesta a dominios de problemas cuyos requerimientos son cambiantes, por ejemplo en contextos donde los clientes no tienen una idea clara de lo que el sistema debe hacer o donde la funcionalidad puede cambiar cada pocos meses.

XP esta pensado para pequeños grupos de desarrolladores de entre dos y doce personas. El equipo de XP no solo incluye a los desarrolladores sino también a los administradores y

clientes. Esto se debe a que el enfoque de XP radica en una constante interacción con el cliente, negociación del alcance y del cronograma, y en test funcionales orientados por el cliente.

Los cuatro valores fundamentales buscados por los programadores que utilizan XP son la comunicación, la sencillez, la retroalimentación y el valor. K. Beck encontró que existen cuatro dimensiones en las que se puede mejorar cualquier proyecto de software:

- Se necesita mejorar la comunicación.
- Se necesita buscar sencillez.
- Se necesita obtener retroalimentación de cuan bien se esta haciendo el trabajo.
- Se necesita siempre avanzar con valor.

XP se caracteriza por 12 prácticas fundamentales:

- El juego de planificación: Determinar rápidamente el alcance para la próxima liberación combinando las prioridades del negocio y las estimaciones de los técnicos. Cuando la realidad sobrepasa el plan, modificar el plan.
- Pequeñas liberaciones: Poner un sistema simple rápidamente en producción, luego liberar versiones nuevas en ciclos cortos.
- Metáfora: Guiar todos los desarrollos compartiendo una historia simple de como el sistema trabaja como un todo, llamadas historias de usuario.
- Diseño simple: El sistema debe ser diseñado tan simple como sea posible en un momento dado. La complejidad extra es sacada tan pronto como es descubierta.
- Pruebas: Los programadores continuamente escriben testeos unitarios, que deben correr sin defectos para poder continuar desarrollando. Los clientes escriben pruebas para demostrar que los requerimientos se han completado.
- Refactoring: Los programadores reestructuran el sistema sin cambiar su comportamiento para sacar duplicados, mejorar la comunicación, simplificar o agregar flexibilidad.
- Programación por pares: Todo el código es producido con dos programadores en una máquina.
- Propiedad colectiva: Cualquiera puede cambiar cualquier código en cualquier momento.
- Integración continua: Integrar y construir el sistema varias veces al día, cada vez que una tarea es completada.
- 40 horas semanales: La regla es no trabajar más de 40 horas a la semana. En caso de trabajar extra, nunca hacerlo dos semanas seguidas.
- Cliente en el lugar: Incluir un cliente (usuario) real en el equipo, disponible todo el tiempo para contestar dudas.
- Estándares de codificación: Los programadores escriben todo el código de acuerdo a reglas que acentúan la comunicación a través del código.

Dynamic Systems Development Method (DSDM)

DSDM [12] es una metodología nacida en el Reino Unido en 1994, siendo de gran aceptación en ese país para el desarrollo rápido de aplicaciones de software. DSDM es un marco de trabajo desarrollado y mantenido por el DSDM Consortium, una organización sin fines de lucro. Fue diseñado para ser independiente de cualquier tipo de proyecto particular o método de desarrollo.

La idea fundamental – y original – de DSDM es que, en lugar de partir de un conjunto fijo de funcionalidades para un producto a desarrollar y a partir de ahí ajustar los recursos, DSDM procede en forma inversa fijando primero el tiempo y los recursos con que se cuenta y ajustando en función de ellos la funcionalidad a ser entregada con el producto.

El proceso de desarrollo de software con DSDM se divide en cinco fases fundamentales:

- Estudio de Factibilidad
- Estudio del Negocio
- Iteración Funcional del Modelo
- Iteración del Diseño y de la Construcción
- Implementación

Las dos primeras fases son secuenciales, y se ejecutan una única vez durante el proceso mientras que las tres últimas son iterativas e incrementales.

Existen además dos fases complementarias que corresponden a la fase previa al inicio del proyecto y a la fase posterior a la finalización del mismo. Ellas son la fase de Pre-proyecto y la fase de Post-proyecto. A la salida de la primera se obtiene la certeza de que el proyecto es correcto mientras que la segunda se corresponde con el cierre del proyecto disolviendo el equipo de desarrollo y comenzando con el mantenimiento y monitoreo del sistema.

DSDM define quince roles diferentes para usuarios y desarrolladores. A diferencia de otras metodologías, pone énfasis en una clasificación de las distintas responsabilidades de los usuarios, contando con roles para ellos como ser, el Sponsor Ejecutivo, el Usuario Embajador, el Usuario Asesor y el Visionario. El primero representa a una persona de la organización cliente con responsabilidades financieras y jerárquicas sobre el proyecto, y con poder de decisión. El segundo juega un importante papel en lo referente al feedback de los usuarios así como de mantener una visión global sobre las funcionalidades requeridas al sistema. Por otro lado, cada Usuario Asesor representa un punto de vista importante sobre las funcionalidades. Por último, el Visionario es aquel usuario con la más precisa percepción de los objetivos del negocio, del proyecto y del sistema.

DSDM se basa en nueve prácticas o principios:

- Es imperativo que los usuarios se involucren activamente.
- Los equipos de DSDM deben estar facultados para la toma de decisiones.
- El foco está puesto en la entrega frecuente de productos.
- El ajuste a los objetivos del negocio es el criterio esencial para la aceptación de los entregables.
- El desarrollo iterativo e incremental es necesario para converger a una solución acertada del negocio.

- Todos los cambios realizados durante el desarrollo son reversibles.
- Los Requerimientos son congelados a alto nivel.
- El testing es integrado en todo el ciclo de vida.
- Es esencial un enfoque de colaboración y cooperación por parte de todos los involucrados.

Por último, debe notarse que DSDM y XP también tienen aspectos en común y aspectos que se complementan. Ambos necesitan del compromiso de los usuarios, aunque, en DSDM es bastante más significativo que en XP. Ambos realizan testing frecuente, entregas rápidas y cuentan con mecanismos rápidos de feedback. Por otro lado, XP presenta carencias en la gestión del proyecto y es difícilmente escalable mientras que DSDM presenta carencias en lo que hace a buenas prácticas de ingeniería y técnicas de programación.

La combinación de DSDM con XP permite ganar los beneficios del Modelo de Gestión de Proyectos de DSDM y enfocarse en el negocio con las prácticas de alta eficiencia y calidad de XP. A esta combinación se le ha denominado Enterprise XP o EXP. Ambas metodologías son compatibles ya que las técnicas en XP pueden utilizarse con el marco de trabajo controlado de DSDM, y los productos en DSDM pueden construirse utilizando técnicas de programación de XP.

2.2.2 Agilizar el Proceso

Generar el modelo de proceso Factor implica una revisión completa de los modelos de proceso existentes en el Gris, y así, una oportunidad para mejorarlos. Uno de los puntos críticos en este sentido es la agilización del modelo de proceso, la cual busca reducir el tiempo y el esfuerzo de desarrollo, manteniendo la calidad de los productos construidos. Por este motivo, se han estudiado diferentes trabajos que buscan integrar las técnicas de varios de los modelos presentados en la Sección anterior. Se presentan a continuación aquellos puntos de tales trabajos que son pertinentes a este proyecto, en la línea de agilización de los procesos.

RUP vs. XP

En general, RUP aparenta ser un proceso “pesado” mientras que XP parece ser “liviano”. Esta apreciación es relativa tal como se describe en RUP vs. XP [\[13\]](#) y Una comparación de RUP y XP [\[14\]](#). Se presenta a continuación las consideraciones más relevantes al respecto.

RUP y XP provienen de filosofías diferentes. RUP puede aplicarse a cualquier proyecto de desarrollo de software, esperándose de la organización desarrolladora que realice una especialización del mismo. Por otro lado XP es un proceso más restringido que necesita agregados para ajustarse a un proyecto completo de desarrollo. Estas diferencias sugieren, de alguna forma, la percepción que tiene la comunidad desarrolladora de software; los desarrolladores de grandes sistemas ven a RUP como la respuesta a sus problemas mientras que la comunidad desarrolladora de pequeños sistemas ve a XP como la solución a los suyos.

RUP es un producto y por lo tanto pretende ser completo, este es el motivo por el cual presenta detalladamente las disciplinas, las actividades, los artefactos y los roles. Presenta además plantillas para muchos de los artefactos que son documentos para que sirvan como guía al equipo que está siguiendo el proceso. Aquí nos enfrentamos a la primera impresión de RUP, al ser un producto completo se presenta bastante más grande que XP.

Otra de las características de RUP es que es “Centrado en la Arquitectura” lo que constituye un pilar fundamental a la hora de construir productos de gran tamaño, productos que XP no podría manejar. XP está pensado para grupos de trabajo de dos a doce personas lo que limita el tamaño de los proyectos que enfrenta. Podría pensarse que un proyecto de gran tamaño puede ser dividido en proyectos menores donde sería posible aplicar XP pero es aquí donde la arquitectura juega un papel fundamental. Es prácticamente imposible llevar adelante proyectos menores y luego integrarlos con éxito si no se tuvo en cuenta una arquitectura común. La arquitectura para RUP no es simplemente infraestructura como lo es para XP, sino que es una solución completa desde determinadas perspectivas y a un nivel apropiado de abstracción.

RUP trata de eliminar los riesgos y ese es el motivo de hacer un ejecutable de la arquitectura. Con el ejecutable se pueden reducir los riesgos asociados a los requerimientos no funcionales así como validar los mismos con el cliente. El enfoque de XP hace que la arquitectura surja del refactoring del código. Algunos tipos de sistemas pueden no ser sensibles a este enfoque, resultando en un gran riesgo comparado con atacar la arquitectura al comienzo como sugiere RUP.

La visión “pesada” de RUP es alentada por el numeroso conjunto de artefactos que sugiere realizar. En cambio, RUP plantea además que se elija qué actividades y qué artefactos aplican a cada proyecto particular. No se sugiere la realización de todos los artefactos, ni que se elaboren todas las partes de aquellos que son seleccionados. La clave está en tener la experiencia y conocimiento suficiente como para hacer una adecuada instanciación o realización que no resulte tan “pesada”.

Otro aspecto relacionado con el tema de los artefactos es que XP tiene de por sí un número bastante menor de artefactos. Esto se debe a que maneja un subconjunto de las disciplinas que maneja RUP y a menor escala. El alcance de XP es menor que el de RUP ya que está orientado principalmente a satisfacer los requerimientos basándose en las personas y no tanto en los artefactos. De todas formas, la cantidad de artefactos de XP no es despreciable aunque parece menor por no estar especificados con claridad en ningún documento.

No siempre es posible modelar los Requerimientos y el Diseño del sistema con las historias de usuarios. El comportamiento deseado de sistemas más grandes y complejos puede ser muy difícil de articular sin un enfoque sistemático como los Casos de Uso. En estos casos, tampoco será posible confiar en las conversaciones entre el cliente y los desarrolladores para elaborar historias de usuarios complejas en forma consistente. Los artefactos que RUP describe en las disciplinas de Requerimientos y Análisis y Diseño le permiten manejar la variedad, tamaño y complejidad de estos sistemas.

RUP permite una mayor formalidad de la disciplina de Gestión de Proyecto para aquellos proyectos que así lo requieran, aumentando consecuentemente la cantidad de artefactos propuestos. Nuevamente, estos artefactos no son necesarios en su totalidad en proyectos pequeños o menos formales.

Algunas diferencias adicionales entre RUP y XP que contribuyen también a la visión “pesada” que se tiene del primero son:

- Las actividades de XP se asemejan más a las disciplinas de RUP, es decir que el grado de granularidad es sensiblemente diferente, siendo mucho más detallado el proceso de RUP.

- Por la escala de los proyectos que puede manejar XP es sencillo lidiar con la Configuración y el Control de Cambios, disciplina que RUP debe manejar con detalle para lograr un proyecto exitoso.

Un último aspecto a destacar es que hay algunas de las prácticas de XP que pueden ser utilizadas en RUP y otras ya están incorporadas.

dX

El Proceso dX [13] fue creado por R. Martin como una versión ágil de RUP. XP y RUP no son necesariamente mutuo excluyentes; utilizando técnicas de ambos métodos se puede derivar un proceso que permite obtener software de buena calidad y en corto tiempo. dX se proclama como “seguidor” u “obediente” de RUP ya que fue construido como una instancia del mismo. Por otro lado, dX esta alineado con XP, dX fue diseñado para contextos en que debe utilizarse RUP pero que se desee hacer uso de las prácticas ágiles de XP.

dX no explicita la utilización de UML ni de ningún otro lenguaje de modelado, el único artefacto intermedio que usa son “use case cards”. Esto no implica que si se usa dX no puede usarse UML cuando se necesita especificar un modelo o testear algunas afirmaciones, lo que implica es que dX no dice cuando dichos modelos deben ser creados.

Exige además la presencia constante de un usuario representante que será además quien elabore, con respaldo de los desarrolladores, los Casos de Uso. Este usuario además será el encargado de priorizar los Casos de Uso y en base a la estimación de esfuerzo y tiempo realizada por los desarrolladores será quien decidirá cuales serán implementados en cada iteración.

Dado que dX no sugiere la elaboración de artefactos de análisis y diseño podría concluirse que no se realizan actividades de esta disciplina. Sin embargo, esta conclusión no es acertada ya que dX propone realizar frecuentes sesiones de diseño de donde salen las decisiones. Las mismas son documentadas típicamente en “index cards”, descartadas generalmente una vez que fueron llevadas a código.

El proceso depende muy fuertemente de la habilidad de los miembros del equipo para comunicarse. Por lo tanto se debe proveer un ambiente amplio para permitir el trabajo y el estrecho contacto de cada miembro del equipo y del usuario representante.

Si se necesita más formalidad, se puede extender dX agregándole adecuada y cuidadosamente actividades y artefactos. Sin embargo esto debería hacerse teniendo el feedback del proceso como para saber si lo que se está agregando es beneficioso o perjudicial.

XUP

XUP [15] es un proceso que combina RUP y XP, junto al proceso de desarrollo propuesto por Microsoft, llamado Microsoft Solutions Framework [16] (MSF). Estos procesos son en su mayor parte compatibles aunque cada uno pone énfasis en distintos aspectos de la conducción y prácticas del desarrollo. Cada una es más efectiva si se aplica en la escala de desarrollo que pretende.

MSF y XP encajan muy elegantemente pero existen ciertos conflictos entre MSF y RUP que Rational está trabajando para resolver. En aquellos puntos en los que hay

desencuentros, XUP utiliza prácticas que se ajusten a los principios de su proceso de desarrollo.

El proceso sugerido por XUP busca utilizar solamente aquellas tareas que se necesitan para cada proyecto particular. El mismo se organiza como se describe a continuación. Las unidades de trabajo más pequeñas son las “Tareas de Ingeniería”, son los bloques fundamentales del proceso que se espera que los desarrolladores realicen y mantengan. Estas tareas son derivadas de las “Historias de Usuario” que representan funcionalidades requeridas por el cliente de las cuales el alcance es determinado por los desarrolladores. El proceso sigue iteraciones de tres semanas en las que se construye un conjunto de “Historias de Usuario”. Así, el proceso de desarrollo se conforma en tres niveles, el proceso se realiza en iteraciones, donde cada iteración ataca un conjunto de “Historias de Usuario”, las cuales se dividen en “Tareas de Ingeniería”.

Con respecto a las fases, XUP adopta las mismas que las propuestas por RUP, aunque varía su duración. XUP invierte menos tiempo en la fase de Elaboración mientras que extiende la fase de Construcción.

Tanto MSF como XP hacen énfasis en la igualdad de los miembros del equipo, sin hacer uso de jerarquías. Estos procesos especifican diferentes roles, pero cualquier miembro del equipo puede ocupar estos roles cuando se requiera. De todas formas, antes de que un proyecto comience, se realiza un Plan de Proyecto para asignar a cada miembro la responsabilidad sobre un conjunto de actividades. Las mismas involucran tareas de gestión del producto, del cronograma, el desarrollo, el testing, la capacitación de usuarios, gestión logística, seguimiento y entrenamiento. Cada miembro es responsable del éxito del proyecto, pero estas asignaciones hacen además que cada miembro se responsabilice de que el equipo entero lleve adelante las tareas involucradas en sus actividades.

2.2.3 Mejorar el proceso

La agilización del modelo de proceso es una de las direcciones a atacar para su mejora. A su vez, otros aspectos de los modelos son susceptibles a mejoras. Como resultado del curso Taller de Gestión de Software, así como de la puesta en práctica de los modelos y las tecnologías por la comunidad de desarrolladores, han surgido trabajos que estudian y proponen mejoras a los modelos de proceso. En función de ellos, se identificaron tres direcciones donde, al incorporar parte de esas sugerencias, podría lograrse mayores mejoras al proceso. Se presenta a continuación una breve descripción de estas líneas de trabajo.

Especificación y Descripción de la Liberación

El trabajo Especificación y Descripción de la Liberación [\[17\]](#) presenta cómo y por qué especificar y describir una liberación. Se entiende por liberación a una versión, parcial o total, del producto en desarrollo. Dicho trabajo propone, para cada iteración, elaborar al comienzo una Especificación de la liberación y al final una Descripción de la liberación.

El artefacto de Especificación de la liberación se presenta como un Plan de la Iteración “concentrado”. El Plan de la Iteración es un artefacto del modelo que se obtiene como salida de la actividad Reunión de Equipo. Contiene una planificación de las actividades y los responsables de las mismas, los hitos y artefactos entregables, y los criterios de evaluación para la Iteración que se está planificando. El artefacto Especificación de la liberación contiene el código y artefactos a liberar, objetivos, criterios a cumplir y plan de cumplimiento, así como también un plan de demostración y procedimientos de demostración.

Luego de la liberación debemos comparar lo que liberamos con lo que se había especificado. La Descripción de la liberación se presenta en el mismo trabajo y plantea que una descripción de una liberación contiene qué fue lo que se liberó y en que línea base. Contiene también las restricciones y limitaciones del producto liberado, los resultados de la evaluación, las acciones a tomar y lo aprendido, que se deben tener en cuenta para próximas liberaciones. La línea base está formada por los ítems de configuración, es decir los entregables más relevantes del proyecto, que juntos constituyen el sistema.

En resumen, la Especificación de la liberación puede servir para concentrarse durante la iteración en un solo documento y a su vez ser un referente para todos los involucrados. Por otro lado la Descripción de la liberación presenta la comparación entre lo planificado y lo realizado realmente en la liberación. Los autores de La Especificación y Descripción de la Liberación presentan guías para obtener los artefactos Especificación y Descripción así como también ejemplos de los mismos.

Integración en los Proyectos de Ingeniería de Software

El trabajo Integración en los Proyectos de Ingeniería de Software [\[18\]](#) presenta un estudio de la realización de la actividad de Integración en el contexto del curso Proyecto de Ingeniería de Software. El objetivo de los autores en este trabajo era conocer cómo se estaban realizando las actividades de integración en los grupos, saber qué aportaban los modelos de proceso a estas actividades y proponer mejoras en caso de ser necesario.

El modelo de proceso empleado por los grupos proveía lineamientos muy generales en lo referente a la Integración. Este hecho se ve reflejado en que los grupos eran ineficientes a la hora de integrar, la tarea tenía gran complejidad, se invertía excesivo esfuerzo para lograrla, y los grupos estaban desorganizados en la tarea. Los problemas encontrados radicaban en la planificación, en la integración misma y en su verificación.

Los autores detectaron, mediante cuestionarios a los grupos, algunas buenas prácticas que adoptaban los grupos para realizar la Integración. Entre estas prácticas se encuentran la elaboración de grafos de precedencia entre implementación e integración, la subida al repositorio de aquellas componentes comunes que tienen funcionalidad verificable y la verificación de los mismos antes de la integración.

Como soporte a estos problemas y apoyados en RUP surgió el rol de Integrador como persona idónea en el área, con conocimientos de la arquitectura, del proceso de integración, con manejo del repositorio y conocimiento de la plataforma de desarrollo. Surgió también una secuencia de actividades para lograr una integración efectiva. Los autores describen las actividades de integración propiamente dichas, anticipan los problemas inherentes a dichas actividades y por último sugieren buenas prácticas.

GXPoints

Uno de los aspectos que se plantea la Ingeniería de Software es cuán medible es el software y qué mecanismo utilizar para hacerlo. Diversas unidades de tamaño han sido propuestas, como ser líneas de código (lines of code), puntos funcionales (functional points), puntos de objetos (object points) y puntos de aplicación (application points), de las cuales solo las tres últimas podrían ser aplicables a GeneXus. En la comunidad de desarrolladores GeneXus es muy utilizada la cantidad de objetos (transactions, work panels, etc.) como unidad de medida. Este mecanismo presenta el problema de que comparar por la cantidad de objetos sin considerar su tamaño no brinda una medición certera. Por otro lado, el clasificar los

objetos según su tamaño, si bien ayuda, sería un criterio subjetivo; por lo que tampoco proporcionaría una medición eficaz.

