



Para citar este artículo, le recomendamos el siguiente formato:

García Mireles, G. y Rodríguez Jacobo, J. (2001). Aplicación del modelado de procesos en un curso de ingeniería de software. *Revista Electrónica de Investigación Educativa*, 3 (2). Consultado el día de mes de año en: <http://redie.uabc.mx/vol3no2/contenido-mireles.html>

Revista Electrónica de Investigación Educativa

Vol. 3, No. 2, 2001

Aplicación del modelado de procesos en un curso de ingeniería de software

Process Modeling in a Software Engineering Course

Gabriel Alberto García Mireles
gagmireles@hotmail.com

Jaime Nunó 2680
Ampliación Hidalgo, 22880
Ensenada, Baja California, México

Josefina Rodríguez Jacobo
jacobo@cicese.mx
Departamento de Ciencias de la Computación
Centro de Investigación Científica y de Educación
Superior de Ensenada (CICESE)

Km. 107, Carretera Tijuana-Ensenada, 22830
Ensenada, Baja California, México

Resumen

La coordinación en un proyecto de desarrollo de software es un factor crítico para liberar un producto de calidad dentro de las restricciones de tiempo, funcionalidad y costo acordadas con el cliente. Una de las estrategias para abordar este problema consiste en utilizar técnicas de modelado de procesos para capturar, evaluar y rediseñar el proceso de desarrollo de software. La valoración de los proyectos realizados en el curso de ingeniería y metodología de la programación impartido en el CICESE, desde la perspectiva de procesos, facilita la especificación de fortalezas y debilidades

del proceso de desarrollo utilizado. Se presenta la evaluación de la parte práctica del curso, las acciones de mejoramiento llevadas a cabo y los resultados preliminares al utilizar el enfoque de procesos en la etapa de análisis de un proyecto de desarrollo de software.

Palabras clave: Modelado de procesos, enseñanza de la ingeniería de software, método de proyectos.

Abstract

Coordination in a software engineering project is a critical issue in order to deliver a successful software product while considering the constraints of time, functionality and budget. An approach to address this problem consists in the use of process modeling techniques to capture, evaluate, and redesign the software process. The appraisal in a process-based view of software projects made in the software engineering course at CICESE, helped to discover the weakness and strength of the process. This paper shows the evaluation of lab sessions, the steps to improve software development and preliminary results of process enactment in the software project analysis phase.

Keywords: Process modeling, software engineering teaching.

Introducción

El propósito de la ingeniería de software es generar y mantener sistemas de software dentro de las restricciones de tiempo, funcionalidad y costos acordados con el cliente. Las metas de esta disciplina tecnológica son mejorar la calidad de los productos desarrollados y aumentar la productividad de los ingenieros de software. El grado de formalidad y el tiempo asignado al proyecto de software varía de acuerdo al tamaño y complejidad del producto que será desarrollado.

Conforme aumenta la complejidad y el tamaño del proyecto, la coordinación se dificulta debido al incremento en la comunicación entre los ingenieros de software, administradores y clientes (Fairley, 1985; Kraut y Streeter, 1995). En el ámbito educativo, las investigaciones realizadas indican que los estudiantes graduados de programas de licenciatura entienden poco el significado de "programar en grande"; es decir, aplicar los principios de la ingeniería de software al desarrollo de un producto por un equipo de personas, durante el cual la coordinación efectiva entre los participantes permite lograr un sistema de software exitoso (Upchurch y Sims-Knight, 1998).

Se utilizan diversas estrategias en la enseñanza de la ingeniería de software. Algunas de ellas se basan sólo en la revisión bibliográfica, sin llevar a la práctica el conocimiento adquirido en un proyecto (Tomayko, 1987). Otras se basan en el trabajo en equipo, y su objetivo central es que el estudiante se ejercite en el desarrollo de un producto para un cliente real, tome decisiones de acuerdo a las opciones o recursos disponibles y se enfrente con los aspectos de comunicación y coordinación típicos del trabajo en grupo (Upchurch y Sims-Knight, 1997).

Un problema recurrente es que el éxito de los proyectos en estos cursos depende de la habilidad y experiencia del instructor para dirigir proyectos. Es probable que distintos

instructores logren resultados diferentes utilizando el mismo modelo; o que el mismo instructor, con otro grupo de estudiantes, obtenga resultados dispares. Además, en algunos de estos cursos sólo se presta atención a las características de una buena arquitectura e implantación, sin integrar los aspectos de aseguramiento de la calidad ni administración. Estos problemas sugieren que muchos de los proyectos de ingeniería del software sufren de deficiencias en el proceso de desarrollo de software utilizado por el instructor (Collofello, Kantipundi y Kanko 1994).

El proceso de desarrollo de software se puede definir como el conjunto de actividades, métodos, prácticas y transformaciones que los individuos emplean para desarrollar y mantener el software, así como los productos asociados (Paulk, M. C., Weber, C. V., Curtis, B. y Chrisis, M. B., 1995). Un proceso definido y efectivo disminuye el esfuerzo en el desarrollo de un producto de software, y aumenta la productividad del grupo de desarrollo (Clark, 2000). De hecho, el modelado y la ejecución del proceso de desarrollo de software son una áreas principales de la investigación en ingeniería de software (Maurer y Kaiser, 1998), y su propósito es proponer soluciones a los problemas en el contexto organizacional con base en la explotación de las tecnologías de coordinación e integración (Warboys, Kawalek, Robertson y Greenwood, 1999).

En la búsqueda de la calidad en la enseñanza de la ingeniería de software basada en procesos, se han utilizado diversos enfoques. Por ejemplo, en ciertos cursos, se entrega a los estudiantes el documento de especificación de requerimientos. El propósito es que realicen el trabajo que se señala, cubriendo todas las etapas del desarrollo del producto, a partir del seguimiento de algunos procesos descritos de manera textual (Upchurch y Sims-Knight, 1998). Otros investigadores han modelado el proceso de desarrollo de software utilizando técnicas orientadas a objetos, con el propósito de comprender la complejidad del proceso paso por paso (Oktaba e Iburgüengoitia, 1998). También, se utilizan estándares y prácticas aceptados por la comunidad como base para introducir el proceso de desarrollo de software (Robillard, 1998; Jaccheri y Lago, 1997; y Mayr, 1997). Además, algunos expertos recomiendan la integración de procesos y el trabajo en grupo en el diseño de planes de estudios relacionados con la ingeniería de software (Bagert, Hilburno, Hislop, Lutz, McCracken y Mengel, 1999).