Se plantea la necesidad de una unidad de medida única y una definición de la categoría de tamaño en forma objetiva. El problema que se presenta es que la programación de los distintos tipos de objetos GeneXus difiere, por lo que resulta adecuado medir el tamaño de los objetos programados. Tampoco resultaría adecuado medir el tamaño de los objetos generados, es decir, del programa fuente en lenguaje destino, ya que no es de utilidad comparar código en distintos lenguajes.

Los autores de GXPoints [\[19\]](#) proponen una métrica para las aplicaciones desarrolladas en GeneXus. Esta plataforma provee generadores de sus objetos (agrupados en una Base de Conocimiento) a diferentes lenguajes destino como ser Visual Basic, Java, RPG, Cobol, etc. Como paso intermedio de esta generación se crea una versión de los objetos llamada SPEC, la cual será transformada luego al lenguaje destino. Un SPEC es una especificación abstracta, integra toda la funcionalidad requerida, y es uniforme para los diferentes tipos de objetos GeneXus. Los GXPoints se definen como el tamaño en líneas de especificación generadas por GeneXus en cientos y resulta sencillo implementar una herramienta de conteo para los mismos.

Los autores realizaron un estudio con diversas Bases de Conocimiento y obtuvieron mediciones basándose en las cuales realizaron una categorización en cuatro rangos de tamaño para los objetos, Muy Grande, Grande, Mediano y Pequeño, de acuerdo a la cantidad de GXPoints que aportan.

Los GXPoints permiten la comparación de Bases de Conocimiento y la categorización de los objetos. Permite además realizar estimaciones y mediciones de costo y esfuerzo. GXPoints parece ser una unidad “natural” de medida de tamaño para desarrollos GeneXus aunque su uso aún no se ha difundido. Permite planificar y hacer un seguimiento del proyecto además de servir como base para realizar estimaciones y mediciones. Permite elaborar una herramienta sencilla para el conteo de los mismos.

2.2.4 Análisis de Impacto

La elaboración del modelo de proceso Factor y sus extensiones implica, como dijimos antes, una revisión de los modelos de procesos existentes en el Gris, y por consiguiente una oportunidad para mejorar. En esta línea se estudiaron los antecedentes a estos modelos y trabajos relacionados a la agilización y mejora de los procesos, todo ello discutido en las secciones anteriores. Para concluir, en esta Sección se resume el impacto que tuvo este estudio en la construcción del modelo de proceso y sus extensiones.

Antecedentes en modelos de proceso

El estudio de RUP permitió comprender los fundamentos que sirvieron como base para la creación de los modelos de proceso Java y MoDSGX, los cuales se busca unificar en el modelo de proceso Factor. A su vez, el considerar una versión más reciente de RUP (2003), permitió incorporar y reconsiderar algunos aspectos incluidos en los modelos de proceso existentes; la más trascendente es la descrita a continuación. Desde la versión de RUP del año 2000, la disciplina de Análisis y la disciplina de Diseño fueron fusionadas en una sola disciplina, llamada Análisis y Diseño. RUP propone la construcción de un Modelo de Análisis como una solución inicial al diseño del sistema. Este modelo es luego refinado en el Modelo de Diseño, el cual incorpora detalles correspondientes a la plataforma tecnológica que se decidió usar. La construcción del Modelo de Análisis es opcional en

RUP, y por tal motivo fue opcional también en el modelo de proceso Java. La experiencia en los últimos años de los grupos del curso Proyecto de Ingeniería de Software indica que tal modelo no es construido. Como indica C. Larman en Applying UML and Patterns [20], mantener el Modelo de Análisis y el Modelo de Diseño en forma independiente y consistente requiere de gran esfuerzo comparado con los beneficios que otorga. P. Kruchten, quien actualmente guía el avance de RUP, dice en The Rational Unified Process - An Introduction [21], que el Modelo de Análisis puede verse como una primera pasada del Modelo de Diseño. Sin desconocer el valor y la importancia de la creación del Modelo de Análisis y su correcta utilización, resulta adecuado para el contexto del modelo de proceso que estamos definiendo, proponer solamente la creación del Modelo de Diseño.

Por otro lado, el estudio de Extreme Programming brindó una visión de un proceso de desarrollo ágil que incluye prácticas completamente orientadas al cliente y a la resolución de los requerimientos. Por consiguiente, proporcionó una perspectiva diferente a la hora de definir el nuevo proceso. Sin embargo, por las características del curso en que se aplica el modelo de proceso, no fue posible obtener un aporte directo de esta metodología. Estas características refieren principalmente a que los desarrolladores rara vez trabajan en el mismo sitio, el cliente no puede estar en el lugar, y la escasez de documentación dificulta la tarea del director del proyecto (rol cumplido por un docente). Más adelante, al discutir el impacto del estudio de mecanismos de agilización del proceso, se retoma este tema.

Por su parte, DSDM brindó otra perspectiva de un modelo de proceso en uso en la industria. El aporte fundamental que rescatamos de esta metodología es lo referente a la definición y caracterización del rol de usuario o cliente. El modelo de proceso MoDSGX ya contempla una caracterización similar de este rol, la cual fue incluida en el modelo de proceso Factor. Es importante notar que la caracterización utilizada se encuentra alineada con la descrita en DSDM.

Agilizar el Proceso

Del estudio de los trabajos sobre la agilización del proceso se pudieron extraer algunos aportes importantes. Los dos modelos que se busca unificar no tienen como base una metodología ágil, sino que, por el contrario, tienen como base a RUP que es conocido por ser un proceso “pesado”. Lo que resulta de este estudio es que RUP no es, en realidad, tan “pesado” como parece, sino que aparenta serlo por ser un proceso completo que pretende contemplar la construcción de cualquier sistema de software. RUP brinda una cantidad de actividades y artefactos que no necesariamente deben ser realizados en todos los proyectos en los que se aplique, sino que cada proyecto debe realizar una instanciación o realización en la que alguien con conocimiento y experiencia decida qué actividades y qué artefactos se deben realizar y con qué nivel de detalle.

Los modelos de proceso existentes en el Gris son también una guía para los equipos que participan del curso Proyecto de Ingeniería de Software, y por consiguiente son pasibles de realizar una instanciación que no resulte en extremo “pesada”. Un detalle que se debe tener en cuenta es que los integrantes de los equipos que cursan la asignatura son estudiantes que, en general, no cuentan con el nivel de conocimiento y experiencia como para realizar una adecuada instanciación, lo que resulta en un esfuerzo adicional. Por otro lado, el hecho de que el modelo de proceso se enmarque en un curso implica que exista documentación orientada al control del avance de los equipos por parte de los docentes, lo que redundará una vez más en un mayor esfuerzo de desarrollo.

Entre los objetivos del modelo de proceso está ser didáctico y brindar una guía fácil de seguir por los grupos, así como proporcionar artefactos que sirvan para su seguimiento por parte de los docentes. Lo anterior ya se logra satisfactoriamente con los modelos de proceso existentes, por lo que alinearse con su enfoque garantiza que el nuevo modelo satisfaga tales objetivos. Si bien GXP fue aplicado en el curso obteniendo resultados positivos, algunas de sus prácticas hacen muy difícil su aplicación. Como ya se dijo, no siempre se cuenta con clientes que puedan estar a disposición en cualquier momento, y el seguimiento de los docentes debe ser realizado más de cerca por no contar con la documentación idónea. Esto conlleva, a su vez, a un mayor esfuerzo por parte tanto de los clientes como de los docentes.

Como ya se mencionó anteriormente, algunas de las mejores prácticas de XP ya se encuentran incorporadas de alguna forma en RUP y en los modelos de proceso existentes. La programación en pares es una de las prácticas requeridas por XP mientras que, si bien no es requerida en RUP, sí es mencionada en trabajos incluidos en el producto. Otras prácticas de XP son difíciles de incorporar en el contexto para el que se construye el modelo de proceso Factor.

Mejorar el Proceso

El trabajo estudiado sobre especificación y descripción de la liberación resultó de interés ya que propone una forma unificada y ordenada de realizar una liberación. Sin embargo, como mencionan los autores, el contenido propuesto por dicho trabajo para cada artefacto ya se encuentra distribuido en diferentes artefactos de los modelos de proceso existentes. Agregar nuevos artefactos entregables, como se sugiere, redundaría en un mayor esfuerzo para los grupos. Por tal motivo, se decidió no incorporarlos ya que, como indica la práctica, la dedicación de los grupos ya se encuentra por encima de la esperada.

Del trabajo sobre la actividad de Integración se extrajeron aportes importantes. Se definió un nuevo rol, el Responsable de Integración, para coordinar la actividad y guiar su avance y éxito. Asimismo, las buenas prácticas recomendadas por los autores se incorporaron como sugerencias a la actividad en el modelo de proceso Factor.

Por último, del estudio del material presentado en el encuentro GeneXus sobre GXPoints se obtuvo una propuesta de unidad de medida para los proyectos con GeneXus. El principal problema afrontado con esta unidad es que es nueva para la comunidad y su uso no se ha estandarizado. Además, y a consecuencia de lo anterior, no se cuenta con antecedentes de mediciones en el curso utilizando esta unidad. Por tal motivo, su introducción constituye una novedad tanto para estudiantes como para docentes, quienes no cuentan con la experiencia necesaria en el uso de la misma. Se decidió entonces no incorporar aún esta unidad de medida para estimaciones y mediciones de los proyectos GeneXus ya que no se cuenta aún con la experiencia y dominio necesario a nivel de la industria y del cuerpo docente, ni con herramientas disponibles para realizar las mediciones.

2.3 Modelos de proceso existentes

Se entiende por modelo de proceso al conjunto de actividades a ser realizadas en forma ordenada para alcanzar cierto objetivo. Un proceso define *quién* está haciendo *qué*, *cuándo* y *cómo*, para alcanzar tal propósito. En la Ingeniería de Software el objetivo es construir un producto de software nuevo o mejorar uno existente. El Capítulo 3 presenta una descripción de la estructura de un modelo de proceso y las relaciones entre sus componentes.

Como se ha mencionado antes, el Grupo de Ingeniería de Software cuenta con cuatro modelos de proceso, a saber, Java, MoDSGX, Integrado y GXP. Cada uno de ellos está pensado para utilizarse en distintos contextos de desarrollo. Los tres primeros, por estar basados en RUP, siguen la estructura mencionada. En esta sección se brinda una breve descripción de cada uno de estos modelos.

2.3.1 Modelo de proceso Java

El modelo de proceso Java fue desarrollado en el año 2000 por A. Delgado y B. Pérez, como resultado de un proyecto de grado del Gris. Cada año desde su creación ha sido puesto en práctica en el curso de Proyecto de Ingeniería de Software.

Este modelo tiene una fuerte base en RUP por lo que podemos encontrar muchas similitudes a nivel de estructura, contenido y características. El enfoque del modelo es iterativo e incremental, dirigido por casos de uso, y centrado en la arquitectura. Incluye además actividades de soporte.

Tal como propone RUP, sus elementos principales son los Roles que describen el *quién*, los Artefactos entregables que describen el *qué*, las Líneas de trabajo (Disciplinas) que describen el *cuándo* y las Actividades que describen el *cómo*. Las Líneas de trabajo se clasifican en Básicas y de Gestión. Las primeras comprenden los Requerimientos, el Análisis, el Diseño, la Implementación, la Verificación y la Implantación, mientras que las segundas comprenden la Gestión de Configuración, la Gestión de Calidad y la Gestión de Proyecto.

2.3.2 Modelo de proceso MoDSGX

El modelo de proceso MoDSGX fue desarrollado en el año 2002 por D. Correa y X. Romano con el objetivo de contar con un modelo de proceso para el desarrollo en un ambiente de trabajo con GeneXus. Hasta ese momento el grupo solo contaba con un modelo de proceso para desarrollo de software con Java. Dado que la tecnología GeneXus ha tenido aceptación a nivel nacional e internacional, resultó de interés contar con un modelo de proceso que guiara desarrollos utilizándola. Este modelo de proceso fue aplicado en el curso Proyecto de Ingeniería de Software desde su creación, y tuvo repercusión en la comunidad de desarrolladores GeneXus.

Como ya se mencionó, los desarrollos con GeneXus constan de una Base de Conocimiento conteniendo diferentes tipos de elementos (transactions, work panels, etc.). Estas construcciones no están directamente vinculadas con el paradigma de Orientación a Objetos. Aunque RUP es un modelo de proceso que sigue este paradigma, igualmente sirvió como base y guía para la elaboración de este nuevo modelo de proceso. Por esta razón se pueden encontrar grandes similitudes entre el modelo de proceso MoDSGX y el modelo Java antes descrito.

El enfoque del modelo MoDSGX también es iterativo e incremental, dirigido por casos de uso, y centrado en la arquitectura. Mientras que el modelo de proceso Java describe su contenido orientado a dicha tecnología, el modelo de proceso MoDSGX lo describe orientándolo al uso de la tecnología GeneXus.

2.3.3 Modelo de proceso Integrado

El modelo de proceso Integrado fue creado en el año 2003 por D. Vallespir como una extensión al modelo de proceso Java. El objetivo es permitir desarrollos donde participen múltiples grupos de trabajo integrados, tanto para construir un único producto como para construir diferentes productos para un mismo cliente que deben interactuar mediante componentes comunes. Fue puesto en práctica ese mismo año con resultados satisfactorios y se propusieron ajustes para el mismo.

La estructura de este modelo es la misma que la del modelo Java. Por otro lado, el contenido comprende al del modelo Java y agrega aquellos aspectos que permiten el trabajo en conjunto de varios equipos. Entre estos agregados se encuentran nuevos roles para fomentar la comunicación entre los grupos y nuevas actividades para permitir su interacción. Asimismo se realizaron modificaciones a algunas de las actividades existentes.

En los objetivos de este proyecto no se incluye la creación de una extensión para este modelo de proceso. A pesar de ello, luego de finalizada la puesta en práctica del modelo de proceso Factor, se creó una extensión general para procesos integrados y a su vez una extensión de este para desarrollos con el paradigma de Orientación a Objetos.

2.3.4 Modelo de proceso GXP

El modelo de proceso GXP fue creado en el año 2003 por B. Pérez buscando obtener una adaptación de Extreme Programming para el desarrollo de aplicaciones con GeneXus. Este modelo fue puesto en práctica ese mismo año obteniendo resultados positivos en el producto desarrollado.

La estructura de GXP está dada por la descripción de los roles participantes, las doce prácticas de XP adaptadas a GeneXus, un acercamiento a las “Historias de Usuario”, una agenda que indica qué tareas realizar y qué artefactos entregables generar, el ciclo de vida del proceso y las estrategias que brindan lineamientos generales en distintos aspectos del desarrollo.

El modelo de proceso GXP es distinto a los anteriores en su presentación y contenido ya que está basado en los fundamentos, las prácticas y las estrategias de XP. Por tal motivo considerar este modelo en la factorización y realizar una extensión para el mismo ha quedado fuera del alcance de este proyecto.

Capítulo 3

Modelo de Proceso para Desarrollo de Software

En este capítulo se detallan las características que describen un modelo de proceso tal como se utiliza en el Gris, y que servirán de base para la construcción del modelo de proceso Factor.

Un modelo de proceso es un conjunto de actividades necesarias para transformar los requerimientos de usuarios y clientes en un sistema de software. Como se mencionó anteriormente, el objetivo de un modelo de proceso es asegurar la producción de software de calidad superior que satisfaga las necesidades dentro de un cronograma y presupuesto predecible. La estructura de un modelo de proceso consta de dos dimensiones, del Modelo y del Tiempo, estando cada una de ellas estructuradas en términos de otros elementos más básicos. En este capítulo se describen tales dimensiones y sus elementos.

Primero, la Sección 3.1 describe la dimensión del Modelo. Luego, la Sección 3.2 detalla la dimensión Tiempo. Por último, la Sección 3.3 presenta la estructura general de un modelo de proceso utilizando un diagrama de clases de UML.

3.1 Dimensión del Modelo

La dimensión del Modelo brinda una visión estática del modelo de proceso. Esta dimensión se estructura en términos de Roles, Actividades y Artefactos entregables. Se describen a continuación estos conceptos y sus relaciones.

3.1.1 Equipo de trabajo y Roles

Un modelo de proceso es utilizado por equipos de trabajo que se identifican por un número. En la práctica, este número corresponde al número de grupo del equipo en el contexto del curso Proyecto de Ingeniería de Software. Cada equipo de trabajo esta integrado por un total de entre diez y trece personas, todos ellos estudiantes del curso, conociéndose para cada uno su nombre y cédula de identidad.

Por otro lado, un rol define el comportamiento y responsabilidades de un individuo, o un conjunto de individuos que trabajan en grupo, dentro del contexto de una organización de desarrollo de software. Cada rol se especifica mediante una descripción y un perfil. Además, los roles se asocian a un conjunto de actividades cohesivas que deben realizar. Estas actividades están estrechamente relacionadas y funcionalmente acopladas, y se realizan mejor por el mismo individuo o grupo de individuos.

Los roles se clasifican en externos e internos. Un rol es externo cuando debe ser cumplido por personas ajenas al equipo de trabajo, como por ejemplo docentes del curso o clientes. Por otro lado, los roles internos son aquellos de deben ser cumplidos por integrantes del equipo de trabajo.

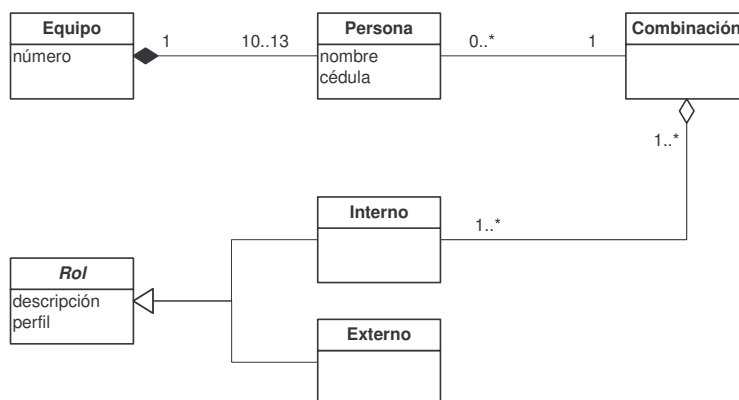


Figura 1. Estructura del equipo de trabajo y los roles.

Por definición, un rol puede ser asumido por más de un integrante de un equipo de trabajo. Asimismo, un integrante puede cumplir varios roles. Como mecanismo para la definición de qué roles pueden ser cumplidos por el mismo integrante, cada modelo de proceso estipula las combinaciones de roles posibles. Una combinación de roles se compone solo de roles internos. Cada una de ellas puede ser asumida por uno o más integrantes del equipo, pero éstos solo pueden cumplir con una combinación. Como ejemplo, puede verse en la Tabla 1 las combinaciones de roles propuestas para la extensión OO, contemplando los distintos tamaños de equipos de trabajo posibles.

En la Figura 1 se presentan estos conceptos y sus relaciones utilizando un diagrama de clases de UML.

3.1.2 Actividades y Artefactos

Una disciplina muestra las actividades que se deben realizar para producir un juego de artefactos relacionados con un interés particular sobre el proyecto, los cuales en su mayoría son entregables. Las disciplinas se identifican por un nombre y se describen mediante un conjunto de actividades, roles y artefactos involucrados.

Una actividad es algo que un rol hace y que proporciona un resultado significativo en el contexto del proyecto. Se describe mediante un nombre, un objetivo y una descripción. En la realización de la actividad participan varios roles, de los cuales uno es el responsable. Cada actividad se realiza en una o más semanas donde en cada una de ellas puede tener una prioridad distinta. Las prioridades que pueden tener las actividades son tres: Alta, Media y Baja.

La prioridad Alta indica que la actividad y el artefacto de salida (si existe) deben realizarse en la semana que se indica. Es importante tener en cuenta que el hecho de que una actividad tenga un artefacto entregable como salida no implica que dicho artefacto tenga que estar completo en su totalidad. Sin embargo, cada artefacto de salida debe contener la información necesaria para cumplir con los objetivos de la fase. Si el artefacto es entregable debe formar parte de la entrega semanal.

La prioridad Media indica que la actividad y el artefacto de salida (si existe) deben realizarse en la semana que se indica. Sin embargo, no es necesario lograr un alto grado de avance en su elaboración para cumplir con los objetivos de la fase. Si el artefacto es entregable debe formar parte de la entrega semanal.