En el Centro de Investigación Científica y de Superior de Ensenada (CICESE), se imparte la asignatura obligatoria de Ingeniería y metodología de la programación, dentro del programa de maestría en Ciencias de la Computación. Las características del curso permiten tomarlo como referencia para realizar un proyecto de mejoramiento de procesos. En este artículo, se describe la aplicación de las técnicas de modelado de procesos en la parte práctica del curso de ingeniería de software para identificar las fortalezas y debilidades del proceso de desarrollo utilizado, y las acciones que se siguieron para mitigar los problemas detectados. En la segunda sección se señala la metodología que sirvió como base para hacer visible el proceso, evaluarlo y proponer mejoras. En la tercera sección se describen los pasos seguidos en el estudio del proceso de desarrollo, al tomar como referencia las etapas de un proyecto de mejoramiento de procesos y adaptarlas a las necesidades específicas del caso de estudio. La cuarta sección reporta las características generales de la sesión de laboratorio del curso, en donde los estudiantes desarrollan un sistema de software y resume las deficiencias detectadas en la coordinación del trabajo

entre los participantes del curso. En la quinta sección se indican las acciones que se tomaron para resolver los problemas detectados. La sexta sección expone los resultados obtenidos al poner en marcha los procesos mejorados. Finalmente, se describen las líneas que se seguirán en el curso para el mejoramiento de procesos y las conclusiones del trabajo.

1. Modelado del proceso de desarrollo de software

La metodología utilizada para mejorar el proceso que se sigue en el curso de ingeniería de software contempla las etapas de definición, captura, evaluación, rediseño y ejecución (Wastell, White y Kawalek 1994; Caputo, 1998; Arthur, 1992; Sommerville, 1995; Warboys *et al.*, 1999). La fase de definición establece los objetivos de un proceso, delimita las fronteras, señala las entradas y salidas principales, indica los clientes que se benefician; y los proveedores de entradas. La fase de representación y captura modela el proceso de manera detallada, partiendo de la información obtenida de entrevistas, revisión de documentos y planes para generar una representación gráfica del proceso. La etapa de evaluación analiza y evalúa el proceso con el propósito general de buscar debilidades y problemas referentes a la ineffectividad o ineficiencia para lograr las metas del proceso establecidas. Los resultados obtenidos de la evaluación facilitan el rediseño del proceso, en el que se utiliza un lenguaje de modelado comprensible para los usuarios del proceso. El proceso rediseñado se pone en marcha en la organización para comprobar que realmente satisface las metas establecidas.

En el modelado de procesos se contemplan cuatro aspectos: funcional, desempeño, organizacional e informativo (Curtis, Kellner y Over, 1992). En el aspecto funcional se consideran las actividades del proceso que están siendo ejecutadas y los flujos de entidades (documentos) más relevantes. En el aspecto de comportamiento o desempeño se presta atención al tiempo en que se realizan las actividades, así como al modo en que se efectúan (condiciones, secuencia e iteraciones). La vista organizacional del proceso se enfoca en el lugar físico, dentro de la organización donde se realizan las actividades y en la persona que tiene la responsabilidad de efectuarlas. Por último, el aspecto informativo aborda el aporte de los documentos en la coordinación y comunicación entre las funciones.

La descripción de un proceso puede tomar muchas formas y es posible usar un lenguaje gráfico para exponerlo. Los modelos gráficos ayudan a representar el proceso en estudio al disminuir la complejidad, propiciar un entendimiento común entre los participantes y permitir el estudio de alternativas (Miers, 1996); también, se pueden utilizar para influenciar, controlar y dirigir lo que acontece en el mundo real (Warboys *et al.* 1999). En el contexto organizacional, permiten la captura del comportamiento del proceso para su análisis posterior y también actúan como depósitos del conocimiento de la organización con el fin de facilitar el aprendizaje acerca de la organización y sus procesos (Ould, 1995).

La técnica para elaborar los diagramas del proceso de desarrollo de software en estudio fue la *Role Activity Diagram* (RAD). Esta técnica describe al proceso desde el punto de vista de roles (Ould, 1995). Quienes ocupan los roles ejecutan actividades y toman decisiones de acuerdo con las reglas de la organización; pueden realizar las actividades en paralelo y tener interacciones con otras funciones mientras progresa el trabajo con el fin de alcanzar

las metas del proceso. Las acciones pueden incluir el uso o producción de información o documentos. En la mayoría de los casos, el proceso de modelado conduce a la identificación inmediata de opciones de rediseño (Miers, 1996).

El éxito del proyecto en el curso depende de la colaboración que se logre entre los estudiantes, en tanto que la coordinación de los esfuerzos que realiza cada uno de los participantes de acuerdo con su puesto, es esencial para cumplir las metas del curso. La investigación de las interacciones, actividades, puestos y documentos que se producen mientras se trabaja en el proyecto permite mejorar el proceso de desarrollo de software.

2. Método de trabajo

El objetivo de esta investigación es sistematizar el proceso de desarrollo utilizado hasta este momento en el curso de Ingeniería y metodología de la programación, identificar las prácticas útiles y determinar las debilidades del proceso con el propósito de mejorarlo. Aunque no existe un enfoque único para resolver los problemas de la ingeniería de software, el mejoramiento de procesos se presenta como una alternativa para aumentar la productividad de los ingenieros de software y generar productos de mayor calidad (Hersleb, Zubrow, Goldenson, Hayes y Paulk, 1997; Clark, 2000). En el mejoramiento del proceso que se sigue en el curso, se consideran las metas de la ingeniería de software (Fairley, 1985; Pressman, 1993) para liberar un producto con la funcionalidad acordada con el cliente, la calidad requerida y dentro del tiempo estipulado para el proyecto. Así, un proyecto de desarrollo de software exitoso en el curso es aquel que cumple las metas descritas. Para este estudio, se toma como referencia las etapas de un proyecto de mejoramiento de procesos. En la Tabla I se resume la aplicación de este marco de referencia al ambiente particular del curso de ingeniería de software.

Tabla I. Etapas en el estudio del proceso de desarrollo de software y el mejoramiento de las actividades de la fase de análisis del proyecto

Etapa	Propósito
1. Definición	Determinar el estado actual del proceso de desarrollo de software desde la perspectiva de cada participante y proponer mejoras a las tareas relacionadas con el análisis de sistemas.
2. Captura	a. Analizar los documentos de proyectos anteriores para conocer las aportaciones de cada participante al proceso de desarrollo de software. b. Realizar entrevistas semiestructuradas aplicadas a estudiantes que trabajaron en los proyectos (descritos en el inciso anterior) con el fin de complementar y verificar las actividades realizadas, dependencias con otros participantes, problemas detectados y sugerencias de mejora. c. Revisar notas de la teoría del curso para determinar el alcance de los temas relacionados con el análisis de sistemas.
3. Representación	Facilitar la comunicación y comprensión del proceso a través de

	modelos gráficos del proceso general de desarrollo y de actividades de cada agente. La información de la etapa anterior se considera para obtener los modelos.
4. Evaluación	Determinar las debilidades de la fase de análisis de sistemas del proceso de desarrollo al tomar como referencia el modelo actual; compararlo con los reportes de la etapa de captura (resultados de las entrevistas y evaluación de las notas del curso) y confrontar esta información con bibliografía actualizada sobre el tema.
5. Rediseño	Crear una propuesta de proceso de la fase de análisis de sistemas, considerando los resultados del apartado anterior (evaluación).
6. Ejecución	Utilizar los modelos generados en un nuevo proyecto. Esta actividad requiere capacitar a los estudiantes para que usen los modelos.
7. Valoración	Determinar el impacto del uso de modelos y estándares en el curso de ingeniería de software.

En la etapa de definición, se aborda el proceso de desarrollo de software como una unidad integrada de las actividades de aseguramiento de calidad, administración e ingeniería del producto. La investigación del proceso de desarrollo comienza a partir de que cada estudiante tiene asignado un cargo en el proyecto; y, termina, cuando se entrega un producto funcional al cliente. En la primera etapa del estudio, se determina el estado actual del proceso y se generan modelos de las actividades e interacciones de cada uno de los agentes. En la segunda etapa, se evalúa, proponen mejoras y se ejerce el proceso de desarrollo tomando como referencia los modelos generados en el rediseño de la fase de análisis de sistemas.

La etapa de captura y representación tiene como objetivo conocer el estado actual del proceso, para lo cual, se usan distintas fuentes de información. En el caso particular de esta investigación, se consideraron los documentos generados durante los proyectos de cursos anteriores, entrevistas a participantes en dichos proyectos y revisión de las notas del curso:

a. Documentación de proyectos realizados con anterioridad. Desde 1994 se ha impartido la asignatura metodología e ingeniería de la programación en el CICESE. Los proyectos elegidos fueron DASIS (enero a abril) y GENSIS98 (septiembre a diciembre). Ambos fueron realizados en 1998 por los estudiantes del primer año de la maestría. La decisión para tomar estos proyectos como base para modelar el proceso de desarrollo se debe a que la información generada se encuentra en un depósito de información electrónico accesible; los documentos están organizados por agentes y son los que están mejor documentados, ya que, a diferencia de los proyectos anteriores, en éstos se encuentra información del perfil de cada agente en el proyecto de software y los documentos generados durante el ciclo de vida del proyecto. Otro factor que se consideró fue la semejanza entre los proyectos que se realizaron, ya que la plataforma de desarrollo, métodos de análisis y diseño, y tipos de revisiones técnicas fueron los mismos. Además, el primer autor participó como estudiante en el proyecto DASIS; y en el segundo, como auxiliar en el proceso de desarrollo. La información de los proyectos corresponde a: 1) planeación estratégica (misión, visión, valores, reglas), ejercicio que se lleva a cabo al

principio del curso; 2) descripción del perfil de cada agente (metas, actividades, relación con otros participantes, herramientas de apoyo y bibliografía); y 3) documentos asociados con el proyecto (requerimientos, especificación del software, diseño, módulos de código, manual de usuario, planes de trabajo de cada una de las funciones, minutas de reuniones técnicas, reportes de avance de actividades, entre otros).

b. Entrevistas semiestructuradas. Se preparó un guión para realizar las entrevistas a los estudiantes que ocuparon un puesto en los proyectos DASIS o GENSIS98. Se consideró el análisis del perfil de cada agente (inciso *a*), con información basada en la bibliografía de esta disciplina tecnológica y no en la experiencia práctica (esta investigación documental es realizada por los estudiantes al principio del curso). De los 28 estudiantes (14 en cada proyecto), se tomó una muestra de 10 estudiantes, cada uno de ellos tenía un puesto distinto. De esta manera, se contó con observaciones para cada uno de los 10 puestos disponibles en el curso. Las entrevistas se realizaron del 23 al 29 de julio de 1999. La duración de cada una fue aproximadamente de una hora. En ellas se investigó las actividades que en realidad fueron ejecutadas, su secuencia, los documentos generados, las herramientas utilizadas y las interacciones con otros puestos, con el fin de verificar y completar la información descrita en el perfil de cada agente. En estas entrevistas, también se preguntó por los problemas que enfrentaron al desempeñar sus actividades, así como sus sugerencias para mejorar la parte de laboratorio del curso.

c. Documentación teórica del curso. En este rubro, se analizó el material utilizado en la teoría del curso, las referencias bibliográficas y la secuencia de los temas, con el propósito de verificar que, en la parte de teoría, se traten los tópicos necesarios para llevar a cabo las actividades del laboratorio del curso. El tema que se revisó con un grado de detalle mayor fue el de análisis de sistemas, ya que éste es la base para el rediseño de procesos en esta investigación.

La captura y representación del proceso parte del análisis de las fuentes descritas en los párrafos previos y permite la generación de varios modelos para comprender el proceso de desarrollo vigente. Los diagramas de apoyo utilizados para enfocarse en las distintas perspectivas del proceso fueron: la imagen rica, cuyo propósito es tener una imagen detallada de la problemática (Warboys *et al.*, 1999; Wastell *et al.*, 1996); diagrama general del proceso de desarrollo de software, en el que se agrupan las actividades en las áreas de administración, control y desarrollo (Donaldson y Seigel, 1997); el diagrama de transición de estados, para identificar el comportamiento dinámico del proceso; y la matriz rol-actividad-documento con el fin de identificar las responsabilidades de cada puesto, las relaciones entre ellos y los documentos que se generan o utilizan en cada una de las actividades.