Combinación de Roles	Equipo 10 personas	Equipo 11 personas	Equipo 12 personas	Equipo 13 personas
Administrador Asistente de Verificación Responsable de la Comunicación	1	1	1	1
Analista Documentador de Usuario Asistente de Verificación Instructor	1	1	1	1
Analista Implementador	2	2	2	3
Analista Diseñador de Interfaz de Usuario Implementador	1	1	1	1
Responsable de Verificación Responsable de SQA	1	1	1	1
Arquitecto Coordinador de Desarrollo Asistente de Verificación	1	1	1	1
Especialista Técnico Implementador	1	2	3	3
Especialista Técnico Implementador Responsable de Reuso	1	1	1	1
Responsable de SCM Especialista Técnico Implementador Responsable de Integración	1	1	1	1

Tabla 1. Ejemplo de combinación de roles tomado de la extensión OO del modelo de proceso Factor.

Por último, la prioridad Baja indica que es conveniente comenzar a realizar tanto la actividad como el artefacto de salida (si existe), pero si no se realizara no comprometería los objetivos de la fase. Si se comienza a realizar el artefacto que sea entregable, y el equipo considera además que aporta valor a la entrega semanal, es bueno que forme parte de ella.

Las actividades están estrechamente relacionadas a los artefactos. Los artefactos son la entrada y salida de las actividades, y comprenden el principal mecanismo mediante el cual se comunica la información entre las actividades. Los artefactos tienen un nombre y un objetivo, y en general, son entregables del modelo, es decir que deben ser entregados a los docentes del curso y en algunos casos también a los clientes.

Los artefactos que son entregables tienen asociadas plantillas que sirven de base y de guía a los integrantes del equipo de trabajo para la realización de los mismos. Cada artefacto tiene un rol responsable que no necesariamente es el responsable de la actividad que lo produjo. Aunque una persona puede ser el responsable del artefacto, otras personas pueden usarlo, incluso actualizarlo si se les da el permiso correspondiente.

En la Figura 2 se muestra la estructura de las actividades y artefactos, y sus relaciones, utilizando un diagrama de clases de UML.

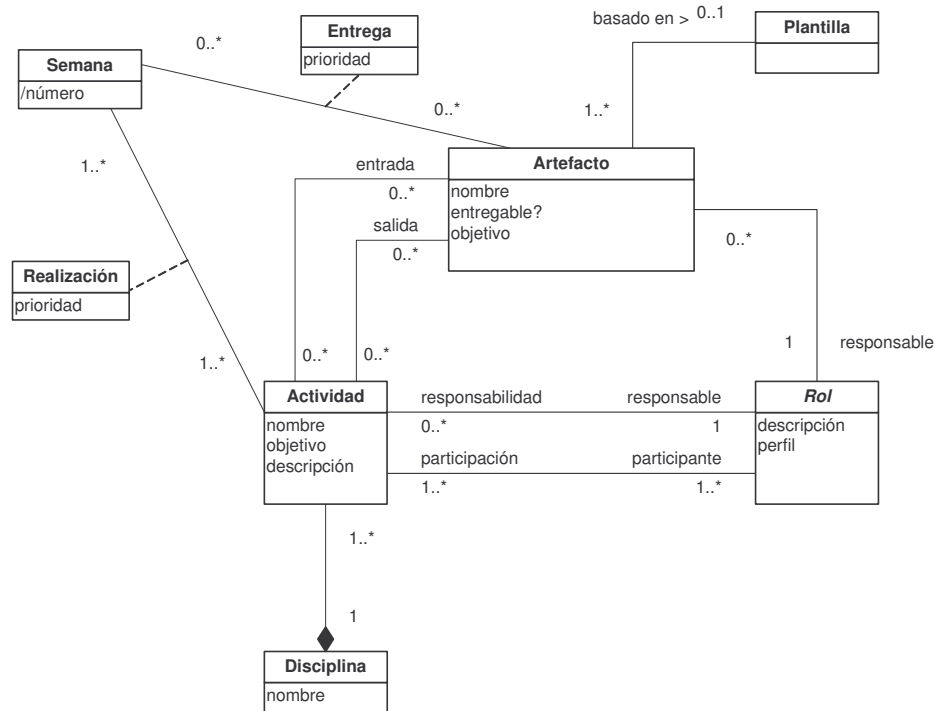


Figura 2. Estructura de las actividades y los artefactos.

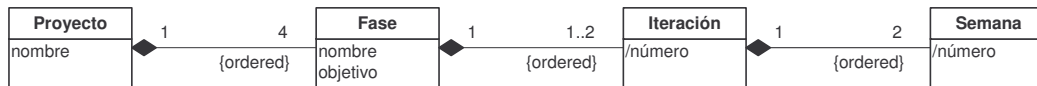


Figura 3. Estructura de la dimensión Tiempo.

3.2 Dimensión Tiempo

En esta sección se describe la dimensión Tiempo. Esta dimensión muestra los aspectos del ciclo de vida del proceso, representa la visión dinámica del mismo, y se define en base a fases, iteraciones y semanas. La Figura 3 presenta la estructura de la dimensión Tiempo mediante un diagrama de clases de UML.

3.2.1 Fases

Desde el punto de vista de la administración y la gestión, el ciclo de vida del software es descompuesto en el tiempo en cuatro fases secuenciales. Cada una se describe mediante un nombre y sus objetivos, y culmina en un hito. Un hito es un punto en el proyecto en el que los artefactos alcanzan un estado de completitud predefinido. Solo así podrá darse por culminada la fase, esto es, cuando los artefactos satisfagan los objetivos de la misma. En cada fin de fase se realiza una evaluación o valoración para determinar si se han cumplido los objetivos planteados. Una evaluación satisfactoria permite avanzar a la próxima fase del proyecto. Las fases no son idénticas en términos de cronograma y esfuerzo, y esto varía considerablemente dependiendo del proyecto.

En la Figura 4 se presentan como ejemplo los hitos planteados para el modelo de proceso Factor.

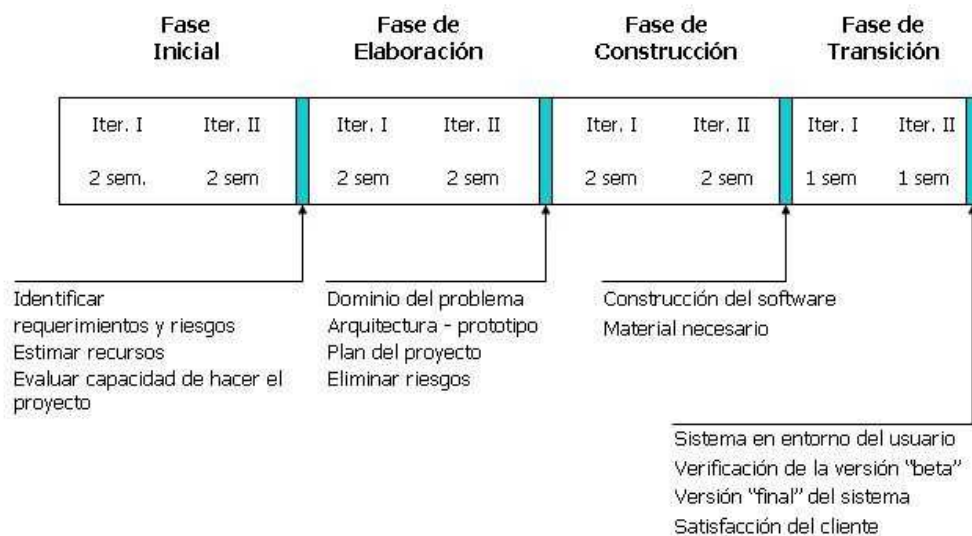


Figura 4. Hitos del modelo de proceso Factor.

3.2.2 Iteraciones y Semanas

De la misma forma en que un proyecto se divide en fases, las fases se dividen en iteraciones siempre con la idea de cumplir los objetivos de la fase. Una iteración es una recorrida completa a través de las disciplinas, resultando en una liberación (interna o externa) que forma parte del conjunto del sistema que crece incrementalmente de iteración en iteración hasta llegar al sistema final.

Cada iteración tiene un número dentro de la fase a la que corresponde y se compone de semanas. Una semana es la unidad básica de tiempo dentro de un proyecto, ya que se realizan en forma semanal las entregas de artefactos y las reuniones de seguimiento del equipo con el director del proyecto.

3.3 Estructura del modelo de proceso

En resumen, la estructura general del modelo de proceso descrito en este capítulo se presenta en la Figura 5. Se puede apreciar en esta figura cómo se relacionan las distintas componentes brindando una visión general de lo que entendemos por modelo de proceso.

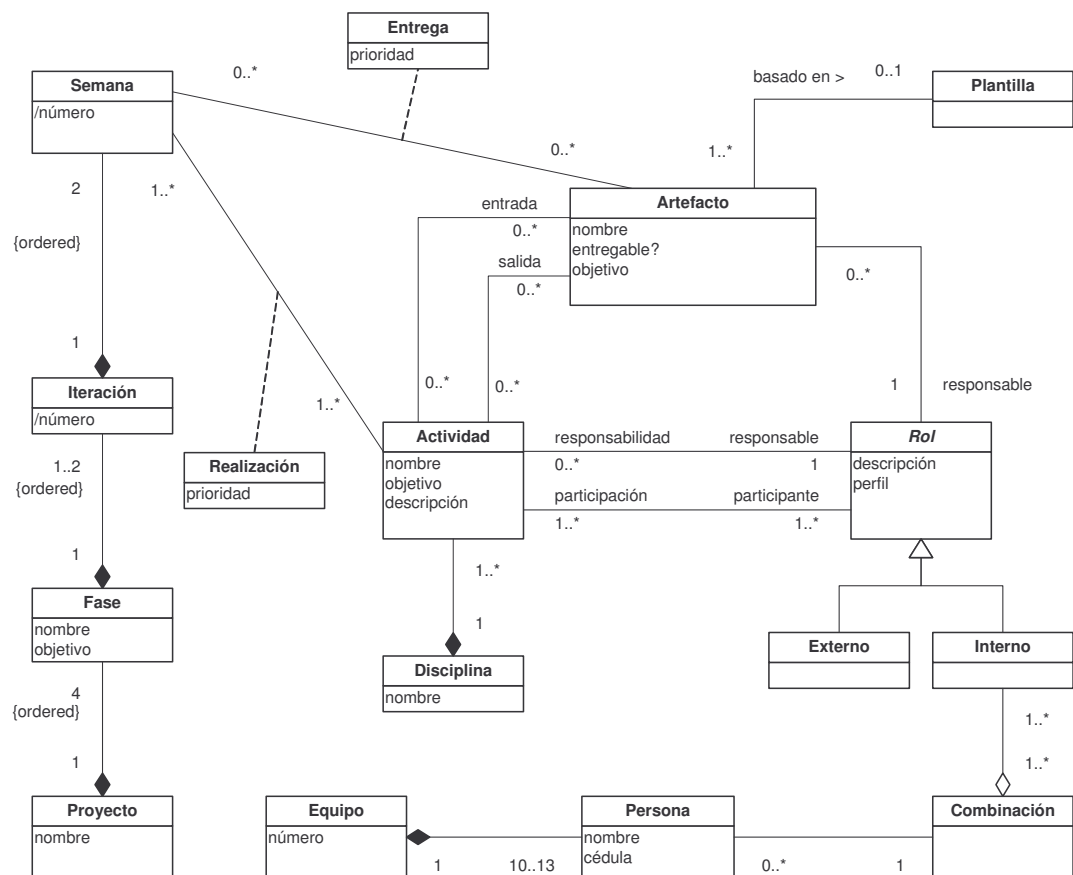


Figura 5. Estructura general del modelo de proceso.

Capítulo 4

Modelo de Proceso Factor y Extensiones

En este capítulo se presenta el modelo de proceso Factor y las extensiones creadas para el desarrollo Orientado a Objetos, el desarrollo con GeneXus, y el desarrollo en un contexto de integración de varios grupos para lograr un único producto o productos relacionados. Este modelo fue creado basándose en las componentes, los principios y las definiciones que se explicaron en el Capítulo 3 y que entendemos por modelo de proceso.

Este Capítulo se estructura de la siguiente manera. La Sección 4.1 recapitula los principios que rigen al modelo de proceso. La Sección 4.2 describe la creación del modelo de proceso Factor y las extensiones. Por último, la Sección 4.3 presenta cómo está definido el modelo de proceso. Para ello, describe el diseño abstracto del sitio web que lo representa.

4.1 Enfoque del Modelo

Como consecuencia de los modelos tomados como base, el modelo de proceso Factor sigue un enfoque iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura. El modelo es iterativo e incremental pues, como vimos, se organiza en iteraciones donde cada una puede ser vista como un mini-proyecto que cumple un ciclo completo de desarrollo, y resulta en un incremento en el producto. El estar dirigido por casos de uso implica que cada iteración avanza atacando un subconjunto de casos de uso. Éstos no solo especifican los requerimientos sino que proveen trazabilidad entre los modelos y definen el alcance de la iteración. Por último, el modelo de proceso es centrado en la arquitectura. Ésta permite mitigar los riesgos en forma temprana y es utilizada para conceptualizar, construir, administrar y evolucionar el sistema en desarrollo.

4.2 Creación del modelo

En la presente Sección se describe la creación del modelo de proceso Factor y sus extensiones. Inicialmente se plantean consideraciones generales, presentando luego aquellas pertinentes al modelo en sí y a cada una de sus extensiones.

4.2.1 Consideraciones generales

Los modelos de proceso existentes fueron estudiados como posibles fuentes de mejoras e incluso de alternativas y sugerencias para la presentación del nuevo modelo. La forma en que fue definido el modelo de proceso Integrado como una extensión del modelo Java resultó una idea práctica a la hora de comenzar a crear el nuevo modelo. Varios aspectos debieron ser tenidos en cuenta y analizados en el proceso de creación; se describen a continuación los más importantes.

Presentación del modelo

Un aspecto fundamental refiere a cómo presentar el modelo a sus usuarios. La presentación debía resultar fácil de acceder, comprender y recorrer por parte de los estudiantes. Además, debía resultar fácil de mantener por parte de los docentes. En los modelos existentes la presentación consiste en un sitio web y en un documento de texto, ambos conteniendo la misma definición. Dado que el nuevo modelo de proceso debía contar no solo con la parte factorizada sino también con instancias para desarrollos en contextos específicos, que se mostrarían como extensiones del primero, no pareció adecuada la idea de un documento de texto ya que resultaría difícil de seguir. Por otro lado, el hecho de tener un documento de texto en forma independiente del sitio web duplicaría el trabajo de mantenimiento del modelo yendo en contra del objetivo de disminuir el mismo.

Se investigó la herramienta DocBook [\[22\]](#) como forma posible de disminuir este trabajo de mantenimiento. Esta herramienta es una aplicación del estándar SGML¹/XML² que incluye una DTD³ propia y que se utiliza en la escritura de documentación técnica, especialmente para documentar todo tipo de material y programas informáticos. DocBook está ganando cada vez más adeptos dentro de la comunidad del software libre y de código abierto. La ventaja de esta herramienta es que una vez construida la estructura se puede transformar a diversos formatos de salida, por lo que utilizarla nos hubiera permitido obtener de una misma estructura el documento y el sitio web. En el momento en que se planteó el uso de esta herramienta el tiempo con que se contaba para la creación del modelo era escaso. Se necesitaba tener disponible todo el modelo de proceso Factor y las extensiones OO y GX en corto tiempo ya que se acercaba el comienzo de los cursos. Previo a ello debía realizarse un depurado y la presentación a los docentes y directores, por lo que no se contó con el tiempo suficiente para realizar un estudio detallado de la herramienta que permitiera hacer un uso adecuado de la misma. En el contexto de este proyecto, utilizarla implicaba un riesgo muy alto de no contar a tiempo con el modelo y las extensiones para el comienzo de los cursos. Por todo lo anterior se decidió descartar esta herramienta y trabajar directamente en la elaboración del sitio web. Luego de construido el sitio web sería relativamente sencillo, aunque trabajoso, obtener el documento con el modelo de proceso. Pero, como se menciono antes, este documento resultaría difícil de seguir para el caso de las extensiones y duplicaría el trabajo de mantenimiento, por lo que fue descartada su creación.

Cuando se estudió la creación de la extensión para desarrollo integrado surgió la idea de utilizar un repositorio con la definición del modelo y sus extensiones, por ejemplo en una base de datos, y desarrollar una herramienta que generara a partir del mismo los correspondientes sitios web o incluso vistas parciales del modelo. Por no contar con los recursos suficientes y por lo avanzado del proyecto, esta idea no pudo llevarse a cabo. Igualmente fue tomada en cuenta y planteada como posible trabajo futuro. El análisis detallado de la estructura del modelo de proceso, presentado en el Capítulo 3, puede servir como base al desarrollo de esta futura herramienta.

¹ Standard Generalized Markup Language, literalmente "Lenguaje de Formato de Documentos Estándar y Generalizado"). Sistema para definir lenguajes de estructuración de documentos, para representar información estructural, presentacional o semántica junto con el contenido.

² eXtensible Markup Language. Una versión redefinida de SGML. Se le ve como el sucesor de HTML. Permite personalizar las etiquetas que describen la presentación y el tipo de los elementos de datos.

³ Data Type Document. Documento que describe la estructura de una página Web escrita en XML

El sitio web resultó ser entonces la forma más didáctica y práctica tanto para el seguimiento y aprendizaje del modelo de proceso como para su mantenimiento.

Creación de la factorización

Una vez estudiados los antecedentes y decidida la forma de presentación a utilizarse se dio paso a la creación del modelo de proceso propiamente dicho. En este proceso se realizó un contraste de las actividades existentes tanto en el modelo de proceso Java como en el MoDSGX de forma de detectar tanto aquellas que resultaran comunes como las específicas de cada contexto. Surgieron aquí, además, algunas actividades que si bien no resultaban comunes de la definición de los modelos aparecieron como buenas prácticas y fueron incorporadas a la factorización. Un proceso análogo se siguió tanto para los entregables como para los roles definidos. En el caso de las plantillas también hubo que realizar un trabajo análogo y adicionalmente lograr un formato único para todas.

Combinaciones de roles

Otro de los aspectos analizados especialmente en el proceso de creación del modelo fue la combinación de roles sugerida. De experiencias anteriores surgía la necesidad de combinar el rol de Responsable de SQA con algún otro rol, de forma de balancear mejor el esfuerzo con los demás integrantes del equipo y hacer más eficiente su tarea. El responsable de SQA debía combinar sus tareas con un rol secundario de Analista cuyo objetivo era que se involucrara en el área de Requerimientos brindando la visión de calidad en los mismos. Como ya se mencionó, la combinación de roles existente no resultaba efectiva, ni equitativa en el esfuerzo. En el modelo Java original el Responsable de SQA combinaba su rol con el de Responsable de SCM pero esto fue ajustado luego de la primera experiencia ya que no se obtuvieron buenos resultados. Surge así una nueva propuesta, para el nuevo modelo, combinando el rol de Responsable de SQA con el de Responsable de Verificación y Analista para cumplir el objetivo antedicho.

Extensiones

Así surgieron en el primer semestre, casi en forma paralela, el modelo de proceso Factor (F) y las extensiones para desarrollo Orientado a Objetos (OO) y con GeneXus (Gx). Es importante destacar que una vez obtenido el modelo Factor, y detectadas las características particulares de cada modelo existente, las extensiones fueron sencillas de obtener agregando al sitio web las páginas necesarias con la información que cada extensión agregaba o modificaba.

Luego de la puesta en práctica del modelo de proceso Factor y las extensiones, y finalizado ya el segundo semestre, se procedió al estudio y agregado de una nueva extensión que contemplara el modelo de proceso Integrado. De ahí surgieron tanto la extensión para desarrollo Integrado (I) como la extensión para desarrollo Integrado Orientado a Objetos (IOO).

En la Figura 6 se muestra la estructura general del modelo creado y sus extensiones utilizando un diagrama de clases de UML. En lo que sigue, se describen las características del modelo de proceso Factor y de cada una de las extensiones, de forma de brindar una perspectiva más clara y explicar en qué forma es extendido el F.

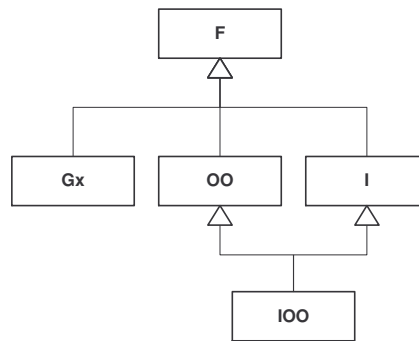


Figura 6. Estructura del modelo de proceso propuesto y sus extensiones.