Se representaron en diagramas RAD la información depurada de las actividades efectuadas por cada uno de los agentes, la interacción que tuvo lugar entre ellos y los documentos o productos que se generaron durante el ciclo de vida del proyecto. El enfoque para representar el proceso tomó como referencia las actividades que ejecuta cada agente o puesto (Ould, 1995; Kawalek, 1994).

La evaluación y el rediseño del proceso se enfocaron a las actividades que se realizan en la etapa de análisis del ciclo de vida de un proyecto de software. Las razones para tratar con esta etapa se deben a que, por ser la primera fase del proyecto, los estudiantes aún no están familiarizados con el proceso que se utiliza. Además, un estudio realizado indica que se dedica mucho tiempo a esta fase del proyecto (34% en DASIS y 46% en GENSIS98) y se descuidan las actividades de codificación y pruebas de sistema (García Mireles y Rodríguez Jacobo, 2000a).

Para evaluar el proceso de análisis, se tomó como referencia la bibliografía básica de ingeniería de software (Sommerville, 1995; Pressman, 1993; Fairley, 1985) y literatura especializada (Dorfman y Thayer, 1990; Ayer, 1992). Fue de especial interés la revisión de temas relacionados con la preparación de cuestionarios para las entrevistas, identificación y clasificación de requerimientos, contenido y formato de los documentos de requerimientos del cliente y especificación de software. Además, se analizaron los temas que corresponden a la administración de éstos, criterios de calidad aplicados en esta etapa del ciclo de vida del proyecto, y la relación de dichas necesidades con la planeación del proyecto y administración de configuración.

El rediseño del proceso fue resultado de la etapa anterior, y en él se presenta la información en diagramas RAD. En esta fase del proyecto de mejoramiento del proceso, se toma como referencia la etapa de análisis del ciclo de vida tradicional (“cascada”) de un proyecto de software y se divide en tres partes, de acuerdo con los hitos de esta fase del proyecto y según las etapas de la administración de requerimientos: elaboración del documento de requerimientos, validación de los éstos por el cliente y elaboración de la especificación del software.

Los nuevos modelos generados para la etapa de análisis del sistema fueron presentados a dos estudiantes involucrados con estas actividades, quienes ocuparon los cargos de analista e ingeniero de control de calidad en los proyectos evaluados. Las pláticas informales que se llevaron a cabo con ellos (se realizaron a finales del mes de agosto de 1999 con una duración aproximada de 30 minutos) y tuvieron como fin validar la claridad y el rediseño completo del proceso. Se les explicó el propósito del modelo y el significado de cada elemento gráfico. De acuerdo con su percepción del proceso de desarrollo, indicaron que el modelo es adecuado.

El propósito de generar modelos es utilizarlos para ejecutar el proceso. Las mejoras al proceso de análisis se aplicaron en el periodo que comprende de septiembre a diciembre de 1999, en el curso ingeniería y metodología de la programación donde participaron 15 estudiantes de nuevo ingreso al programa de maestría. Se presentó una clase en referencia al proceso de desarrollo de software y se entregaron los modelos a todos los estudiantes para que se familiarizaran con las actividades relacionadas con el análisis de sistemas. Después de la clase, se entrevistó informalmente a tres de los estudiantes, directamente involucrados con las actividades de esta etapa (ingeniero de requerimientos, arquitecto del sistema e ingeniero de control de calidad). Al final del curso se preparó un cuestionario de opinión y se entrevistó a los participantes responsables del análisis en ese curso. El propósito es conocer el impacto del enfoque de procesos en el curso de ingeniería y metodología de la programación.

3. Descripción del curso

El objetivo del curso Ingeniería y metodología de la programación es “entender el desarrollo de proyectos de software de mediana escala (14 a 20 personas), evaluar los procedimientos, conjuntar herramientas, definir procesos y acumular la memoria organizacional” (Licea, Rodríguez y Favela, 1996), con el propósito de mejorar el proceso de desarrollo de software. La estrategia que se utiliza es una adaptación del modelo presentado por Tomayko (1987), en el que se describe el desarrollo de un proyecto de software para un cliente real, y cada estudiante de la clase desempeña un papel o puesto en el proceso de desarrollo del software.

El curso contempla una parte teórica y otra práctica. El propósito de la parte teórica es mostrar las actividades involucradas en el desarrollo de un sistema de software y las relaciones entre la ingeniería del producto, la seguridad de la calidad y la administración del proyecto. Además, se analizan los métodos y técnicas que existen en la actualidad en cada una de las disciplinas de la ingeniería del software.

En la parte práctica, los estudiantes forman una "empresa" con el fin de desarrollar un sistema de software. El instructor se desempeña como consultor. Los estudiantes, junto con el instructor, determinan el nombre de la empresa y el logotipo que los identifica. Trabajan en la definición de la misión, visión, valores y reglas del grupo que deberán observarse durante la vigencia del curso. Cada estudiante ocupa un puesto en la empresa; éste se determina a través de una entrevista de trabajo con el instructor, en la cual se evalúan las habilidades y conocimientos de cada participante. Los puestos disponibles son: administrador del proyecto, ingeniero de control de calidad, ingeniero de validación y verificación, especialista en documentación, administrador de configuración, analista, diseñador, programador, ingeniero de pruebas e ingeniero de mantenimiento.

En la parte práctica, los estudiantes llevan a cabo reuniones semanales de revisiones técnicas regidas por una agenda. En estas reuniones se evalúa el trabajo realizado y se toman decisiones respecto a la calidad de los documentos presentados, problemas a los que se han enfrentado en el desarrollo del sistema y la repercusión de sus decisiones en el calendario de tareas del proyecto. Durante las revisiones técnicas, algunos estudiantes fungen como revisores y el resto como observadores del proceso. Al final de la revisión, el instructor aporta sugerencias a los estudiantes sobre las actividades que deberían realizar para corregir los problemas detectados.

El proyecto de software asociado al curso tiene un cliente real, quien validará el sistema cuando sea liberado. Sin embargo, el proyecto debe ajustarse a ciertas restricciones implícitas en los objetivos y en el entorno del curso de ingeniería del software (García y Rodríguez, 2000b):

- La fecha de entrega del sistema por desarrollar se rige por el calendario escolar (aproximadamente 3 meses).

- Los estudiantes cursan otras asignaturas (tres, en promedio) mientras trabajan en el proyecto.
- Algunos estudiantes no han cursado con anterioridad materias relacionadas con esta disciplina tecnológica.
- La metodología de análisis y diseño orientada a objetos que se utiliza es *Unified Modelling Language* (UML).
- La plataforma de desarrollo para el sistema de software es el *World Wide Web* (WWW), por lo que es necesario familiarizarse con los lenguajes de programación enfocados a esta plataforma.
- Se requiere que los participantes generen un depósito de información centralizado en la plataforma WWW.
- Además, se proporcionan las facilidades para que utilicen el correo electrónico como apoyo a la comunicación.

Las metas principales de este curso implican que los estudiantes aprendan a trabajar en equipo, comprendan la relación que existe entre la ingeniería del producto, las actividades de garantía de calidad y la administración; que conozcan las metodologías, técnicas y herramientas más recientes en el desarrollo de sistemas de software y las apliquen en la resolución de un problema real. Un punto muy importante es que todo el grupo participa en el desarrollo del proyecto y nadie, por sí sólo, puede llevar a término el proyecto de software. Este aspecto del curso se ve reflejado en el peso de la evaluación, ya que 50% de la calificación se obtiene del sistema de software terminado y validado con la funcionalidad requerida por el cliente.

Los modelos generados en la etapa de captura y representación del proceso facilitan la comprensión de las tareas que se llevan a cabo en los proyectos. Las actividades realizadas se separaron según las fases típicas del ciclo de vida tradicional de un proyecto de software: análisis, diseño, codificación y pruebas. A su vez, los distintos puestos que pueden tomar los estudiantes se agrupan en las áreas de administración, control y desarrollo (Donaldson y Seigel, 1997).

En el área de administración, el propósito es planear, organizar, dirigir y controlar las tareas del grupo de desarrollo para entregar un producto bajo las restricciones de tiempo, costo y funcionalidad requerida por el cliente. En nuestro estudio, este grupo de actividades fueron desempeñadas por el puesto administrador del proyecto.

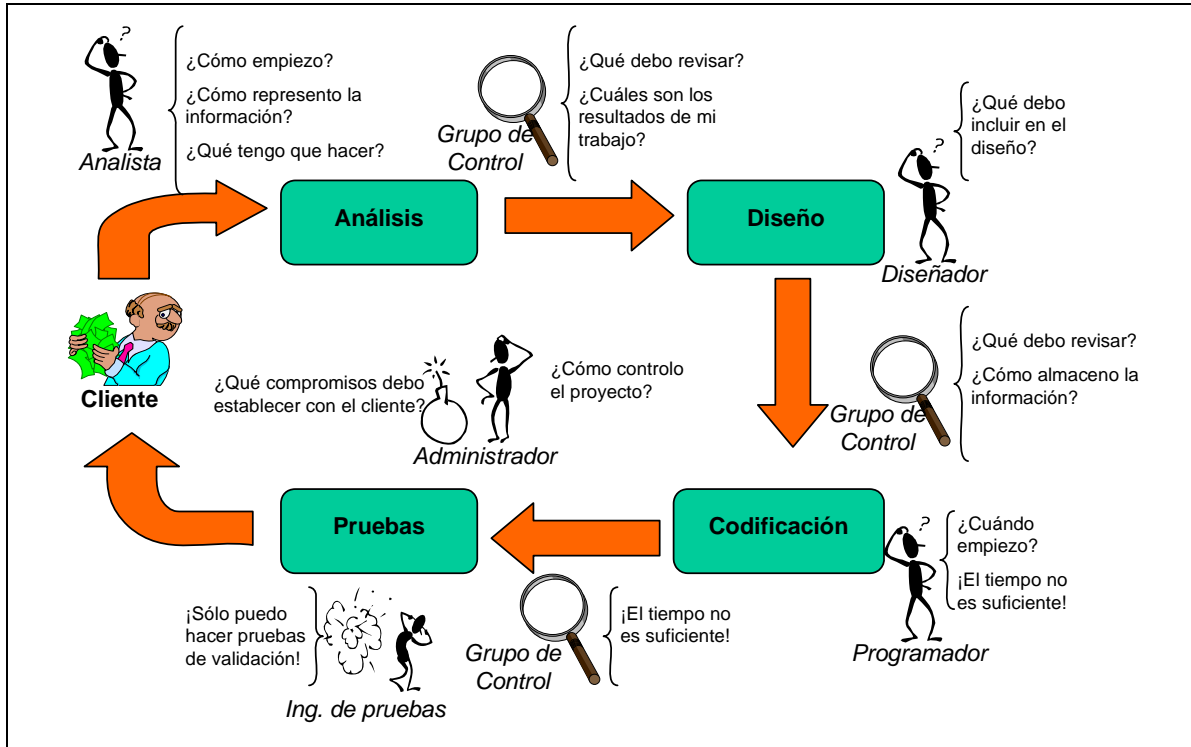


Figura 1. Imagen rica del proceso general de desarrollo de software

El grupo de control garantiza que se logre y mantenga la integridad del producto; confirma que el desarrollo de software ha seguido un proceso disciplinado y satisface las necesidades del cliente, proporcionando visibilidad al proyecto. Este grupo está compuesto por el ingeniero de control de calidad, el ingeniero de validación y verificación, el ingeniero de mantenimiento, el administrador de configuraciones y el especialista en documentación.

El grupo de desarrollo se enfoca en las actividades de ingeniería del producto, derivadas de las etapas del ciclo de vida del proyecto: análisis, diseño, codificación y pruebas. Los puestos que pertenecen a este grupo son: el analista, el diseñador, el programador y el ingeniero de pruebas. Este grupo ejecuta las actividades clave del proceso de desarrollo de software.

A continuación, se describe brevemente el flujo de la tareas ejecutadas, la interacción entre los distintos grupos y los problemas detectados en las etapas del proyecto de desarrollo de software, según los resultados de la evaluación de los cursos.