4.2.2 Modelo de proceso Factor (F)

El modelo de proceso Factor constituye la columna vertebral de este proyecto ya que logra reunir, de forma independiente del contexto de desarrollo, las mejores prácticas de los dos modelos tomados como base, e incorpora a su vez algunas modificaciones con el objetivo de mejorarlo.

Como ya fue mencionado, la decisión de realizar en un primer paso la factorización de los modelos Java y MoDSGX, dio lugar a un análisis en profundidad, en el proceso de creación, de los puntos en común de ambos modelos y aquellas características propias de cada tecnología. Este proceso llevó un tiempo considerable porque implicó el estudio detallado y pormenorizado de cada modelo, contrastando, detectando similitudes y diferencias, carencias y fortalezas, que permitieran obtener lo mejor de cada uno. Aunque el modelo Java se presentaba más estable, MoDSGX presentaba mejoras que resultaron interesantes para incorporar al nuevo proceso.

El uso de este modelo está orientado al curso Proyecto de Ingeniería de software lo que implica que debe ser lo más intuitivo, claro y pedagógico posible. Los estudiantes deben aprender en este curso a seguir un modelo de proceso para desarrollo de software y enfrentarse a una realidad de desarrollo, que pocas veces se da de forma tan estricta en el mercado, pero que resulta en una experiencia formativa.

El modelo de proceso Factor sigue las líneas que se describieron en el Capítulo 3 para un modelo de proceso, por lo que para definirlo se utilizaron las dos dimensiones antes descritas: la del Modelo y la del Tiempo. Se describen cada una de las actividades así como los artefactos entregables y se brinda una plantilla para la mayoría de ellos. Se describen además los roles existentes y sus responsabilidades. Se brinda también una agenda de actividades y entregables que sirve como guía a los equipos para saber qué actividad deben estar haciendo en cada momento y que artefactos deben entregar cada semana.

Resumen de elementos del modelo de proceso Factor

Para fijar ideas y proveer mayor entendimiento acerca del modelo de proceso Factor, se presenta un pantallazo general del mismo mostrando los roles, las actividades y los artefactos más importantes dentro del modelo de proceso.

Roles

El modelo de proceso define cuatro roles externos y dieciséis roles internos. Los roles externos son el Director de Proyecto, el Consultor, el Auditor y el Cliente. Entre los roles

internos, los más importantes que pueden mencionarse son el Analista, el Arquitecto, el Administrador, el Especialista Técnico, el Responsable de SQA y el Responsable de Verificación. Por otro lado, existen dos roles internos que son opcionales, el Asistente de Arquitecto y el Diseñador de Interfase de Usuario, cuya utilización depende de las características de cada proyecto.

El Analista debe analizar el sistema, conducir y coordinar el relevamiento de requerimientos y el modelado de Casos de Uso estableciendo la funcionalidad y el alcance del sistema. Especifica los requerimientos detallando la funcionalidad del sistema y es responsable de los Casos de Uso que especifica y de los Casos de Uso contenidos. Valida los requerimientos con los usuarios o clientes.

El Arquitecto conduce y coordina las actividades y entregables técnicos a lo largo del proyecto. Su visión es general de todo el proyecto, no en profundidad como en el caso del Analista. Establece la estructura para cada punto de vista de la arquitectura: la descomposición, la agrupación de elementos, y las interfases entre los grupos. Como diseñador del sistema define las responsabilidades, funciones, atributos y relaciones de uno o varios subsistemas, y determina como se ajustan al entorno de implementación. El Arquitecto debe apoyarse en los analistas para hacer el diseño y en el Asistente de Arquitecto si se decide incorporar este rol al equipo.

El Administrador es el responsable de asignar recursos, establecer prioridades, coordinar las interacciones con clientes y usuarios y generalmente mantiene al equipo del proyecto enfocado en la meta correcta. También establece prácticas que aseguran la integridad y calidad de los entregables del proyecto. Es quien realiza el plan de proyecto y lo ajusta en cada fase. Debe conocer perfectamente el modelo de proceso a seguir y entender los requerimientos del sistema. Hace el seguimiento del proyecto, hace Estimaciones y Mediciones de tiempo, esfuerzo y tamaño del proyecto y controla el avance para cada disciplina. Es responsable de que se analicen y evalúen los riesgos en todas las áreas del proyecto. Es el responsable también de realizar la Reunión de Equipo y de llevar adelante la misma.

El Especialista Técnico es el responsable del soporte técnico en el proyecto. Esto incluye seleccionar y adquirir las herramientas, configurarlas, prepararlas y verificar que éstas funcionen correctamente. Debe tener mucho conocimiento del ambiente de desarrollo y las herramientas de programación elegidas. Debe estudiar o investigar las tecnologías que se consideren necesarias para el desarrollo del sistema. Participa en la planificación de la integración y se encarga de preparar el entorno de capacitación.

El Responsable de SQA debe asegurar la calidad de los productos generados en el proyecto y del proceso utilizado. Para asegurar la calidad debe revisar la calidad de los entregables de planificación del proyecto y los entregables de valoración del proyecto. Además revisa el nivel de apego al modelo de proceso de desarrollo de software y a los planes de Verificación, Gestión de Proyecto y Gestión de Calidad, documentando las desviaciones encontradas. Debe conocer los conceptos y técnicas de Gestión de Calidad del Software. Debe identificar las propiedades de calidad que deben cumplir los productos del proyecto así como centralizar y revisar las entregas que se realizan durante el ciclo de vida del proyecto.

Por último el Responsable de Verificación debe planificar, diseñar, implementar y evaluar las pruebas. Debe generar los planes de prueba y los casos de prueba. Debe implementar los procedimientos de prueba, evaluar la cobertura, los resultados y efectividad de las

pruebas. Debe hacer también los informes de evaluación de las pruebas. El Responsable de Verificación debe apoyarse en los Asistentes de Verificación para realizar las actividades.

Actividades

Cada rol tiene asignadas un conjunto determinado de actividades que debe llevar a cabo para el avance del proyecto. Existen setenta y tres actividades diferentes que se organizan en diez disciplinas distintas. Las disciplinas existentes son Requerimientos, Diseño, Implementación, Verificación, Implantación, Gestión de Configuración y Control de Cambios, Gestión de Proyecto, Gestión de Calidad, Comunicación y, por último, Formación y Entrenamiento.

Desde el punto de vista de las actividades, mostramos a continuación las más importantes dentro del modelo de proceso, actividades clave para el buen desarrollo de un proyecto.

- Especificar los Requerimientos

Esta actividad tiene como objetivo realizar la especificación de todos los requerimientos del sistema a construir. Estos requerimientos son funcionales, no funcionales y de interfase de usuario.

Dada la reunión con el Cliente y/o usuario los Analistas realizan la especificación de los requerimientos del sistema a construir. Se realiza un documento donde se describen las funcionalidades que debe cumplir el producto. Se debe especificar el Modelo de Casos de Uso como un todo (diagramas, descripciones, etc.) Se debe describir brevemente cada Caso de Uso primero, y luego detallar cada uno. Esta especificación será utilizada por el Arquitecto del sistema para realizar el diseño del sistema.

El artefacto de entrada es el Acta de Reunión de Requerimientos y tiene como artefactos de salida la Especificación de Requerimientos y el Modelo de Casos de Uso.

El rol responsable de esta actividad es el Analista mientras que hay otros tres roles que deben estar involucrados atendiendo a las necesidades de su rol específico, estos son el Responsable de SQA, el Responsable de Verificación y el Diseñador de Interfase de usuario si es que ha sido asignado dentro del proyecto.

- Definir el Glosario

Esta actividad tiene como objetivo definir los términos utilizados en los documentos del proyecto para que sea entendible la lectura de los mismos. Uno de los Analistas genera un documento Glosario basado en la plantilla diseñada para tal fin. A medida que se van generando documentos, todos los términos de estos documentos que se considere que deben ser explicados, para el correcto entendimiento de los documentos, se deben agregar en el Glosario con la explicación correspondiente.

No tiene artefactos de entrada y su salida es el artefacto entregable denominado Glosario.

El rol responsable de esta actividad es el Analista pero se encuentran todos los roles involucrados ya que en el Glosario debe aparecer toda la terminología del proyecto.

- Planificar el Proyecto

Esta actividad tiene como objetivo realizar la planificación del proyecto, para que éste pueda realizarse de manera controlada.

De acuerdo al documento Plan de Proyecto, el Administrador deberá realizar la planificación del proyecto. Esta planificación comienza con el conocimiento de las actividades, áreas de trabajo, modelo de proceso, etc., relacionados con la Gestión de Proyecto. A partir de esto se realiza la planificación del proyecto como está especificado en el documento de Plan de Proyecto.

Los artefactos de entrada para esta actividad son diversos ya que esta actividad tiene como objetivo realizar un plan completo del proyecto. Los artefactos son la Especificación de Requerimientos, el Alcance del Sistema, las Estimaciones y Mediciones, el Documento de Riesgos, Plan de Configuración, el Plan de Calidad, el Plan de Verificación y Validación, el propio Plan de Proyecto si existieran versiones anteriores, los Informes de Situación del Proyecto y la Planificación del Desarrollo. El artefacto de salida es el Plan de Proyecto.

El rol responsable es el Administrador pero se encuentran involucrados el Responsable de SQA, el Responsable de Verificación y el Responsable de SCM.

- Gestión de Riesgos

El objetivo principal de esta actividad es evaluar los riesgos del proyecto y realizar un seguimiento de los mismos. La actividad comprende: Identificar en cada disciplina problemas potenciales antes de que ocurran, registrar estos problemas en una lista y entregar esta lista al Administrador del Proyecto. En la reunión de seguimiento de proyecto se comunicarán los riesgos a todo el equipo y se realizará la clasificación general de todos los riesgos de acuerdo a su impacto de ocurrencia en el Proyecto. Comprende además la asignación de responsables y planificación de recuperación.

Los riesgos del proyecto se deberán detallar en un documento que especifique: el nombre de cada riesgo, la descripción del mismo, la probabilidad de ocurrencia en el proyecto, impacto en el proyecto si ocurriera, mecanismos de monitoreo, estrategia de mitigación del riesgo y plan de contingencia.

Los artefactos de entrada son el Plan de Proyecto, el modelo de proceso en sí mismo, Informes de situación de Proyecto, Documento de Riesgos si existieran versiones anteriores y las Estimaciones y Mediciones. El artefacto de salida es el Documento de Riesgos.

El rol responsable es el Administrador y se encuentran involucrados todos los roles ya que en los riesgos a considerarse deben comprender al proyecto en su totalidad.

- Describir la Arquitectura

Esta actividad tiene como objetivo realizar la descripción de la Arquitectura. Comprende realizar el diseño de los Casos de Uso que componen la arquitectura del Sistema.

Los artefactos de entrada son la Especificación de Requerimientos, el Modelo de Casos de Uso, el Glosario y el Modelo de Diseño. El artefacto de salida es la Descripción de la Arquitectura.

El rol responsable es el Arquitecto mientras que el rol de Analista y el rol de Diseñador de Interfase de Usuario en caso de haber sido designados, también deben estar involucrados en esta actividad.

- Diseñar el Sistema

Esta actividad tiene como objetivo realizar el diseño completo del sistema a construir. Comprende realizar el diseño de los Casos de Uso que componen la Arquitectura del Sistema. Dado el documento de especificación técnica de requerimientos, el Arquitecto realizará el diseño de sistema en modalidad Top-Down, lo que significa que se partirá de un diseño macro de sistema y se irá detallando de manera incremental e iterativa.

Se deben identificar todos los módulos y los objetos más importantes cuyas instancias son necesarias para ejecutar las secuencias de eventos de los casos de uso. Se deben definir y mantener dependencias de los módulos, verificar que realiza las operaciones correctas, definidas por las interfases que provee, describir interacciones entre los objetos más relevantes, identificar submódulos e interfases, describir interacciones entre submódulos y diseñar la interfase de usuario.

Los artefactos de entrada son la Especificación de Requerimientos, el Modelo de Casos de Uso, el Glosario y las Pautas para la Interfase de Usuario (si existe este documento en dependencia de las características del proyecto y de la importancia que tenga para el mismo la interfase de usuario). El artefacto de salida es el Modelo de Diseño.

El rol responsable es el Arquitecto y están involucrados en la actividad tanto el rol de Analista como el de Documentador de Usuario.

- Planificar la Verificación

Esta actividad tiene como objetivo realizar la planificación de las actividades relacionadas con la Verificación del producto a ser desarrollado.

Basándose en el documento que se proporciona como base para el Plan de Verificación, el responsable de Verificación deberá realizar la planificación de las actividades que se realizarán para verificar el sistema a ser desarrollado.

Esta actividad comprende planificar como se va a validar y verificar todo, identificar y describir los métodos a ser usados para verificar que los casos de uso han sido aprobados por los analistas de requerimientos y los usuarios e identificar y describir los métodos a ser usados para verificar que se mantienen las trazas entre los modelos.

Los artefactos de entrada son la Especificación de Requerimientos y el Plan de verificación y Validación si existieran versiones anteriores. El artefacto de salida es el Plan de verificación y Validación.

El rol responsable es el Responsable de Verificación y no tiene otros roles involucrados.

- Planificar la Implantación

Esta actividad tiene como objetivo planificar la liberación del producto de software al ambiente del usuario. Se debe determinar cómo y cuando el producto estará disponible al usuario final. La planificación de la transición al ambiente del usuario requiere un alto nivel de colaboración y preparación por parte del cliente. El Plan de Transición debe lograr un

software pronto para liberar en el ambiente del usuario, material de capacitación y material de apoyo para asegurar que el usuario final pueda usar exitosamente el producto liberado.

La conclusión exitosa del proyecto puede ser muy influenciada por factores que están fuera del alcance del desarrollo del software como son la infraestructura edilicia o la infraestructura de hardware no adecuada y personal no preparado para trabajar con el nuevo sistema.

Los artefactos de entrada son el Plan de Proyecto, el Plan de Desarrollo, el Plan de Verificación y Validación, el Plan de Calidad, el Plan de Configuración, el Alcance del Sistema (se toma como plan de aceptación) y la Especificación de Requerimientos. El artefacto de salida es el Plan de Implantación.

El rol responsable es el Especialista Técnico pero tiene como involucrados a varios roles claves del modelo de proceso. Los roles involucrados son el Administrador, el Arquitecto, el Coordinador de Desarrollo, el Responsable de Verificación, el Responsable de SQA, el Responsable de SCM, el Responsable de Integración, el Documentador de Usuario y el Cliente.

Artefactos

Cada actividad que es realizada tiene por lo general algún artefacto de entrada o salida como vimos anteriormente. Vamos a ver ahora los artefactos más importantes para el modelo de proceso Factor relacionándolos con los roles y las actividades más importantes ya presentadas. La Figura 7 muestra un esquema de los artefactos principales y sus relaciones.

Comencemos describiendo la Especificación de Requerimientos que es uno de los artefactos más importantes dentro del modelo de proceso. El objetivo de este artefacto entregable es especificar formalmente todos los requerimientos del sistema relevados en las reuniones de requerimientos con el cliente. Este artefacto es salida de la actividad Especificar los Requerimientos así como también de la actividad Definir Pautas para la Interfase de Usuario en caso de que no exista un documento específico para ello (en dependencia de la importancia que tenga la interfase de usuario para el cliente). Es entrada de las actividades más importantes del modelo ya que en este documento se basa toda la construcción del sistema. El rol responsable de este artefacto es el rol de analista quien juega un papel principal en la recolección de los requerimientos.

El Modelo de Casos de Uso tiene como objetivo especificar detalladamente los requerimientos del sistema utilizando para ello la herramienta de Casos de Uso. Este artefacto entregable es también salida de la actividad de Especificar los requerimientos y el rol responsable es el analista. Estos dos artefactos son básicos en el modelo de proceso y se realizan en la Fase Inicial, Fase de Elaboración y Fase de Construcción iterando e incrementando cada uno de ellos.

Luego tenemos el Glosario que debe definir las abreviaturas, siglas y términos usados en el proyecto (documentos y software). El Glosario se genera en la Especificación de Requerimientos pero todas las actividades pueden y deben enriquecerlo. El rol responsable es el analista iterando e incrementándolo en la Fase Inicial, Fase de Elaboración y Fase de Construcción.

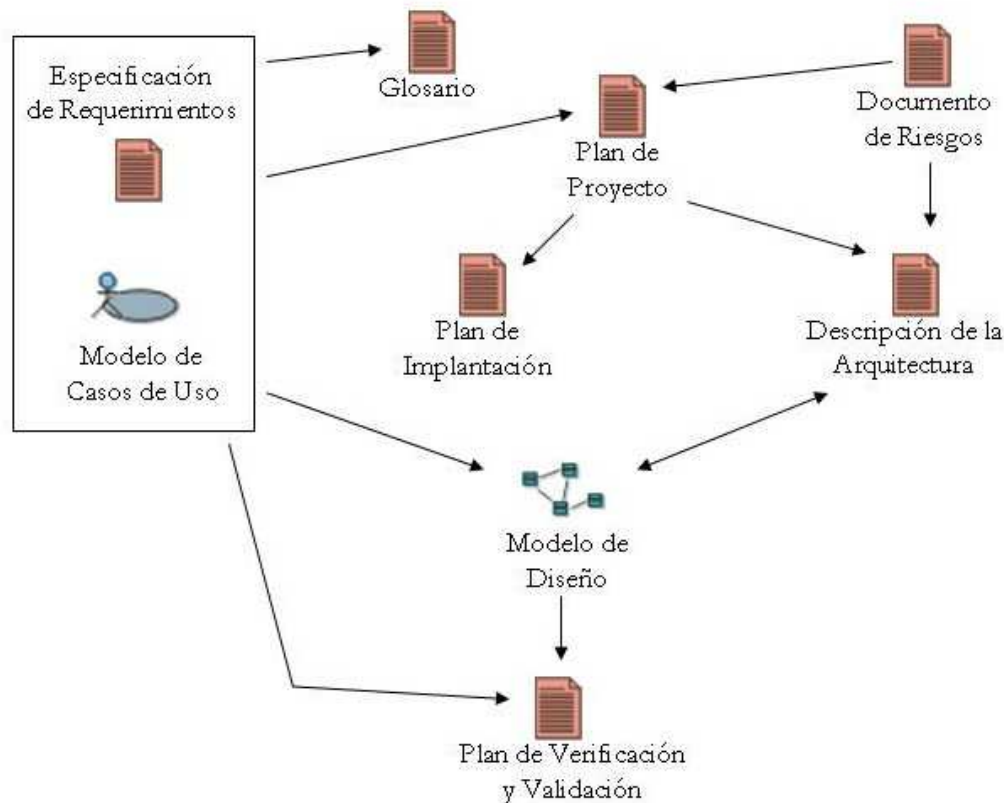


Figura 7. Artefactos principales.

El Documento de Riesgos debe detallar los riesgos del proyecto especificando: el nombre de cada riesgo, la descripción del mismo, la probabilidad de ocurrencia en el proyecto, impacto en el proyecto si ocurriera, mecanismos de monitoreo, estrategia de mitigación del riesgo y plan de contingencia. Es un documento muy importante ya que un riesgo no identificado puede llevar al fracaso del proyecto, por lo anterior debe ser seguido muy de cerca y por lo tanto se debe iterar e incrementar en las cuatro fases; Inicial, Elaboración, Construcción y Transición. Es salida de la actividad de Gestión de Riesgos y entrada de la actividad de Planificar el Proyecto, la propia Gestión de Riesgos y Realizar el Informe final de Proyecto. El rol responsable es el administrador pero todos los integrantes del equipo deben cooperar en la detección de posibles riesgos, clasificación y seguimiento.

El Plan de Proyecto describe la planificación del proyecto, para que éste pueda realizarse de manera controlada. Se describen las actividades de gestión de proyecto que deben ser llevadas a cabo durante el proceso de desarrollo del proyecto. Se debe realizar tanto en la Fase Inicial como en la Fase de Elaboración y es salida de la actividad Planificar el Proyecto, además es entrada de otras muchas actividades que se basan en este documento para hacer el seguimiento y gestión de las diferentes áreas del proyecto. Es responsabilidad del Administrador del equipo.

El Modelo de Diseño especifica formalmente el diseño del sistema, esto incluye su descomposición en subsistemas o módulos, la interacción entre las partes, y los datos que maneja el sistema. Es salida de la actividad Diseñar el Sistema, el rol responsable es el arquitecto y debe realizarse en las fases Inicial, Elaboración y Construcción.

Luego de tener el Plan de Proyecto se puede realizar el Plan de Implementación que debe describir cómo y cuando va a estar disponible el producto a la comunidad de usuarios. Es salida de la actividad Planificar la Implantación y su rol responsable es el Especialista Técnico. Se debe realizar en las fases Elaboración, Construcción y Transición. Es entrada de actividades relacionadas con la liberación de versiones y con la capacitación de los usuarios.

La Descripción de la Arquitectura es también un artefacto clave del modelo. Debe describir la Arquitectura del Sistema desde el punto de vista de Modelo de Casos de Uso, Modelo de Diseño, Modelo de Implementación, Modelo de Distribución y las trazas entre ellos. El rol responsable es el arquitecto y es salida de la actividad Describir la Arquitectura. Debe realizarse en las fases Inicial, Elaboración y Construcción. Es entrada de las actividades relacionadas con la comunicación del diseño a los implementadores y con la implementación propiamente dicha.

Por último el Plan de Verificación y Validación se apoya en el Modelo de Diseño. Debe identificar los componentes de software y documentos que deben ser verificados así como describir las estrategias de verificación que serán usadas e identificar los recursos necesarios y proporcionar una estimación de esfuerzo para realizar la verificación. Se debe realizar en las fases Inicial y Elaboración y es salida de la actividad Planificar la Verificación. El rol responsable es el Responsable de Verificación y es entrada para todas las actividades relacionadas con la verificación del sistema.

Una vez entendido el modelo de proceso Factor la extensión consiste en adaptarlo para desarrollos en contextos particulares, esto puede incluir el agregado de nuevos roles, actividades y artefactos entregables e incluso la modificación de los ya existentes. Vamos a ver ahora que adaptaciones fue necesario realizar para cada una de las extensiones creadas.

4.2.3 Extensión para desarrollo Orientado a Objetos (OO)

Una vez obtenido el modelo de proceso Factor ya se contaba con una buena base para la obtención de las extensiones deseadas para desarrollo Orientado a Objetos y GeneXus. El análisis realizado para detectar los puntos en común de los modelos ya existentes arrojó también, como resultado paralelo, un estudio de aquellas características que correspondían a cada contexto en particular que deberían ser incorporadas en las extensiones.

La extensión para el desarrollo OO corresponde al modelo de proceso existente para el desarrollo con Java. En realidad, las prácticas allí sugeridas no eran específicas para dicha tecnología, por lo que la extensión generada se propone para cualquier tecnología alineada con el paradigma de Orientación a Objetos.

En particular, esta extensión agrega un rol que resulta opcional para cada equipo de trabajo y nuevas actividades, así como modificaciones y/o agregados a otras. Agrega además nuevos artefactos entregables del modelo, y con ellos, las plantillas que sirven como guía para su realización.

4.2.4 Extensión para desarrollo con GeneXus (Gx)

Análogamente al trabajo realizado para obtener la extensión para desarrollo Orientado a Objetos se detectaron aquellas características que resultaban exclusivas para el desarrollo con GeneXus.

Esta extensión agrega un documento, que forma parte del modelo existente MoDSGX, con una metodología para desarrollo con GeneXus que describe una forma de trabajo para administrar las Bases de Conocimiento. Agrega además nuevos roles y modifica otros existentes en el modelo de proceso Factor. Lo mismo ocurrió con las actividades, entregables y plantillas.

4.2.5 Extensión para desarrollo Integrado (I)

Una vez finalizado el segundo semestre y finalizada la puesta en práctica del modelo de proceso Factor y de las extensiones para desarrollo Orientado a Objetos y GeneXus, se decidió incorporar también el modelo de proceso Integrado existente en el Gris.

Como se mencionó anteriormente, el modelo de proceso Integrado fue definido originalmente como una extensión al modelo de proceso Java por lo que se podría obtener directamente de la extensión OO una extensión para desarrollo integrado orientado a objetos. Teniendo en cuenta el trabajo ya realizado y para mantener la línea buscada en el proyecto, se decidió construir primero una extensión del modelo Factor para desarrollo integrado que fuera independiente de la tecnología.

Para obtener la extensión para desarrollo integrado (I) lo que se hizo fue detectar en el modelo de proceso Integrado aquellas actividades dependientes e independientes de la tecnología, aquellas actividades que extendían o modificaban actividades del modelo Java y aquellas actividades que se agregaban para apoyar la interacción de los equipos. Así, la extensión para desarrollo integrado contiene principalmente aquellas actividades que se agregan al modelo Factor para permitir la interacción entre los grupos así como también dos nuevos roles con el mismo objetivo, aquellas actividades que modifican o agregan algo del modelo Factor y plantillas modificadas para los entregables que son salidas de dichas actividades. Esta extensión incorpora además algunas actividades nuevas y modifica otras buscando mejorar el proceso atendiendo a los resultados obtenidos de la puesta en práctica en el año 2003 del modelo de proceso Integrado.

4.2.6 Extensión para desarrollo Integrado Orientado a Objetos (IOO)

Una vez obtenida la extensión para desarrollo integrado independiente de la tecnología, surgió como resultado paralelo también la extensión para desarrollo integrado orientado a objetos (IOO). Esta extensión contiene actividades, entregables y plantillas que se modifican tanto del I como del OO.

4.3 Definición del Modelo de proceso

El modelo de proceso definido se encuentra accesible en el CD que acompaña al informe (ver Apéndice A). Allí, se puede acceder al modelo de proceso Factor (F) independiente de la tecnología y del ambiente de desarrollo, y las cuatro extensiones para desarrollos en contextos específicos: la extensión para desarrollo Orientado a Objetos (OO), la extensión para desarrollo con GeneXus (Gx), la extensión para desarrollo Integrado independiente de

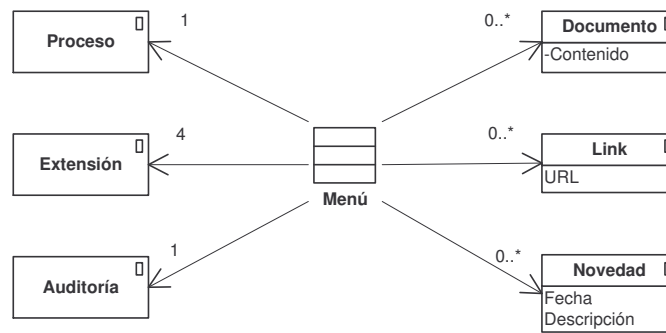


Figura 8. Modelo de la Estructura de Navegación del sitio web.

la tecnología y del ambiente de desarrollo (I), y por último una extensión para desarrollo Integrado Orientado a Objetos (IOO).

En la presente Sección se describe el diseño abstracto del sitio web. El mismo contiene todo lo referente al modelo de proceso definido y los materiales que se brindan en el curso como apoyo a los estudiantes. El diseño del sitio se realizó siguiendo una metodología basada en UML para el diseño de Hypermedia [23]. Esta metodología fue propuesta por R. Hennicker y N. Koch basada en un perfil de UML para el dominio hypermedia.

4.3.1 Sitio Web del modelo

La metodología utilizada propone comenzar con un análisis de casos de uso y un modelo conceptual de la aplicación. Luego provee lineamientos para pasar del modelo conceptual a un modelo del espacio de navegación, el cual especifica qué objetos pueden ser visitados mediante una navegación. Asimismo, describe cómo derivar de éste un modelo de la estructura de navegación, el cual muestra cómo dichos objetos son alcanzados. Finalmente propone un modelo de presentación que puede ser implementado directamente con frames HTML. Todos los modelos planteados se representan usando una extensión de UML para diseño hypermedia.

Los modelos considerados importantes para la comprensión del sitio web son el modelo conceptual, el modelo del espacio de navegación y el modelo de la estructura de navegación. El modelo de presentación se utiliza para diseñar el sitio al nivel de detalle que involucra los frames, las imágenes y los fragmentos de texto, lo cual no aporta para nuestro objetivo. Por otro lado el modelo del espacio de navegación queda, en este caso, igual al modelo de la estructura de navegación, excepto por el menú del sitio, ya que el sitio no contiene aspectos dinámicos como índices y búsquedas. Por tal motivo, no brinda ningún valor adicional tener dos modelos prácticamente idénticos, ya que solo el modelo de la estructura de navegación cumple con el objetivo planteado. Otro detalle importante es que el modelo construido no muestra toda la navegabilidad permitida en el sitio ya que complicaría la comprensión del mismo. Se muestra solo la forma más directa e intuitiva de llegar a cada lugar en el sitio.

El modelo conceptual requerido corresponde al que fue presentado en la Figura 5 en el Capítulo 3. Por otro lado, la Figura 8 muestra el modelo de estructura de navegación para el sitio web. Este modelo incluye el menú del sitio y muestra qué se puede acceder desde el mismo. Como se puede ver, desde el menú del sitio se accede al proceso, que corresponde

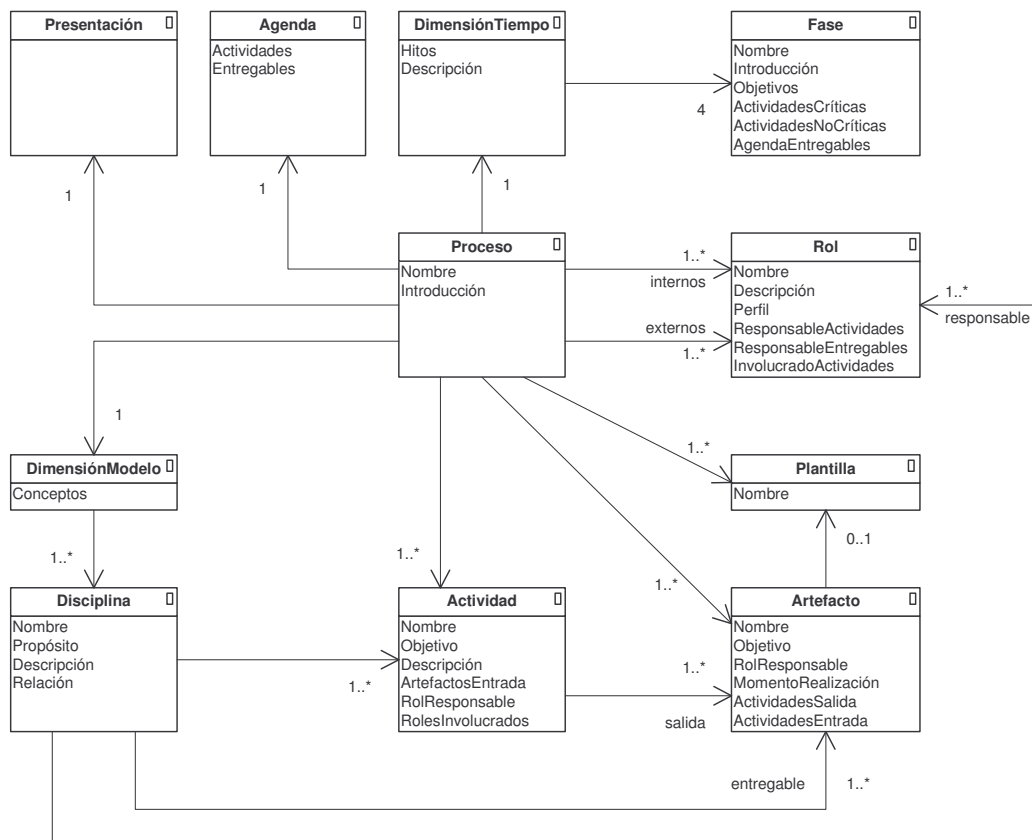


Figura 9. Modelo de la Estructura de Navegación del proceso.

a la definición del modelo Factor, a las cuatro extensiones, léase OO, Gx, I e IOO, y se accede también al material de las auditorías realizadas, a los documentos de apoyo para las distintas disciplinas, a los links con información adicional y a las novedades.

Luego, en la Figura 9 se presenta la porción del modelo de la estructura de navegación que correspondiente a la definición del proceso. El proceso tiene como atributo un nombre y una descripción y desde el mismo se puede acceder a la presentación en diapositivas realizada a los estudiantes al comienzo del curso así como a la agenda de actividades y entregables propuesta. Se puede acceder también a la dimensión del Modelo que explica los conceptos básicos del modelo y permite acceder a las disciplinas que constituyen dicha dimensión. Desde una disciplina se puede acceder a los roles responsables de la misma así como a las actividades y artefactos entregables. Por otro lado, desde el proceso se puede acceder a la dimensión Tiempo que tiene como atributos una descripción y los hitos dentro del mismo. Desde la dimensión Tiempo se puede navegar hacia las fases del proceso cuyos atributos son un nombre, una introducción, los objetivos, las actividades críticas, las no críticas y la agenda de entregables por iteración. Desde el proceso también podemos acceder a los roles tanto externos como internos, las actividades, los artefactos y las plantillas. Por último, desde una actividad podemos acceder a los artefactos que son salida de la misma y desde un artefacto podemos acceder a su plantilla.

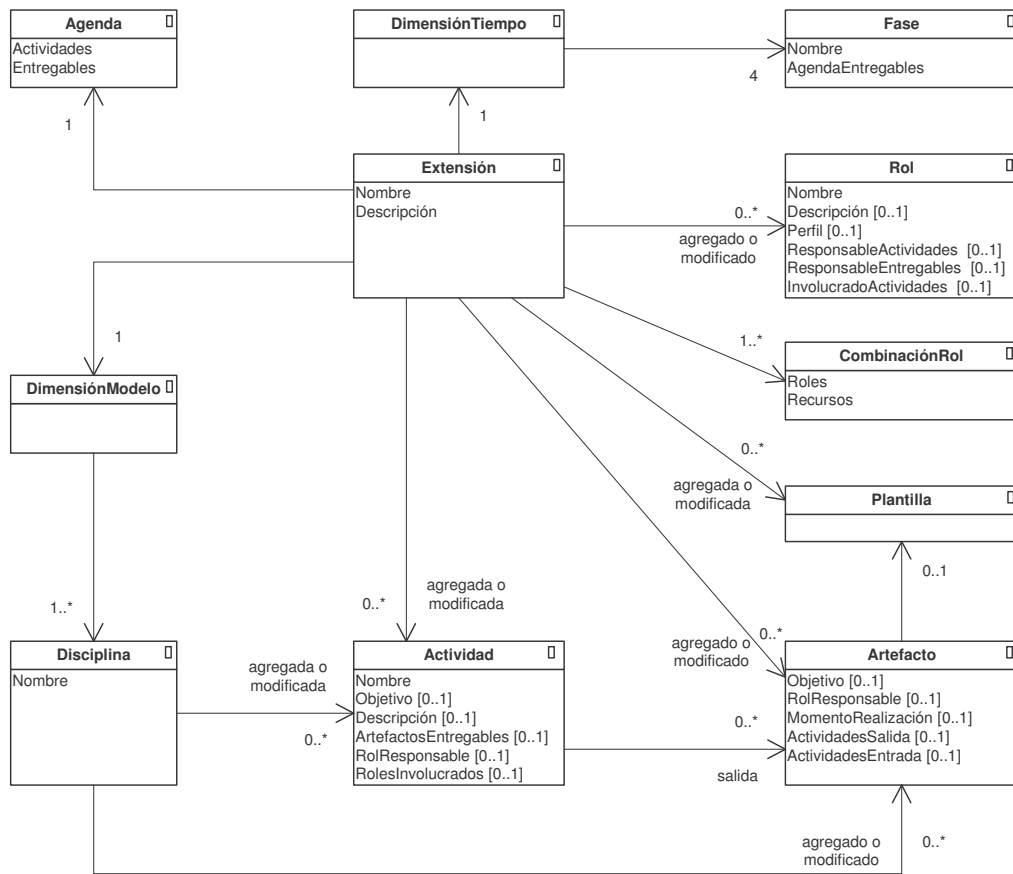


Figura 10. Modelo de la Estructura de Navegación de las extensiones.

En la Figura 10 se puede ver la porción del modelo que muestra la parte del sitio correspondiente a una extensión. Esta parte del sitio es, en su estructura, casi idéntica a la correspondiente al proceso, pero las clases navegacionales no son las mismas. Una clase navegacional modela una clase cuyas instancias son visitadas por el usuario durante la navegación. Existen algunas diferencias entre la parte del modelo para el proceso y la parte del modelo para las extensiones que se enumeran a continuación. En primer lugar, debe decirse que el concepto de extensión se muestra agregando o modificando información de las instancias de las clases navegacionales. Como ejemplo se puede citar el caso de las actividades donde una extensión puede tanto agregar una actividad nueva como modificar una ya existente en el modelo de proceso Factor. Por lo anterior, las clases navegacionales de la porción del modelo para las extensiones no son las mismas que para el proceso, pueden tener los mismos atributos pero no siempre los tendrán todos. Siguiendo con el ejemplo de las actividades, si se está agregando una actividad nueva a una extensión, ésta deberá tener todos los atributos, pero si se está modificando una actividad, ésta solo tendrá los atributos que se están modificando; por dicha razón se utiliza [0..1] para cada atributo ya que puede aparecer (caso en 1) o no aparecer (caso en 0).

Otra diferencia entre proceso y extensión es que no existe una presentación en diapositivas exclusiva para las extensiones ya que el mecanismo fue descrito brevemente en la presentación del proceso. La clase navegacional proceso se sustituye por la de extensión y aparece el atributo descripción. Desde la extensión se puede navegar hacia la agenda de

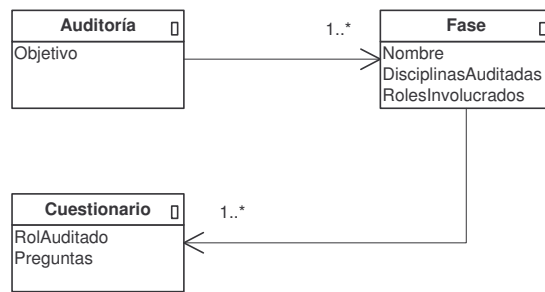


Figura 11. Modelo de la Estructura de Navegación de las auditorías.

actividades y entregables específica tal y como se hace en el proceso. Se puede acceder también tanto a la dimensión Tiempo como a la dimensión del Modelo, pero existen algunas diferencias tales como que la fase no tiene introducción, ni objetivos, ni actividades críticas ni no críticas. Esto se debe a que por ser una extensión está heredando esta información de su proceso padre. Algo análogo ocurre con las demás clases navegacionales cuyos atributos no son modificables. La última diferencia existente es que en la porción del modelo correspondiente a las extensiones, existe una nueva clase navegacional que representa la combinación de roles propuesta para cada extensión a la cual se puede llegar desde la dimensión del Modelo.

Por último, en la Figura 11 se presenta la parte del modelo que corresponde a las auditorías. Las auditorías tienen un objetivo y se puede navegar hacia las fases en las que se realizan.

Las fases tienen como atributos un nombre, las disciplinas a auditar y los roles involucrados en la auditoría. Luego, desde cada fase se puede navegar hacia los cuestionarios que deben ser completados, donde cada cuestionario es específico para un rol auditado y contiene las preguntas que deben ser contestadas.

El mantenimiento del modelo de proceso se reduce al mantenimiento del sitio web. Como se mencionó antes, agregar o modificar cualquier aspecto del modelo de proceso Factor involucra analizar el impacto del cambio en las extensiones existentes. A nivel del sitio web agregar una actividad, rol, entregable o plantilla no implica más que agregar una página y agregar referencias a ella donde sea necesario. Modificar cualquiera de estas características implica modificaciones en las páginas web que no son de mayor envergadura. A su vez, el agregado de una extensión en el sitio se traduce simplemente en agregar páginas web que cubran los aspectos correspondientes.

Capítulo 5

Prueba del Modelo de Proceso

En este capítulo se presenta la prueba del modelo de proceso realizada en el segundo semestre de 2004. El modelo de proceso Factor fue probado durante el curso Proyecto de Ingeniería de Software para el que fue creado. Fue utilizado por todos los grupos que cursaron la asignatura constituyendo un total de diez grupos de los cuales siete utilizaron Java como tecnología de desarrollo, uno utilizó la tecnología .NET y dos utilizaron GeneXus. En la Tabla 2 podemos observar la composición y las características de cada grupo.

Tanto los siete grupos Java como el grupo .NET utilizaron el Proceso para desarrollo Orientado a Objetos que incluye el modelo de proceso Factor y la extensión para desarrollo OO. Por otro lado, los tres grupos restantes utilizaron el Proceso para desarrollo con GeneXus que incluye el modelo de proceso Factor y la extensión para desarrollo GeneXus. De los diez grupos, hay dos de ellos, el 5 y el 6, que si bien utilizaron como guía el modelo de proceso, no resultan de interés para el análisis de la puesta en práctica ya que se enmarcaban en una experiencia de captura de requerimientos a distancia para lo cual no fue pensado este modelo y que se encuentra en estudio en otro proyecto. Por otra parte, el grupo 10 tenía como objetivo extender un producto existente para lo que tampoco esta pensado el modelo construido, por lo que también queda fuera del alcance el análisis de su experiencia.