Las actividades del grupo de control están inmersas en el proceso de desarrollo de software; su notoriedad se logra al final de cada etapa, al evaluar y aprobar los productos resultantes de cada una de las etapas del ciclo de vida del proyecto. La carga mayor de trabajo de administración está al inicio del proyecto, ya que es ahí donde se debe establecer el plan de administración y el contrato con el cliente. Durante el desarrollo del sistema, el trabajo del administrador consiste en controlar el progreso del proyecto.

Cada una de las etapas del proyecto presenta una serie de retos a los participantes (véase Figura 1). Al principio del curso, los estudiantes deben investigar qué tipo de actividades corresponden a su puesto, cómo ejecutarlas, definir los formularios y documentos que se utilizarán en el desarrollo del proyecto, y elegir las herramientas de apoyo y metodologías. Los participantes que están bajo presión en esta etapa son el administrador del proyecto y el analista, ya que son los primeros en entrar en acción sin comprender claramente sus responsabilidades (debido a las limitaciones de tiempo en el curso). De acuerdo a los resultados de las entrevistas, los participantes indican que sería deseable tener la información de apoyo necesaria para cumplir con sus responsabilidades, tales como el estándar para el documento de requerimientos, aspectos por verificar en la presentación de requerimientos y la estructura del plan de administración del proyecto.

Muchos de los estudiantes no han participado con anterioridad en un proceso de revisión técnica, en el cual se prepara una agenda, se definen los temas por discutir y el tiempo asignado a cada uno de ellos. La organización y comunicación entre los participantes son escasas durante las primeras sesiones. Además, la revisión de un producto de software, que es uno de los objetivos de la reunión, también tiene sus debilidades en las primeras sesiones. Aunque los estudiantes en teoría conocen los atributos de calidad que deben revisar, se les dificulta aplicarlos al momento de evaluar los documentos, ya que el problema principal al que se enfrentan es entender claramente el problema que resolverán (documento de requerimientos), descrito por otra persona (el analista).

Durante el segundo mes del curso, los estudiantes logran controlar la logística de la revisión técnica, pero los agentes enfocados al área de control aún tienen dificultades para determinar la calidad del producto debido a que no existen estándares que sirvan de base para ese trabajo.

En las últimas tres semanas del curso, nuevamente el grupo pierde el control del proyecto. Esto se debe a que están por terminar las otras asignaturas y necesitan dedicar más tiempo para terminar sus trabajos finales. Además, los retrasos en la liberación de los productos generados en la etapa de análisis provocan que se recorra el calendario de actividades programadas, con lo que se reduce el tiempo para las actividades de codificación y pruebas. En la información disponible en los depósitos de los proyectos, no se encuentra referencia a informes de calidad realizados durante este periodo, ni siquiera se controla la configuración del código generado.

A pesar de estos contratiempos, se logra tener un prototipo del sistema de software al final del trimestre, el cual se presenta al cliente para que lo valide. Después de esta sesión, el administrador del proyecto entrega la documentación generada y el código fuente del prototipo. Finalmente, se realiza una evaluación del legado del proyecto, desde la perspectiva de cada puesto.

4. Cambios realizados en el curso

La evaluación de los modelos del proceso de software obtenidos y la comparación de las funciones realizadas por cada puesto, según lo establecido en la bibliografía de esta disciplina tecnológica, permiten distinguir las áreas problemáticas del proceso que se sigue en el CICESE. Además, las entrevistas realizadas indican que la falta de información confiable, la ausencia de estándares, la falta de recursos de hardware y software, el exceso de revisiones formales y la falta de referencias bibliográficas actualizadas son factores que producen un impacto negativo para liberar el sistema de software en el tiempo estipulado.

Aunque el modelo del proceso de desarrollo de software se realizó con la información de la parte práctica del curso de ingeniería del software, las deficiencias detectadas también afectan la parte teórica. La actualización de la teoría incluyó cambiar el orden en que se imparten los temas, al tomar como referencia las etapas en que se lleva a cabo el proyecto de software e introducir el tema de ingeniería de procesos con, el propósito de preparar a los estudiantes para que comprendan los modelos que se les presentarán en las sesiones de laboratorio del curso.

Los resultados de las entrevistas y la evaluación de los modelos generados mostraron poca visibilidad en la fase de análisis del proyecto. Por tal motivo, se profundizó en la teoría de análisis de sistemas, y se indicaron los aspectos clave en la determinación de los requerimientos y la especificación, así como la administración de éstos durante el ciclo de vida del proyecto. Otro aspecto abordado en la teoría se refiere a los atributos de calidad que se revisan en los documentos producidos en la etapa de análisis y los tipos de revisiones técnicas que se pueden aplicar.

Finalmente, se proporcionó bibliografía actualizada y clasificada de acuerdo con el tema de estudio, con el fin de presentar y analizar los métodos y técnicas utilizados en las distintas disciplinas de la ingeniería del software. Para cada uno de los temas, se indicaron de dos a cinco referencias clave. Además, se incluye una lista de bibliografía de apoyo con el fin de utilizarse como una guía para que el estudiante conozca, con mayor profundidad, las actividades que corresponden al puesto asignado.

En la parte práctica del curso, la estrategia de mejora implicó rediseñar los procesos que corresponden a la etapa de análisis, en el ciclo de vida del proyecto, debido al impacto que tiene esta fase en el desarrollo del sistema de software. Con el propósito de verificar la precisión de los modelos generados, fueron validados por participantes en los cursos anteriores. En la nueva propuesta, las responsabilidades del analista fueron asignadas a los puestos de ingeniero de requerimientos y de arquitecto del sistema. La meta del primero es elaborar el documento de requerimientos, etapa en la que es indispensable la comunicación estrecha con el cliente y sus representantes, para identificar las necesidades que el sistema de software debe satisfacer; y la meta del segundo es proponer una solución a las necesidades del cliente, que sea representada en un lenguaje de análisis de sistemas (UML).

Las tareas que se realizaron durante la etapa de análisis del proyecto y las interacciones que se llevan a cabo entre los participantes permitieron generar los modelos actualizados para los procesos de elaboración del documento de requerimientos, validación de requerimientos y construcción de la especificación del sistema de software (véase Figura 2). Con la nueva definición de procesos, los estudiantes pueden entender mejor qué deben hacer, qué pueden

esperar de sus compañeros y qué deben proveer. Los procesos están divididos en roles. Un rol involucra un conjunto de acciones que se llevan a cabo por un individuo o un grupo dentro de la organización. Además, un rol incluye la lógica que controla las acciones de acuerdo con las reglas de la organización. También, un rol tiene los recursos necesarios para lograr sus actividades (Ould, 1995).