El presente Capítulo se estructura de la siguiente manera. La Sección 5.1 describe la presentación del modelo de proceso a los involucrados, realizada en forma previa al comienzo del curso. La Sección 5.2 presenta las auditorías realizadas a los grupos durante el curso. Luego, un análisis de los indicadores de esfuerzo y tamaño es presentado en la Sección 5.3. Por su parte, la Sección 5.4 detalla los resultados obtenidos en la aplicación del modelo de proceso en base a la satisfacción de estudiantes y clientes. Finalmente, la Sección 5.5 concluye presentando los ajustes que fueron realizados al modelo Factor y sus extensiones.

5.1 Presentación a los involucrados

El modelo de proceso Factor fue presentado en una primera instancia el 27 de Julio de 2004 a los docentes del curso y a los directores de proyectos. Con tal motivo, se realizó una presentación con diapositivas en la que se dio una visión general de las características más relevantes del modelo, la forma de trabajo, y los elementos que lo componen.

Posteriormente, el 2 de agosto de 2004 se presentó el modelo a los estudiantes del curso mediante una charla introductoria, utilizando diapositivas similares a las anteriores y permitiendo las consultas de los asistentes. En ésta charla participaron también los docentes del curso.

Número de Grupo	Lenguaje	Cantidad	Director	Cliente
1	Java	11	D. Vallespir	CABAL
2	Java	11	J. Goyoaga	SCI
3	Java	11	J. Goyoaga	GRIS
4	Java	11	A. Delgado	CEI
5	Java	11	J. Triñanes	Universidad Politécnica de Madrid
6	Java	10	J. Triñanes	Universidad Politécnica de Madrid
7	Java	10	M. Freira	Link All
8	.NET	11	J. Tocar	CORREO
9	GeneXus	13	B. Perez	CORREO
10	GeneXus	12	L. Bustos	SECIU

Tabla 2. Composición de los grupos.

Las diapositivas utilizadas en ambas presentaciones se encuentran disponibles en el CD del proyecto.

Luego de la presentación a los estudiantes, el sitio web del modelo quedó disponible on-line para su consulta. Asimismo, quedó habilitado el newsgroup de la asignatura, que, entre otras, fue utilizado como mecanismo de consulta sobre el modelo por parte de los estudiantes y para realizar las correspondientes respuestas por nuestra parte.

5.2 Auditorías

Las auditorías tienen como objetivo contrastar el modelo de proceso propuesto con el desarrollo real de cada proyecto del curso. De esta forma se puede conocer el grado de apego al proceso que se está teniendo, la causa de los desvíos si los hubiera, y las actividades que, aún no estando contempladas en el modelo, cada equipo está llevando a cabo. Las auditorías sirven de guía y apoyo a los equipos que están realizando su proyecto y brindan la posibilidad de detectar elementos a ajustar y/o mejorar en el modelo.

En esta Sección se describen las dos auditorías realizadas y los resultados obtenidos en las mismas. Los cuestionarios realizados para la preparación de las auditorías, el resumen de las respuestas brindadas por los grupos auditados, así como los informes realizados para los mismos se encuentran disponibles en el CD del proyecto.

5.2.1 Auditoría de Fase Inicial

La primera auditoría se realizó en la quinta semana del curso, correspondiente al fin de Fase Inicial. Las disciplinas consideradas de vital importancia en esta fase, y que por lo tanto se auditaron, fueron Requerimientos, Diseño, Gestión de Proyecto, Gestión de Calidad, y Gestión de Configuración y Control de Cambios.

Cinco grupos fueron seleccionados para ser auditados. El criterio de selección se basó principalmente en el interés desde la perspectiva del modelo de proceso. En algunos casos, la conveniencia de ayudar a alguno de los grupos también fue utilizada como criterio. Los grupos seleccionados para la Auditoría de Fase Inicial fueron los grupos 1 y 7 con desarrollo en Java, el grupo 8 con desarrollo en .NET y los grupos 9 y 10 con desarrollo en GeneXus. Los grupos elegidos abarcan todas las extensiones y tecnologías. Por su parte,

auditar el grupo 10, cuyo desarrollo implicaba la mejora de un sistema existente, brindó una perspectiva diferente de aplicación.

Para la auditoría se elaboraron cuestionarios específicos para cada rol auditado los que debieron ser completados y entregados en forma previa a la auditoría. Se convocó a una reunión donde debieron asistir todos los integrantes cuyo rol se auditaba. La auditoría consistió en la realización de preguntas para contrastar entre sí lo contestado por cada uno de los integrantes, y de preguntas adicionales contenidas en un cuestionario aparte y particular para cada una de las extensiones. Este cuestionario se confeccionó con preguntas referentes al desarrollo del proyecto en general, y a cuestiones específicas de cada extensión (OO y Gx).

Para la auditoría de Fase Inicial los roles auditados fueron Analistas, Arquitecto, Administrador, Responsable de SQA y Responsable de SCM. Los cuestionarios fueron puestos a disposición en el sitio web del modelo y se solicitó que fueran completados luego de la reunión de fin de fase de cada grupo y enviados vía mail por el administrador. Aunque todos los grupos debieron completar el cuestionario, solo se realizó un resumen de las respuestas, una reunión de auditoría, y un informe final, para los grupos seleccionados para ser auditados.

5.2.2 Auditoría de Fase de Elaboración

La segunda auditoría se realizó en la novena semana del curso, correspondiente al fin de Fase de Elaboración. Las disciplinas consideradas de importancia en esta oportunidad fueron Implementación, Gestión de Proyecto, Verificación, y Gestión de Configuración y Control de Cambios.

Se seleccionaron nuevamente cinco grupos para ser auditados, siguiendo los mismos criterios que en la primera auditoría, intentando a su vez que la mayoría de los grupos fueran auditados al menos una vez. Los grupos seleccionados para la Auditoría de Fase de Elaboración fueron los grupos 2, 3 y 4 con desarrollo en Java, el grupo 8 con desarrollo en .NET y el grupo 10 con desarrollo en GeneXus.

Nuevamente se elaboraron cuestionarios específicos para cada rol auditado y se convocó a una reunión donde debieron asistir todos los integrantes cuyo rol se auditaba. En esta ocasión los roles auditados fueron Implementadores, Administrador, Responsable de Verificación y Responsable de SCM.

5.3 Indicadores

En esta Sección se estudia algunos indicadores de modo de contar con herramientas objetivas para evaluar el proceso. Los indicadores elegidos son el esfuerzo y el tamaño.

Por las características de algunos de los proyectos llevados adelante, resulta de interés el estudio de los resultados de cinco de los grupos, tres de ellos con desarrollo en Java, uno con desarrollo en .NET y el último con desarrollo en GeneXus.

5.3.1 Esfuerzo

En la Tabla 3 se presenta, para los equipos cuyos resultados se están analizando, mediciones de esfuerzo interesantes.

Número de Grupo	Lenguaje	Total de horas	Promedio de horas por integrante	Promedio de horas del Responsable de Verificación - SQA - Analista
2	Java	4036.1	26,2	27,8
3	Java	3123.2	20,3	20
4	Java	4091.5	24,7	24,9
8	.NET	3323.1	21,6	23,8
9	GeneXus	4149.3	22,9	21,7

Tabla 3. Mediciones de esfuerzo.

Como se puede observar, en los cinco grupos que tomamos como muestra, se superan ampliamente las 15 horas de dedicación previstas para el curso, realizando en todos los casos más de 20 horas promedio. Esta situación no es nueva en el curso, pues en los resúmenes de horas de dedicación de años anteriores puede notarse esta tendencia. Esta alta dedicación se explica por la gran carga horaria que dedican los estudiantes al área de implementación y a la de gestión, motivados principalmente por obtener productos que satisfagan los requerimientos del cliente.

En particular es de interés el esfuerzo invertido en la combinación de roles Responsable de Verificación - SQA - Analista, ya que es una combinación que se introdujo nueva en este modelo de proceso. El esfuerzo invertido en esta combinación de roles es en general algo superior al promedio lo que se puede explicar por la dificultad en la comprensión de las responsabilidades de cada uno de los roles que conforman la combinación. El rol de Analista en esta combinación cumple el objetivo de brindar en el análisis de los requerimientos la visión del área de calidad y de la verificación. Por su parte, el rol de Responsable de Verificación debe gestionar la verificación en el proyecto, es quien se debe encargar de planificar, diseñar, implementar y evaluar las pruebas en conjunto con sus asistentes de verificación. A su vez, el Responsable de Verificación debe generar los planes y los casos de prueba, implementar los procedimientos de prueba, y evaluar la cobertura, los resultados y la efectividad de las pruebas. Debe también hacer los informes de evaluación de las pruebas. El Responsable de SQA, por su lado, debe asegurar la calidad de los productos generados en el proyecto y del proceso utilizado. Para asegurar la calidad debe revisar la calidad de los entregables de planificación del proyecto y los entregables de valoración del proyecto. Además revisa el nivel de apego al modelo de proceso de desarrollo de software y a los planes de Verificación, Gestión de Proyecto y Gestión de Calidad, documentando las desviaciones encontradas. Debe conocer los conceptos y técnicas de Gestión de Calidad del Software y debe identificar las propiedades de calidad que deben cumplir los productos del proyecto.

A los estudiantes que realizaron esta combinación de roles les costó entender las responsabilidades de cada rol y separarlas en forma adecuada, por lo que se vio invertida una mayor cantidad de esfuerzo tanto en el área de Verificación como en el área de Calidad.

5.3.2 Tamaño del Producto

En la Tabla 4 se puede ver un resumen de las mediciones de tamaño de los productos finales realizadas por los equipos cuyos resultados se están analizando.

Número de Grupo	Lenguaje	Mediciones de Tamaño
2	Java	24.466 (LOC's Java)

3	Java	54.016 (LOC's Java)
4	Java	21.783 (LOC's Java) + 11.662 (LOC's Jsp)
8	.NET	No realizaron mediciones
9	GeneXus	1.984 (LOC's C#) + 291 Objetos GeneXus

Tabla 4. Mediciones de tamaño de los productos.

Como se puede observar en la tabla, los productos obtenidos tienen tamaños considerables, factor que esta presente también en años anteriores. Se debe notar aquí que productos de estos tamaños construidos por equipos numerosos serían difíciles de obtener sin un modelo de proceso que guía su trabajo, interacción y organización.

5.4 Resultados

Como se mencionó anteriormente, el hecho de que el modelo sea de aplicación en un curso es uno de los motivos por el cual este proceso se torna “pesado”. Por otro lado, el equipo de trabajo no cuenta con la suficiente experiencia y formación como para hacer una correcta y adecuada instanciación del mismo. Además, muchos de los artefactos sugeridos surgen como parte de la formación que se pretende ofrecer y muchos otros forman parte de las herramientas con que cuentan los docentes para evaluar el desempeño de los grupos.

El hecho de unificar el material existente en un solo modelo tiene como consecuencia directa la disminución del esfuerzo de mantenimiento. Antes de contar con el modelo de proceso Factor y sus extensiones, agregar una actividad o modificar una ya existente debía realizarse en cada uno de los modelos multiplicando el esfuerzo necesario y dando lugar a errores e inconsistencias. Con el modelo de proceso Factor y sus extensiones este esfuerzo y riesgo de errores se ve sensiblemente disminuido, aunque al agregar o modificar aspectos del Factor se debe analizar como influyen en las extensiones existentes.

Es de interés evaluar los resultados desde el punto de vista del cliente y de los grupos que realizaron la experiencia y que utilizaron el modelo de proceso. Esta Sección estudia los niveles de satisfacción de ambas partes.

5.4.1 Satisfacción de los grupos

Con el objetivo de evaluar la satisfacción de los grupos con el modelo de proceso propuesto se elaboró una encuesta que fue respondida por cada uno de los integrantes de cada equipo de trabajo. La encuesta contiene preguntas que permitan evaluar diversos aspectos tanto del entendimiento logrado del modelo de proceso como de la adecuación o apego al mismo. Se pretende evaluar si el modelo de proceso sirvió como guía a los estudiantes para llevar adelante sus proyectos, así como saber cuál fue el resultado obtenido, es decir el producto final.

El estudio de satisfacción se realiza para siete de los diez grupos que realizaron la experiencia que hacen un total de setenta y cinco estudiantes. No se incluyen en este estudio a los grupos 5 y 6, que si bien utilizaron como guía el modelo de proceso propuesto, se encontraban enmarcados en una experiencia de requerimientos a distancia para lo que no fue pensado este modelo. Tampoco se incluye en este estudio al grupo 10 puesto que si bien, como en el caso anterior, utilizaron como guía el modelo de proceso

propuesto, la experiencia consistió en la extensión/modificación de un producto ya existente para lo que tampoco está pensado este modelo. Como se mencionó anteriormente, tanto los grupos que desarrollaron con Java como el grupo que desarrolló en .NET utilizaron el Proceso para desarrollo de software Orientado a Objetos, mientras que el restante grupo utilizó el Proceso para desarrollo de software con GeneXus.

Los resultados de la encuesta realizada se presentan mediante tablas las cuales resumen las respuestas obtenidas en aquellas preguntas que consideran los aspectos más importantes. Estos aspectos fueron evaluados en una escala de 1 a 5, cuyo significado se adecua a cada pregunta realizada. El cuestionario completo, así como también las respuestas a los mismos que brindó cada grupo, se encuentran disponibles en el CD del proyecto.

En la Tabla 5 se puede apreciar un resumen de las respuestas obtenidas a las preguntas sobre los roles, las actividades relacionadas y el esfuerzo incurrido. De este resumen se destacan varias cosas importantes. El primer aspecto que puede destacarse es que el 96% de los estudiantes consideran la definición de las actividades entre aceptable y muy buena y que el 82,7% de los estudiantes entendió entre bastante y mucho las actividades propuestas. Sobre el esfuerzo incurrido se puede observar y confirmar la tendencia de que los estudiantes se sienten bastante cargados de trabajo, punto ya analizado a lo largo de este informe donde se había notado que el modelo creado no resultaría para los estudiantes sensiblemente más liviano que los existentes.

El tercer aspecto importante es que al 68% de los estudiantes le pareció adecuada la combinación de roles propuesta. En general, si se observa las respuestas a los cuestionarios completos (en estas tablas no se muestran los comentarios realizados por los estudiantes) se puede apreciar que la combinación de roles más conflictiva fue la que ya analizamos anteriormente: Responsable de Verificación - SQA - Analista. Los estudiantes que llevaron adelante este rol se sintieron muy sobrecargados de trabajo y en general, si bien los datos obtenidos muestran que la cantidad promedio de horas invertidas fue cercana a los promedios de los demás integrantes, los equipos sintieron una sobrecarga en esa combinación de roles.

En la Tabla 6 se muestra un resumen de las respuestas obtenidas al preguntar sobre los entregables del modelo. En lo referente a las plantillas propuestas, un 53,4% de los estudiantes respondió que le fueron entre bastante y muy útiles mientras que un 41,3 respondió que la utilidad había sido media. Ninguno de los encuestados respondió que las plantillas no le habían resultado útiles. Es importante destacar también que el 96% de los estudiantes pudo comprender entre medio y mucho las plantillas propuestas.

Un 94,7% respondió que fue necesario realizar modificaciones a las plantillas. Este dato no es considerado negativo ya que las plantillas, así como el modelo propiamente, deben servir de guías para los estudiantes que cursan la asignatura, por lo que el contenido de las mismas debe cambiar en dependencia del proyecto. El contenido de los documentos debe ser modificado por cada equipo en función del producto que están construyendo, y, como ya se mencionó anteriormente, ésta es una de las prácticas que se recomienda para lograr un proceso liviano.

Escala	Pregunta	% de Respuestas	% 1	% 2	% 3	% 4	% 5
	Los Roles						
	Sobre las actividades propuestas para su rol						

CT	¿Las leyó?	100	0	0	5,3	64	30,7
CL	¿Cómo le resultó la definición de las mismas?	100	0	4	53,3	40	2,7
CT	¿Las entendió?	100	0	0	17,3	68	14,7
CT	¿Las realizó?	100	0	0	18,7	57,3	24
CT	¿Se apegó a las descripciones?	100	0	1,3	33,3	54,7	10,7
	Sobre el esfuerzo incurrido						
	¿Cómo se sintió con respecto a la carga de trabajo?						
CO	Fase I - Inicial	100	13,3	32	34,7	16	4
CO	Fase II - Elaboración	100	17,3	49,3	32	1,4	0
CO	Fase III - Construcción	100	25,3	41,3	25,4	8	0
CO	Fase IV - Implantación	100	24	30,6	28	10,7	6,7
	Sobre la Combinación de Roles propuesta						
SN	¿Le pareció adecuada?	100	68	32			
SN	¿Asignaron Asistente de Arquitecto?	100	73,3	26,7			
CT	¿Fue de utilidad?	82,7	9,7	11,3	16,1	46,8	16,1

Tabla 5. Resumen de respuestas sobre los roles.

Escala	Pregunta	% de Respuestas	% 1	% 2	% 3	% 4	% 5
	Los Entregables						
	Sobre las plantillas						
CT	¿Le fueron de Utilidad?	100	0	5,3	41,3	36	17,4
CT	¿Las entendió?	100	0	4	42,6	46,7	6,7
CT	¿Tuvo que realizar modificaciones a las mismas?	100	5,3	44	33,4	16	1,3
	Sobre los entregables que fueron su responsabilidad						
SN	¿Realizó todos los entregables previstos en el modelo?	100	64	36			
CT	¿Pudo seguir la agenda propuesta?	100	0	1,3	29,3	64	5,4
SN	¿Tuvo que agregar entregables?	98,7	16,2	83,8			

Tabla 6. Resumen de respuestas sobre los entregables.

Un último aspecto a destacar de la Tabla 6 es que el 69,5% pudo seguir entre bastante y mucho la agenda de entregables propuesta y el 83,8% no tuvo necesidad de agregar ningún entregable.

Escala	Pregunta	% de Respuestas	% 1	% 2	% 3	% 4	% 5
	El Director						
CT	¿Cuan útil resultó el seguimiento realizado por el Director?	98,7	0	1,4	17,6	55,4	25,6
CT	¿Le pareció adecuada la frecuencia de las reuniones con el Director fijada en la agenda?	98,7	1,4	1,4	10,8	60,8	25,6
	Los Requerimientos y el Cliente						

CT	¿Las semanas marcadas para las presentaciones y validaciones con el Cliente le parecieron adecuadas?	100	0	5,3	17,3	65,4	12
SN	¿Le parecieron adecuados los entregables marcados para validar con el Cliente?	100	96	4			

Tabla 7. Resumen de respuestas sobre el director, los requerimientos y el cliente.

Escala	Pregunta	% de Respuestas	% 1	% 2	% 3	% 4	% 5
	El Producto						
CL	Cumplimiento del alcance planificado.	100	0	0	5,4	29,3	65,3
CL	Cumplimiento de los requerimientos de calidad establecidos por el cliente.	100	0	2,7	20	52	25,3
CL	Nivel de Correctitud logrado.	98,7	0	0	24,3	59,5	16,2
CL	Nivel de Funcionalidad logrado.	100	0	0	6,7	60	33,3
CL	Nivel de Confiabilidad logrado.	100	0	4	41,3	45,4	9,3
CL	Nivel de Robustez logrado.	100	2,7	12	40	36	9,3
CL	Nivel de Amigabilidad logrado.	100	0	1,3	18,7	50,7	29,3
CL	Nivel de Verificabilidad logrado.	100	0	5,3	30,7	52	12
CL	Nivel de Performance logrado.	100	0	9,3	33,4	44	13,3
CL	Nivel de Portabilidad logrado.	100	0	5,3	25,3	29,4	40
CL	Nivel de Interoperabilidad logrado.	98,7	2,7	12,2	24,3	43,2	17,6
SN	¿Se realizó la implantación del producto en el ambiente del cliente y las correspondientes pruebas de aceptación?	98,7	100	0			

Tabla 8. Resumen de respuestas sobre el producto.

En la Tabla 7 se muestra el resumen de respuestas obtenidas al preguntar sobre el director de proyecto, los requerimientos y el cliente. A un 86,4% de los estudiantes le resultó entre bastante y muy adecuada la frecuencia de las reuniones con el director de proyecto fijada en la agenda. Con respecto a los requerimientos y al cliente, se puede decir que a un 77,4% de los estudiantes le parecieron entre bastante y muy adecuadas las semanas marcadas en la agenda para la validación y presentaciones al cliente, además un 96% consideró adecuados los entregables marcados para validar.

En la Tabla 8 puede observarse el resumen de las respuestas obtenidas sobre el producto construido. Los aspectos más importantes a destacar en esta tabla son los porcentajes relacionados con el alcance planificado y el cumplimiento de los requerimientos del cliente. El 100% de los encuestados considera que se cumplió en forma entre aceptable y muy buena con el alcance planificado mientras que el 97,3% considera que el cumplimiento de los requerimientos de calidad establecidos por el cliente fue entre y aceptable y muy bueno.

Escala	Pregunta	% de Respuestas	% 1	% 2	% 3	% 4	% 5
	El modelo de proceso						
CT	Esfuerzo empleado en entender el trabajo correspondiente a su rol.	100	0	14,7	52	25,3	8
CT	Conocimiento de las actividades realizadas por los demás integrantes del equipo de trabajo.	100	0	5,3	33,4	48	13,3
CT	Apego del equipo al modelo de proceso	100	0	6,7	36	49,3	8

	propuesto						
CT	¿El modelo de proceso propuesto resultó útil como guía para el desarrollo del Proyecto?	100	0	5,3	33,3	56,1	5,3
CT	¿Considera adecuadas la cantidad de iteraciones por fase propuestas, así como su duración y objetivos?	100	0	9,3	32	49,4	9,3
	Las Auditorías						
CT	¿Fueron de utilidad al equipo las auditorías realizadas?	98,7	4	31,1	39,2	21,6	4,1

Tabla 9. Resumen de respuestas sobre el modelo de proceso y las auditorías.

En la Tabla 9 se aprecia un resumen de las respuestas obtenidas referentes al modelo de proceso específicamente y a las auditorías realizadas. Un porcentaje grande de los estudiantes encuestados considera que se debe emplear un esfuerzo entre medio y bastante para entender las responsabilidades de cada rol. Esto se debe a que por lo general el curso de Proyecto de Ingeniería de Software es la primera experiencia de este tipo en la que participan los estudiantes por lo que les cuesta adaptarse a la modalidad de trabajo en equipos tan numerosos donde cada uno tiene las responsabilidades especificadas.

En general pudieron apegarse bastante al modelo propuesto y el 61,4% considera que el modelo resultó entre bastante y muy útil como guía para el desarrollo del proyecto. Además, en general, también estuvieron de acuerdo con las iteraciones propuestas y su duración y objetivos.

Por último, con respecto a las auditorías podemos decir que a un 96% de los encuestados le resultaron de alguna utilidad las auditorías realizadas. Es importante resaltar aquí que el objetivo principal de las auditorías no es ayudar a los equipos aunque sí es un objetivo secundario en caso de que éstos lo necesiten.

Como resumen de todo lo anterior, en la Tabla 10 se presentan las respuestas obtenidas sobre la satisfacción con el resultado del proyecto y con el proceso utilizado. El 33,3% de los estudiantes se considera satisfecho con el resultado del proyecto mientras que el 61,3% se considera muy satisfecho. Por otro lado, el 28% de los estudiantes se manifestó neutral con respecto a la satisfacción del equipo de trabajo con el proceso utilizado mientras que un 56% se manifestó satisfecho y un 9,3% se manifestó muy satisfecho.

Las distintas escalas utilizadas para cada una de las preguntas del cuestionario se encuentran detalladas en la Tabla 11.

Escala	Pregunta	% de Respuestas	% 1	% 2	% 3	% 4	% 5
	El Proyecto						
SA	Satisfacción del equipo de trabajo con el resultado del proyecto.	100	0	2,7	2,7	33,3	61,3
SA	Satisfacción del equipo de trabajo con el proceso utilizado.	100	0	6,7	28	56	9,3

Tabla 10. Resumen de respuestas sobre el proyecto.

Escala Cuantitativa	Escala Cualitativa (CL)	Escala de Comodidad (CO)	Escala Si o No	Escala de Satisfacción
---------------------	-------------------------	--------------------------	----------------	------------------------

(CT)			(SN)	(SA)
1 - Nada	1 - Malo	1 - Muy Apresurado	Si	1 - Muy Insatisfecho
2 - Poco	2 - Regular	2 - Apresurado	No	2 - Insatisfecho
3 - Medio	3 - Aceptable	3 - Normal		3 - Neutral
4 - Bastante	4 - Bueno	4 - Cómodo		4 - Satisfecho
5 - Mucho	5 - Muy Bueno	5 - Muy Cómodo		5 - Muy satisfecho

Tabla 11. Escalas utilizadas.

5.4.2 Comparación con experiencias anteriores

Para tener otra perspectiva de la puesta a prueba del modelo y sus resultados, se realiza una comparación de los datos obtenidos en esta experiencia con los datos obtenidos en la experiencia 2002 con el modelo de proceso Java. Solo se realiza la comparación con los datos del 2002 ya que no se cuenta con la información de otros años. Los datos utilizados corresponden a un informe del procesamiento de los cuestionarios contestados por los estudiantes realizado por el Gris [\[24\]](#).

Es importante destacar que los cuestionarios entregados a los estudiantes en la experiencia 2002 con el modelo de proceso Java y en la experiencia 2004 con el modelo de proceso Factor y las extensiones no son iguales, por lo que solo puede compararse los resultados de aquellas preguntas que por ser iguales o similares, pudo trazarse una correspondencia entre ambos cuestionarios.

En la Tabla 12 podemos ver un resumen de la información de los grupos que cursaron la asignatura en el año 2002. Como puede observarse, en general los grupos eran más grandes que en el año 2004, pero sin embargo, el promedio de horas por integrante en el 2002 es más alto que en el 2004. Si hacemos un promedio de los promedios de horas por integrante en cada año resulta que en el 2002 el promedio de horas invertidas por cada estudiante fue de 26 horas mientras que el 2004 fue de 23,6. En ambos casos está muy por encima de las 15 horas previstas para el curso, pero como se mencionó anteriormente, esta es una situación que se viene dando año a año. Incluso en el mismo informe del 2002 se sugiere revisar la cantidad de créditos otorgados en el curso.

Número de Grupo	Lenguaje	Cantidad	Director	Cliente	Total de horas	Promedio de horas por integrante
1	Java	12	J. Goyoaga	IdeaSoft (Botto)	3500	24,3
2	Java	11	J. Goyoaga	IdeaSoft (Botto)	4073	30,8
3	Java	13	M. Freira	CSI (Motz)	3581	22,9
4	Java	13	M. Freira	CSI (Ruggia)	3716	23,8
5	Java	13	J. Triñanes	CSI (Motz)	4914	31,5
6	Java	14	J. Triñanes	CSI (Ruggia)	5000	29,7
7	Java	14	J. Triñanes	CSI (Ruggia)	3200	19,0

Tabla 12. Resumen de información de los grupos del 2002.

Número de	Lenguaje	Mediciones de Tamaño
-----------	----------	----------------------

Grupo		
1	Java	14.771 (LOC's Java s/c)
2	Java	21.521 (LOC's Java s/c)
3	Java	14.025 (LOC's Java s/c)
4	Java	26.500 (LOC's Java s/c)
5	Java	33.000 (LOC's Java s/c)
6	Java	26.000 (LOC's Java s/c)
7	Java	24.000 (LOC's Java s/c)

Tabla 13. Mediciones de tamaño para los productos del 2002.

El dato del promedio de horas por estudiantes es muy importante pero es muy difícil de comparar en forma fría y objetiva, ya que las horas trabajadas dependen no solo de las características particulares de cada proyecto sino también de las características particulares de cada grupo. Estas particularidades aportan elementos subjetivos cuya influencia es difícil de medir con exactitud. En conclusión, lo que se pretende notar es que si bien, según los datos con que se cuenta el promedio de horas bajó con el modelo de proceso Factor y las extensiones, esto no indica un resultado absoluto, sino que por el contrario, se requerirá de más experiencias de prueba para lograr resultados más precisos.

En la Tabla 13 se observan las mediciones de tamaño realizadas por los grupos del 2002. Comparando los grupos del año 2002 con los grupos que usaron la extensión OO en el año 2004, se observa que la cantidad de líneas de código producida es bastante superior en el último. Mientras que el promedio en el 2002 fue de 17646,4 LOC's, en el 2004 tenemos un promedio de 37309 LOC's Java y JSP calculado en base a los grupos de los que se presentaron las mediciones en la sección anterior. Resulta sospechosa la gran diferencia en el promedio de LOC's. Una posible causa es que las mediciones realizadas por los grupos en el 2004 incluyeran comentarios a diferencia de las realizadas en el 2002, pero esto no es posible afirmarlo. En particular el grupo 3 del año 2004 presentó una medición de LOC's producidos muy alta, que es la que está disparando el promedio calculado; ese grupo afirmó haber realizado la medición sin tener en cuenta los comentarios.

Se analizan a continuación los resultados obtenidos para las preguntas del cuestionario. Como se mencionó anteriormente, no fue posible trazar una correspondencia para todas las preguntas del 2002 con las del 2004. Igualmente, se realiza el análisis de los datos con la información existente.

Pregunta	% de Respuestas	% 1	% 2	% 3	% 4	% 5	Prom. 2002	Prom. 2004
Los Roles								
Sobre las actividades propuestas para su rol								
¿Las leyó?	100	0	0	4,4	37,8	57,8	4,4	4,3
¿Cómo le resultó la definición de las mismas?	100	1,1	2,2	25,6	53,3	17,8	3,8	3,4
¿Las entendió?	100	0	1,1	7,8	52,2	38,9	4,3	4
¿Las realizó?	100	0	0	2,2	55,6	42,2	4,4	4,1
¿Se apegó a las descripciones?	100	0	0	18,9	57,8	23,3	4	3,7

Tabla 14. Resumen de respuestas sobre los roles para el 2002.

Pregunta	% de Respuestas	% 1	% 2	% 3	% 4	% 5
Sobre los entregables que fueron su responsabilidad						
¿Pudo seguir la agenda propuesta?	96,7	39	61			

Tabla 15. Resumen de respuestas sobre los entregables para el 2002.

En la Tabla 14 se muestra el resumen de las respuestas sobre los roles. En este caso no se encontró información sobre el esfuerzo incurrido en cada fase ni la opinión de los estudiantes sobre la combinación de roles propuesta.

Según los datos y comparando fríamente los promedios para cada año se puede decir que en general en el 2004 se leyeron menos las actividades, se entendieron menos, se realizaron menos y hubo un menor apego a las descripciones. De todas formas, las diferencias existentes no son de gran magnitud. Seguramente estas variaciones sean explicadas, en parte, por la subjetividad inherente a las respuestas. Por otra parte, puede deberse al hecho de que en el 2002 el modelo de proceso utilizado se presentó en el curso de Introducción a la Ingeniería de Software en el semestre previo al curso Proyecto de Ingeniería de Software, por lo que los estudiantes tuvieron un contacto previo que les permitió conocer más las actividades, que seguramente se tradujo en mejores índices de lectura de las mismas, realización y apego a las descripciones.

En la Tabla 15 se presenta el resumen de las respuestas relacionadas con los entregables. En este caso no se encontró información que pudiera servir para realizar comparaciones acerca del contenido de las plantillas, ni tampoco sobre si se habían realizado o no todos los entregables previstos en el modelo.

Con respecto a la agenda propuesta se puede decir que en el 2004 un 69,5% pudo seguirla entre bastante y mucho mientras que en el 2002 un 61% de los estudiantes manifestó haber podido seguir la agenda y un 39% dijo no haber podido seguirla. Un detalle importante para analizar esta comparación es que, además de contar con la subjetividad inherente a las respuestas brindadas por los estudiantes, se debe tener en cuenta la diferencia de escalas existente, ya que en el 2002 se utilizó una escala de Si o No mientras que en el 2004 se utilizó una escala cuantitativa.

Pregunta	% de Respuestas	% 1	% 2	% 3	% 4	% 5	Prom 2002	Prom 2004
El Director								
¿Cuan útil resultó el seguimiento realizado por el Director?	95,6	10,5	10,5	27,9	38,3	12,8	3,3	4,1
Los Requerimientos y el Cliente								
¿Las semanas marcadas para las presentaciones y validaciones con el Cliente le parecieron adecuadas?	95,6	1,2	1,2	22,1	55,8	19,7	3,9	3,8
¿Le parecieron adecuados los entregables marcados para validar con el Cliente?	96,7	0	0	15	58,6	26,4	4,1	5

Tabla 16. Resumen de respuestas sobre el director, los requerimientos y el cliente para el 2002.

En la Tabla 16 se muestra el resumen de las respuestas sobre el director, los requerimientos y el cliente. No se encontró información sobre la opinión de los estudiantes sobre la frecuencia de las reuniones con el director fijada en la agenda.

Los promedios indican que hubo una mejora en el seguimiento realizado por los directores de proyecto y una mayor aceptación de los entregables marcados para validar con el cliente. Nuevamente es importante tener en cuenta a la hora de analizar los datos la subjetividad de las respuestas brindadas.

En la Tabla 17 se brinda un resumen de las respuestas obtenidas sobre el producto de cada grupo. No se encontró información sobre el cumplimiento de los requerimientos de calidad establecidos por el cliente en el 2002. De las propiedades de calidad del producto solo se muestran los promedios de cada año.

El promedio de cumplimiento del alcance en ambos años es igual. A grandes rasgos podemos decir que los promedios relacionados con las propiedades de calidad son muy similares para ambos años. Existen diferencias en algunos casos que por ser tan menores hacen pensar que se deban nuevamente a la subjetividad de las respuestas.

Con respecto a la implantación y pruebas de aceptación de los productos en el año 2002 un 90,6% de los estudiantes contestó que sí se realizaron mientras que en el 2004 fue el 98,7% los que dijeron que sí se habían realizado. Aunque no es un resultado concluyente, como en los casos anteriores, podría decirse que aumentó el grado de implantación de los productos.

En la Tabla 18 puede verse un resumen de las respuestas sobre el modelo de proceso y las auditorías.

Existe una leve diferencia en el esfuerzo empleado en entender el trabajo correspondiente al rol. Esta diferencia puede explicarse por las diferencias existentes en lo que tiene que ver con las extensiones, es decir, puede que en el 2004 el modelo de proceso nuevo al principio les insuma un esfuerzo adicional en la comprensión del uso correcto de las extensiones. La diferencia puede deberse también, como en los casos anteriores, simplemente a la subjetividad.

Pregunta	% de Respuestas	% 1	% 2	% 3	% 4	% 5	Prom 2002	Prom 2004
El Producto								
Cumplimiento del alcance planificado.	100	0	0	1,1	31,1	67,8	4,6	4,6
Nivel de Correctitud logrado.							4,1	3,9
Nivel de Funcionalidad logrado.							4,5	4,3
Nivel de Confiabilidad logrado.							3,8	3,6
Nivel de Robustez logrado.							3,3	3,4
Nivel de Amigabilidad logrado.							4,4	4,1
Nivel de Verificabilidad logrado.							3,7	3,7
Nivel de Performance logrado.							3,5	3,6
Nivel de Portabilidad logrado.							4,4	4
Nivel de Interoperabilidad logrado.							3,4	3,6
¿Se realizó la implantación del	94.4	90,6	9,4					

producto en el ambiente del cliente y las correspondientes pruebas de aceptación?								
---	--	--	--	--	--	--	--	--

Tabla 17. Resumen de respuestas sobre el producto para el 2002.

Es difícil comparar el conocimiento de las actividades realizadas por los demás integrantes del equipo ya que se utilizaron escalas distintas usándose una escala de Sí o No en el 2002 y una escala cuantitativa en el 2004. En el 2002 el 97,8% de los estudiantes contestó conocer las actividades realizadas por lo restantes integrantes del equipo mientras que en el 2004 solo el 38,7% contestó conocer entre bastante y mucho las actividades que realizaban los demás integrantes del equipo y otro 33,4% dijo conocer un término medio. Se ve reflejado aquí el desconocimiento de los estudiantes del modelo de proceso utilizado por lo que se sugiere brindar más apoyo en el aprendizaje y la comprensión del mismo.

En cuanto al apego al modelo de proceso, los valores son muy similares y la diferencia existente no es de importancia. Los equipos no tienen obligación de apegarse totalmente al modelo de proceso sino más bien deben seguirlo tomándolo como guía y haciendo la instanciación necesaria del mismo de acuerdo al proyecto que estén llevando adelante. Los puntos de diferencia pueden estar explicados en que el grupo 1 del año 2004 tomó el proceso como guía estrictamente, y en muchos casos no se apegó al mismo, contando, claro está, con el apoyo del director del proyecto.

En el 2002 un 97,8% respondió que el modelo de proceso le había resultado útil como guía para el desarrollo del proyecto, mientras que en el 2004 un 94,7% respondió que el proceso había sido entre bastante y muy útil. Nuevamente aquí juega un rol fundamental la subjetividad en las respuestas. Por otro lado, la posibilidad de respuestas que brinda una escala cuantitativa frente a una escala de Sí o No es mucho más amplia y en general la tendencia con este tipo de escalas es a no contestar los extremos. Una vez dicho esto, se analizan las respuestas obtenidas. Las respuestas en 1 y 2 podemos considerarlas como un no rotundo mientras que las respuestas en 4 y 5 pueden consideradas como un sí rotundo. El problema se plantea con las respuestas 3 que podrían ser tomadas tanto por sí como por no. Por el conocimiento que tenemos de la experiencia 2004 podemos decir que el 33,3% de los estudiantes que contestaron con un 3 a la pregunta seguramente tienden en su mayoría al sí más que al no, pero debemos sacar una conclusión acerca de este número. La conclusión que podemos sacar es que la utilidad del modelo como guía se ve reducida por la falta de conocimiento del mismo.

Pregunta	% de Respuestas	% 1	% 2	% 3	% 4	% 5	Prom. 2002	Prom. 2004
El modelo de proceso								
Esfuerzo empleado en entender el trabajo correspondiente a su rol.	100	4,4	22,2	41,2	28,9	3,3	3	3,3
Conocimiento de las actividades realizadas por los demás integrantes del equipo de trabajo.	100	97,8	2,2					3,7
Apego del equipo al modelo de proceso propuesto	100	1,1	2,2	26,7	48,9	21,1	3,9	3,6
¿El modelo de proceso propuesto resultó útil como guía para el desarrollo del Proyecto?	100	97,8	2,2					3,6
¿Considera adecuadas la	97,8	58	42					

cantidad de iteraciones por fase propuestas, así como su duración y objetivos?								3,6
Las Auditorías								
¿Fueron de utilidad al equipo las auditorías realizadas?	98,9	5,6	5,6	27	42,7	19,1	3,6	2,9

Tabla 18. Resumen de respuestas sobre el modelo de proceso y las auditorías para el 2002.

Por consiguiente, los estudiantes necesitan más apoyo por parte de los docentes para conocer el modelo y entenderlo de forma de poder utilizarlo como la guía que pretende ser. Igualmente una nueva experiencia de prueba del modelo podría arrojar más información para realizar ajustes que permitirían mejorar este indicador.

En el año 2002 solo el 54% de los estudiantes consideró adecuadas la cantidad de iteraciones por fase propuestas así como su duración y objetivos mientras que en el 2004 el 90,7% de los estudiantes respondió que eran entre bastante y muy adecuadas. Estos resultados son muy difíciles de comparar ya que en el 2002 ocurrió una huelga general que obligó a recortar el curso y a eliminar dos semanas de las catorce previstas. En el 2004 un 32% de los estudiantes contestó con un término medio a la pregunta (con un 3). Esto obliga a pensar en la posibilidad de probar alguna otra combinación de fases; por ejemplo acortar la fase inicial en una semana y la fase de transición en una semana agregándoselas a la fase de elaboración y construcción respectivamente.

Por último, con respecto a la utilidad de las auditorías para los grupos, el nivel del 2002 es bastante superior al del 2004. Esto puede explicarse por la experiencia, sobre todo como docentes, de los auditores del 2002, frente a la escasa experiencia en esta área de quien realizó las auditorías en el 2004. Aunque se trató de amortiguar este hecho con la compañía de un docente, los niveles de utilidad resultaron menores. De todas formas es importante destacar que no es un objetivo principal para las auditorías el ser de utilidad a los grupos sino que más bien éste es un objetivo secundario que se cumple como resultado de la auditoría.

Escala Cuantitativa (CT)	Escala Cualitativa (CL)	Escala Si o No (SN)
1 - Nada	1 - Malo	Si
2 - Poco	2 - Regular	No
3 - Medio	3 - Aceptable	
4 - Bastante	4 - Bueno	
5 - Mucho	5 - Muy Bueno	

Tabla 19. Escalas utilizadas en el 2002.

Previo a concluir, debe decirse que en el 2002 se utilizó una escala del 1 al 5 para calificar las respuestas, la cual se usa como escala cuantitativa y cualitativa dependiendo del caso. En el caso en que lo que se pida sea una respuesta cualitativa la escala se corresponde con CL, mientras que en el caso en que se pide una respuesta cuantitativa la escala se corresponde con CT. También se utilizó una escala SN para algunas de las preguntas. En la Tabla 19 podemos ver las distintas escalas utilizadas en el 2002.