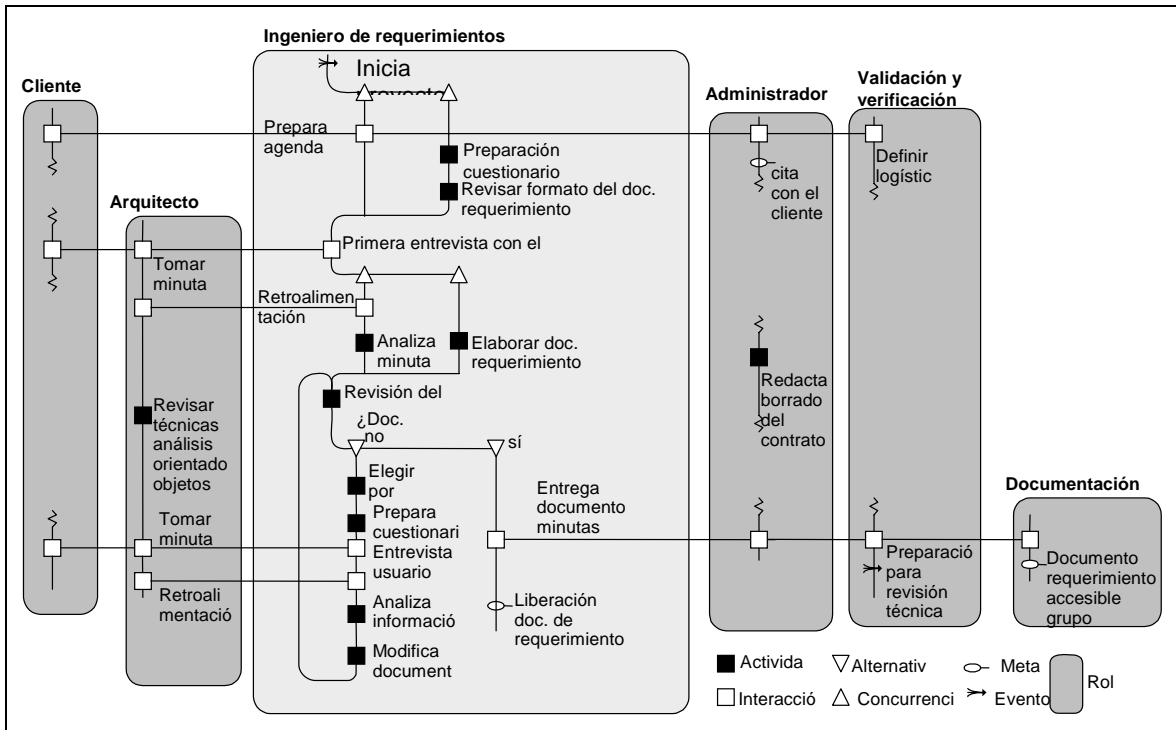


Figura 2. Diagrama RAD para la elaboración del documento de requerimientos

Los productos que se generan (documento de requerimientos del cliente y especificación del sistema de software) tienen como función principal facilitar la comunicación entre el cliente y el grupo de desarrollo, guiar la creación y modificación de los productos de software, dirigir la planificación del proyecto y garantizar la calidad del sistema de software construido (Rombach, 1990). La creación de estos documentos brinda a los estudiantes la oportunidad de modelar un problema real de un cliente, y de enfrentarse a las cuestiones de la descripción del producto de software. Al hacerlo, se introduce a los participantes en los problemas asociados a las descripciones en lenguaje natural y la representación de las características del producto de acuerdo con la metodología de análisis de sistemas empleada (UML, en el caso de estos cursos). Durante la elaboración de estos documentos, se reconcilian conflictos e inconsistencias producidos en la comunicación entre el cliente y el grupo de desarrollo, así como los que se presentan entre los mismos miembros del grupo de desarrollo de software. Además, obliga a los estudiantes a trabajar en sus habilidades de lectura y comprensión (Upchurch y Sims-Knight, 1998).

Otro aspecto importante en el rediseño de procesos se refiere a las entidades (documentos) que sirven de entrada o salida para las actividades señaladas en los modelos (Curtis *et al.*, 1992). La evaluación de los documentos utilizados en esta fase del proyecto permitió la elaboración de estándares para los documentos de requerimientos del cliente y la especificación del sistema de software. También, se desarrollaron las guías para realizar la primera entrevista con el cliente y la lista de atributos de calidad por revisar en el documento de requerimientos. Como lo indica Sommerville (1995), los documentos generados durante el desarrollo del proyecto de software son particularmente importantes, ya que es la única manera tangible de representar el software en las etapas tempranas de su evolución. Además, Humprey (1989) señala que la estandarización de los documentos utilizados en el proceso de desarrollo ayuda a reducir el problema de entrenamiento y facilita la verificación de los criterios de calidad acordados.

Un proyecto de modelado de procesos no termina con el rediseño del proceso, sino que se requiere entrenar a los usuarios de éste y evaluar el desempeño con los nuevos cambios. Además de incluir el tema de ingeniería de procesos en la parte teórica del curso, también se proporcionó asesoría a los estudiantes involucrados en la etapa de análisis, para resolver sus dudas respecto al uso de los modelos de procesos y documentos estandarizados. Sin el entrenamiento adecuado, los efectos de la actualización del proceso degradan su calidad, en lugar de lograr la meta de un proceso efectivo y eficiente (Sommerville, 1995).