Conclusiones

Como conclusiones generales puede decirse que es muy difícil analizar datos que no fueron obtenidos con las mismas preguntas ni utilizando las mismas escalas. Además no se puede sacar conclusiones acertadas comparando dos experiencias aisladas donde la subjetividad en las respuestas y las características únicas de cada proyecto juegan un papel fundamental en los datos. Las diferencias existentes en algunos casos son tan mínimas que no permiten sacar una conclusión certera. Lo que sí se puede decir es que no se percibe que haya habido un retroceso en el modelo de proceso construido y la extensión OO frente al modelo Java existente, lo que es considerado un resultado importante. Se puede notar la falta de conocimiento y de comprensión de los estudiantes con respecto al modelo por lo que se sugiere mayor apoyo por parte de los docentes. El hecho de agregar las extensiones al modelo y la forma de presentación utilizada puede haber complicado en cierta forma su comprensión, lo cual se vería seguramente subsanado con una presentación con vistas como se comentó al explicar la creación del modelo.

Se pone una vez más en evidencia que el esfuerzo incurrido de los estudiantes es superior a las 15 horas previstas en cualquiera de los modelos utilizados. Se sugiere además una nueva puesta en práctica del modelo que permita realizar ajustes para la mejora del mismo y que se vuelvan a analizar los datos. Una nueva puesta en práctica del modelo permitiría además presentar a los estudiantes y explicarles el modelo en el curso de Introducción a la Ingeniería de Software, por lo que contarían con un mayor conocimiento del mismo.

5.4.3 Satisfacción del cliente

Con el objetivo de evaluar la satisfacción del cliente con el proceso seguido y con el producto obtenido como resultado del mismo, se elaboró también una encuesta para ser respondida por los clientes de cada quipo de trabajo. La encuesta debió ser realizada por cada equipo a su cliente como parte de la disciplina de Comunicación en la Fase de Transición. Se pretende con esta encuesta obtener una evaluación del producto obtenido e indirectamente obtener también una evaluación del proceso seguido. Tanto el cuestionario como las respuestas a los mismos brindadas por los clientes se encuentran disponibles en el CD del proyecto.

No Contesta	Muy Insatisfecho (1)	Insatisfecho (2)	Indiferente (3)	Satisfecho (4)	Muy Satisfecho (5)	Promedio de Satisfacción
¿Cómo se siente con respecto al trabajo del equipo durante la etapa de relevamiento de requerimientos?						
1	0	1	0	2	0	3,33
¿Cómo calificaría los métodos utilizados por el equipo de trabajo (Reuniones, documentos, Demos, etc.) para el relevamiento y la validación de los requerimientos con usted?						
0	0	0	1	3	0	3,75
¿Cuán satisfecho se considera con la comunicación con el equipo de trabajo?						
0	0	1	0	1	2	4
¿Cuán satisfecho se considera con el conocimiento técnico de los distintos integrantes del equipo de trabajo?						
0	0	0	0	4	0	4
¿Cómo calificaría la flexibilidad del equipo de trabajo frente a los cambios por usted solicitados?						
0	0	0	0	3	1	4,25

¿Cómo calificaría la organización y gestión del equipo de trabajo durante la última etapa del proyecto?						
0	0	1	0	2	1	3,75
¿Cuán satisfecho se considera con la funcionalidad del producto desarrollado?						
0	0	0	0	2	2	4,5
¿Cuán satisfecho se considera con la disponibilidad y confiabilidad del producto desarrollado?						
2	0	0	1	0	1	4
¿Cuán satisfecho se considera con la integración del producto desarrollado a su entorno de trabajo?						
0	0	0	0	3	1	4,25
¿Cuán satisfecho se considera con la efectividad y facilidad de uso de los mecanismos de ayuda del producto desarrollado?						
2	0	0	1	1	0	3,5
¿Cuán satisfecho se considera con lo amigable a la operación del usuario del producto desarrollado?						
0	0	0	0	4	0	4
¿Considerando todos los aspectos de su experiencia, cuán satisfecho se considera con el producto obtenido?						
1	0	0	0	2	1	3,25

Tabla 20. Resumen de respuestas al cuestionario de satisfacción del cliente

El estudio se realiza para los cuatro clientes de aquellos grupos que realizaron el cuestionario. A continuación, en la Tabla 20 se muestra un resumen de las respuestas obtenidas y un promedio de satisfacción.

Como se puede observar en la tabla, el promedio de satisfacción de los clientes que contestaron el cuestionario es superior a 3 en todos los casos. Si bien no se pueden sacar conclusiones generales, por no contar con todos los datos, se puede decir que el nivel de satisfacción de los clientes con los productos obtenidos y los métodos utilizados por los equipos de trabajo es alto. Puede decirse entonces que el modelo propuesto sirvió como guía a los estudiantes para lograr un producto de calidad cumpliendo con su objetivo principal.

5.5 Ajustes Realizados

Durante la puesta en práctica del modelo de proceso, y una vez finalizada la misma, surgieron algunos ajustes como sugerencias de los estudiantes y docentes del curso. Entre los ajustes realizados se puede destacar la modificación del contenido de algunas de las plantillas, sugerido por los docentes. El objetivo de tal modificación buscó obtener más información sobre el trabajo de los estudiantes y facilitar la realización del artefacto entregable.

Durante el transcurso del semestre se modificaron algunos links corruptos en el sitio web del modelo y se agregó información en el Plan de Configuración sobre la nomenclatura a utilizar para los artefactos entregables. Se sustituyeron las actividades Validar con Prototipo y Validar los Requerimientos por una sola actividad Validar con el Cliente que tiene una plantilla para el artefacto de salida. Esta decisión se tomó para facilitar la comprensión y realización de la actividad de validación ampliándose la descripción y sugiriendo en ella una agenda de artefactos a validar en cada fase del proyecto.

Se realizaron correcciones de inconsistencias existentes entre las responsabilidades de los roles Analista y Arquitecto. Además, se ampliaron las descripciones de algunos artefactos entregables atendiendo a las dudas de algunas estudiantes, es decir, se incorporaron a las descripciones de las actividades las aclaraciones que iban surgiendo como respuestas a las dudas de los estudiantes.

Se modificó también la plantilla del Modelo de Casos de Prueba a pedido de los docentes para adaptarla a lo enseñado en el curso de Ingeniería de Software. El objetivo es que los estudiantes se sintieran familiarizados con el contenido de la plantilla identificándolo con lo aprendido en el curso de Ingeniería de Software, que es previo y que sirve de marco teórico para el curso de Proyecto de Ingeniería de Software.

Se modificaron las plantillas del Informe Final de Proyecto y el Informe Final de Verificación atendiendo al pedido de los docentes de solicitar más información en los mismos. En las nuevas plantillas se solicitó más información sobre las mediciones realizadas. En la plantilla de Informe Final de Proyecto las modificaciones sugerían algunas mediciones de esfuerzo interesantes para realizar la evaluación final del proyecto. Como ejemplo de mediciones que se sugirieron pueden mencionarse la cantidad de horas trabajadas por semana del equipo, la cantidad de horas trabajadas por fase e iteración del equipo, las horas totales por integrante, entre otras. En el Informe final de Verificación se detalló específicamente la información que se requería solicitando datos como una gráfica y una tabla con datos de los casos de prueba y la cantidad de errores detectados por fase para cada versión, cantidad de verificadores para la versión y cantidad de horas invertidas. Para cada caso de uso se solicitó también información como la cantidad de casos de prueba, la cantidad de errores detectados, cantidad de verificadores y cantidad de horas invertidas.

Una vez concluido el semestre se eliminaron los artefactos entregables de Requerimientos Candidatos y Requerimientos Suplementarios fusionándose toda esta información en el documento de Especificación de Requerimientos y el Alcance del Sistema. Este cambio fue sugerido por los docentes y llevado adelante por algunos de los grupos, incluso en el transcurso del proyecto. El ajuste al proceso se realizó una vez concluida la prueba del mismo ya que se vio como una forma de aliviar el proceso en esa disciplina y a su vez contribuir a la facilidad de comprensión por parte de los estudiantes.

En el proceso de agregado de la extensión para desarrollo Integrado e Integrado Orientado a Objetos se decidió también pasar la actividad Planificar la Integración de la Iteración de la extensión OO al F, ya que se entendió que era una actividad que no dependía de la tecnología y que resultaba útil en cualquier contexto. Lo mismo ocurrió con la actividad de Definir Estándares de Documentación de Usuario incorporada al modelo de proceso Factor.

Los anteriores corresponden a los ajustes más importantes que se llevaron adelante durante la prueba del proceso y una vez finalizada la misma.

Capítulo 6

Conclusiones y Trabajos futuros

La industria y la Facultad de Ingeniería inmersa en ella no se encuentran ajenas al problema que se plantea con la calidad del software que se construye actualmente. En este proyecto se pudo construir un modelo de proceso único con extensiones para desarrollo específico Orientado a Objetos y con GeneXus. Se agregaron además una extensión para desarrollo Integrado de múltiples grupos de trabajo y otra extensión para desarrollo Integrado Orientado a Objetos.

El modelo de proceso construido resulta “pesado” a los estudiantes que lo utilizan ya que contiene muchas actividades y entregables a realizar. Se estudiaron metodologías ágiles para agilizar el proceso, pero, dado el contexto en que el proceso se aplica y la dimensión de los proyectos para los cuales es utilizado, el incorporar tales técnicas conlleva a la pérdida de algunas de sus propiedades, cómo consistencia de contenido y seguimiento docente entre otras, incumpliendo así con algunos de sus requerimientos.

El modelo de proceso creado y las extensiones OO y Gx fueron puestos en práctica con resultados satisfactorios tanto para los equipos de trabajo como para los clientes de los mismos. Según las encuestas realizadas y los datos recabados de las mismas, el modelo de proceso resultó útil como guía a los estudiantes que lo utilizaron en el curso Proyecto de Ingeniería de Software. Además, según los datos que se obtuvieron con respecto a la satisfacción del cliente, se pudo ver que los productos construidos cumplieron en gran medida las expectativas de los mismos. Igualmente, se realizaron ajustes basados en las sugerencias de los estudiantes y de los docentes que fueron vertidas en el newsgroup de la asignatura y en las auditorías.

Uno de los objetivos principales de este proyecto es lograr un modelo único unificando los modelos existentes y disminuyendo el esfuerzo de mantenimiento. Este objetivo se cumplió satisfactoriamente con la creación del modelo de proceso Factor. Luego, para proveer la posibilidad de extensión del modelo para su uso en contextos específicos se ideó un mecanismo que permite fácilmente crear extensiones del modelo Factor con solo modificar y/o agregar actividades, roles y entregables. Utilizando ese mecanismo se agregaron las extensiones OO, Gx, I e IOO. Por su parte, la creación de la extensión I para el desarrollo integrado independiente de la tecnología es un resultado de valor ya que permitirá crear nuevas extensiones para el desarrollo con otras tecnologías. Es importante notar, que el haber ganado en mantenibilidad del modelo de proceso tuvo una repercusión negativa en su usabilidad, ya que resulta difícil tener una visión cabal del modelo Factor y la extensión particular que se desea aplicar. Una herramienta de generación de los modelos a partir de un repositorio único podría dar solución además a éste problema.

Las auditorías realizadas durante la puesta en práctica del modelo de proceso cumplieron con su objetivo. Mientras que la opinión general de los grupos es que fueron de alguna utilidad, los directores de los proyectos afirman que resultaron útiles y que hubiera sido

bueno realizar más de una. Por otra parte, los directores estuvieron de acuerdo en que las conclusiones obtenidas en los informes realizados para cada grupo reflejaban la realidad de los equipos.

Con respecto a la experiencia con la combinación de roles Responsable de SQA- Responsable de Verificación se puede decir que se notaron algunos problemas reflejados sobre todo en el esfuerzo invertido. Del seguimiento de los grupos tanto en el newsgroup como en las auditorías, parece desprenderse que existe un problema en la comprensión de estos roles tan importantes. Resulta difícil a los estudiantes captar la esencia del rol de Responsable de SQA y confunden constantemente sus actividades con las del área de Verificación. El concepto de calidad es muy difícil de comprender y sobre todo de materializar, por lo que generalmente causa problemas a los estudiantes. Una posibilidad entonces es que la combinación de roles propuesta no sea mala por sí misma sino que necesite de apoyo por parte de los docentes del curso y de los directores de proyecto. El apoyo puede brindarse en clases extraordinarias como se realiza por ejemplo para el área de requerimientos. Otra solución posible, además del apoyo mencionado, es crear un nuevo rol de asistente de SQA que podría ser combinado con el Administrador para el aseguramiento de la calidad desde el punto de vista de la gestión del proyecto y por el Arquitecto para el aseguramiento de la calidad desde el punto de vista de la arquitectura del sistema.

Como trabajo futuro se propone la realización de la extensión para desarrollo Integrado GeneXus, que no se llevó adelante por falta de recursos y de tiempo. Además, se sugiere realizar una segunda experiencia de prueba del modelo para realizar nuevas mejoras y poner a prueba nuevamente la combinación de roles Responsable de Verificación – Responsable de SQA - Analista brindando más apoyo en la comprensión cabal de ambos roles.

Otra línea de trabajo comprende la realización de una nueva forma de presentación dinámica a partir de un repositorio o base de datos, que permita obtener vistas del modelo y cada una de sus extensiones.

En lo personal, la realización del proyecto aportó enormemente al aprendizaje en el área de Ingeniería de Software, que es de mi interés. El hecho de realizarlo sola me implicó un esfuerzo adicional, lo cual se compensó con la satisfacción de haber elaborado un modelo de proceso que al ponerlo en práctica arrojó resultados positivos.

Apéndice A

Contenido del CD

En este apéndice se describe el material elaborado durante el transcurso del Proyecto de Grado, incluido en el CD que acompaña al presente informe. Este material comprende al modelo de proceso Factor y sus extensiones, descritas en el Capítulo 4, teniendo la forma de un sitio web como se describe en la Sección 4.3.1. Asimismo, se incluyen las agendas de actividades y entregables para cada proceso, el material de las auditorías realizadas y los cuestionarios finales a estudiantes y clientes.

Se detalla a continuación todo el material disponible en el CD.

- Modelo de proceso Factor y Extensiones
Sitio web del modelo de proceso y sus extensiones.
- Agenda de Actividades y Entregables completa
Agenda completa de Actividades y Entregables para el modelo de proceso Factor y sus Extensiones.
- Herramienta para generación de agendas y páginas htm
Herramienta para generar las agendas particulares tanto del modelo de proceso como de cada extensión. Permite generar, además, páginas htm conteniendo la agenda de de entregables para cada iteración.
- Agenda de Actividades y Entregables para el Modelo Factor
- Agenda de Actividades y Entregables para la extensión OO
- Agenda de Actividades y Entregables para la extensión Gx
- Material de la primera Auditoría
Cuestionarios, resúmenes de respuestas de los grupos auditados, información de los grupos auditados, cronograma e informes de la auditoría de Fase Inicial.
- Material de la segunda Auditoría
Cuestionarios, resúmenes de respuestas de los grupos auditados, información de los grupos auditados, cronograma e informes de la auditoría de Fase de Elaboración.
- Agenda de Actividades y Entregables para la extensión I
- Agenda de Actividades y Entregables para la extensión IOO

- Cuestionarios finales a Estudiantes
Cuestionarios contestados por los estudiantes y resúmenes de respuestas a los cuestionarios.
- Cuestionarios finales a Clientes
Cuestionarios contestados por los clientes para evaluar la satisfacción de los mismos con el producto obtenido.
- Informe final – Factorización de dos modelos de proceso
Versión electrónica del presente documento.

Referencias Bibliográficas

- [1] Grupo de Ingeniería de Software, Instituto de Computación, Facultad de Ingeniería, Universidad de la República.
<http://www.fing.edu.uy/inco/grupos/gris/> Último acceso: Mayo de 2005.
- [2] Instituto de Computación, Facultad de Ingeniería, Universidad de la República.
<http://www.fing.edu.uy/inco> Último acceso: Mayo de 2005.
- [3] Andrea Delgado, Maria Freira, Beatriz Pérez, Jorge Triñanes y Diego Vallespir. *Curso de Proyecto de Ingeniería de Software*.
<http://www.fing.edu.uy/inco/cursos/ingsoft/pis/index.htm> Último acceso: Mayo de 2005.
- [4] Rational Corp. *Producto del Proceso Unificado de Rational: Documentación on-line basada en web del producto RUP, versión 2002.05.01.01*.
- [5] Kent Beck y Ward Cunningham. *Extremme Programming*.
<http://www.extremeprogramming.org/> Último acceso: Mayo de 2005.
- [6] Andrea Delgado y Beatriz Pérez. *Modelo de proceso Java*. Proyecto de grado InCo 2000. Versión existente al 2004.
<http://www.fing.edu.uy/inco/cursos/ingsoft/pis/procesos.htm> Último acceso: Mayo de 2005.
- [7] Diego Vallespir. *Proceso Integrado*. Trabajo final para la asignatura de postgrado “Gestión de Software” dictado 2003.
<http://www.fing.edu.uy/inco/cursos/ingsoft/pis/procesos.htm> Último acceso: Mayo de 2005.
- [8] Diego Vallespir. *Proceso Integrado*. Trabajo de investigación realizado para el Gris. 2003. Versión draft.
<http://www.fing.edu.uy/~dvallesp/publicaciones/PI.pdf> Último acceso: Mayo de 2005.

- [9] Doris Correa y Ximena Romano. *MoDSGX - Un Modelo de Proceso para Desarrollo con GeneXus*. Proyecto de grado InCo 2002. Versión existente al 2004.
<http://www.fing.edu.uy/inco/cursos/ingsoft/pis/procesos.htm> Último acceso: Mayo de 2005.
- [10] Beatriz Pérez. *GXP - Aplicando eXtreme Programming*. Trabajo de investigación para el Grupo de Ingeniería de Software.
<http://www.fing.edu.uy/inco/cursos/ingsoft/pis/procesos.htm> Último acceso: Mayo de 2005.
- [11] Ivar Jacobson, Grady Booch y James Rumbaugh. *The Unified Software Development Process*. Addison-Wesley, 1999.
- [12] DSDM Consortium. *Dynamic Systems Development Method*.
<http://www.dsdm.org/> Último acceso: Mayo de 2005.
- [13] Grady Booch, Robert C, Martin y James Newkirk. *Object Oriented Analysis and Design with Applications, 2d. ed.* Addison Wesley Longman, Inc. 1998.
- [14] John Smith. *Una comparación de RUP y XP*. Trabajo realizado para la empresa Rational®. TP167, mayo de 2001. Paper incluido en la documentación on-line de RUP.
- [15] Peter Merel. *XUP: XP + RUP + MSF*. Versión 1.1 beta. 2000.
<http://home.san.rr.com/merel/XUP.doc> Último acceso: Mayo de 2005.
- [16] Microsoft Corp. *Microsoft Solutions Framework*.
<http://www.microsoft.com/technet/itsolutions/msf/default.msp> Último acceso: Mayo de 2005.
- [17] Gabriel Claramount y Luis Remuñan. *Especificación y Descripción de la liberación*. Trabajo realizado en el 2003 para la asignatura “Taller de Gestión de Software”.
<http://www.fing.edu.uy/inco/cursos/ingsoft/tgs/material.htm> Último acceso: Mayo de 2005.
- [18] Doris Correa y Dieter Spangenberg. *Integración en los Proyectos de Ingeniería de Software*. Trabajo realizado en el 2003 para la asignatura “Taller de Gestión de Software”.
- [19] Cecilia Belletti y Daniel Dávila. *GxPoints: una propuesta de métricas para aplicaciones GeneXus*. Material de la charla del XIII Encuentro Internacional GeneXus. Abril de 2004.
<http://www.gxtechnical.com/main/hevprgprogram.aspx?8,60,581,19%3a16%2f06%2f2004%3a%3a0> Último acceso: Mayo de 2005.
- [20] Craig Larman. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process*. Prentice Hall, segunda edición, 2002.

- [21] Philippe Kruchten. *The Rational Unified Process: An Introduction*. Addison-Wesley, tercera edición, 2003.
- [22] Norman Walsh y Leonard Muellner. *DocBook: The Definitive Guide*. Publicado por O'Reilly & Associates, Inc.
Sitio oficial <http://www.docbook.org/>
Último acceso: Mayo de 2005.
- [23] Rolf Hennicker y Nora Koch. *Una metodología basada en UML para el diseño hypermedia*.
<http://www.pst.informatik.uni-muenchen.de/personen/kochn/Uml2000.pdf>
Último Acceso: Mayo de 2005.
- [24] Andrea Delgado. *Informe procesamiento cuestionarios finales estudiantes año 2002*. Trabajo realizado para el Gris.