5. Evaluación de la experiencia

Las mejoras al modelo fueron aplicadas en el periodo de septiembre a diciembre de 1999, en el curso de Ingeniería y metodología de la programación, en el que participaron los 15 estudiantes de nuevo ingreso al programa de maestría en Ciencias de la Computación en el CICESE. Se llevaron a cabo las estrategias señaladas en el apartado anterior y, al finalizar el trimestre, se entrevistó una muestra de tres estudiantes (ingeniero de requerimientos, arquitecto del sistema e ingeniero de control de calidad), ya que fueron los responsables de ejecutar las actividades descritas en los modelos rediseñados para la etapa de análisis del sistema. Durante la entrevista, cada estudiante expresó su percepción del curso a través de un cuestionario (Tabla II). El instrumento tiene nueve preguntas que se califican en una escala de Likert (donde 1 = totalmente en desacuerdo, 5 = totalmente de acuerdo) y se enfocan en los tópicos siguientes: congruencia del modelo del proceso con las actividades reales del proyecto, uso de estándares y capacitación sobre procesos.

Tabla II. Concentrado de frecuencias de la encuesta de opinión aplicada a los estudiantes (N = 3) sobre el enfoque de procesos en el curso de ingeniería y metodología de la programación

1 = Estoy totalmente en desacuerdo; 2 = Estoy en desacuerdo; 3 = No estoy seguro; 4 = Estoy de acuerdo; 5 = Estoy totalmente de acuerdo.	Escala de Likert				
	1	2	3	4	5
Congruencia					
El modelo es de utilidad para guiar las actividades del proceso.	-	-	-	2	1
El grado de detalle en el modelo es adecuado.	-	-	-	2	1
Las actividades descritas en el modelo corresponden a las del proyecto.	-	-	1	2	-
Las interacciones del modelo corresponden a las que suceden en el proyecto.	-	-	-	1	2
Es fácil identificar las dependencias del proceso.	-	-	1	-	2
Estándares					
Las plantillas de documentos son adecuadas.	-	-	2	-	1
Las listas de revisión de atributos de calidad son útiles.	-	-	-	2	1
Capacitación					
La clase teórica de procesos es útil.	-	-	1	1	1
La comunicación fuera de clase con el asesor es importante para aclarar dudas.	-	-	-	1	2

Como se puede observar, se obtuvieron resultados prometedores al utilizar el enfoque de procesos en el curso. La mayoría de las calificaciones se encuentran en las categorías “estoy de acuerdo” y “estoy totalmente de acuerdo”. Es natural que existan diferencias en el grado de detalle entre el proceso de modelado y las actividades reales del proyecto. Caputo (1998) señala que la meta del mejoramiento de procesos no es crear un proceso perfecto, sino generar procesos que permitan mejorar el desempeño. De hecho, Warboys *et al.* (1999) hacen la diferencia entre proceso genérico y proceso adaptado. En el caso particular de este estudio, se puede considerar que el modelo utilizado tiene los elementos fundamentales para lograr las metas del proceso, pero debido a la naturaleza del curso de ingeniería del software, no está adaptado y asociado a individuos y máquinas para llevar a cabo el proceso ni a las características particulares del proyecto que realizaron los estudiantes. Esto podría ser una de las razones para que algunas preguntas fueran calificadas como “no estoy seguro”.

Se requiere poner mayor atención a los documentos que se generan durante la etapa de análisis. Los resultados de la entrevista indican que falta claridad en la descripción de cada una de las partes del documento de requerimientos y especificación. También, indican que se puede integrar la herramienta para generar los diagramas en lenguaje UML con el procesador de textos para facilitar y automatizar la generación del documento de especificación de software.

- **Congruencia del modelo del proceso con las actividades reales del proyecto.** Los estudiantes señalaron que los diagramas resultaron útiles como guía para conocer el trabajo

que implica su puesto, porque les permiten visualizar, de manera simple, las actividades que están realizando y enfocar sus esfuerzos hacia una meta. El grado de detalle presentado en los modelos fue adecuado, ya que les brindó la facilidad para determinar las actividades que prosiguen sin necesidad de especificar una técnica particular para ejecutarlas. Los modelos presentados en el curso cumplen lo que establece Miers (1996) en el sentido de que una buena definición de proceso debe concentrarse en la esencia de éste, reflejando el mundo real y manejando la complejidad del mismo. Además, se ajustan a lo que indica Ould (1995), quien señala que una buena descripción del proceso será aquella que comunique detalladamente las actividades a quienes las efectuarán y que sea suficientemente precisa para permitir una evaluación de conformidad. También, como lo expresan Warboys *et al.* (1999), los RAD permiten la descripción de comportamientos complejos de una manera altamente legible y logran la captura apropiada de las reglas de la organización.

- **Estándares en el proyecto.** Los participantes indicaron que los estándares fueron una buena guía para elaborar los documentos de requerimientos y especificación, y sirvieron de referencia en las revisiones de este tipo de productos. Este resultado constata lo establecido por Humprey (1989), en el sentido de que la estandarización ayuda a reducir el problema de entrenamiento. Sin embargo, sugirieron cambiar el orden de algunas secciones de los documentos para mejorar la presentación, y solicitaron un ejemplo escrito completo que sirviera de guía.

- **Capacitación.** La clase teórica de procesos les permitió ubicarse en el contexto general de desarrollo de sistemas de software, aunque solicitaron que se les proporcionaran los modelos de proceso para cada uno de los puestos en la empresa. Consideraron que las asesorías fuera de clase permitieron exponer, de forma relajada, los aspectos difíciles en las actividades que estaban realizando y les sirvió para orientar sus esfuerzos en el desarrollo del sistema de software.

Los participantes apreciaron la importancia de compartir información en un proyecto colaborativo e identificaron la necesidad de los procesos de software para guiar las prácticas del equipo y hacer visible el proceso. Se percataron de los costos y beneficios del trabajo en equipo y comprendieron que un proyecto de desarrollo de software no sólo es generar código en un lenguaje de programación.

6. Conclusiones

Se ha presentado el problema de coordinación de esfuerzos entre los participantes de un proyecto de desarrollo de software y las medidas que se tomaron para mitigarlo, desde la perspectiva de procesos en el curso de ingeniería de software. Se describieron, de manera general, las características del curso de Ingeniería y metodología de la programación que se imparte en el CICESE y los pasos que se siguieron en el proyecto de modelado de procesos para investigar las interacciones, actividades, roles, documentos y metas que forman parte del proceso de desarrollo de software utilizado en la parte práctica de la asignatura.

Se observó que las técnicas de modelado de procesos son herramientas eficaces para hacer visible el proceso, permiten identificar la problemática particular de los proyectos de software analizados e identificar las áreas del proceso que son susceptibles de mejora. En esta investigación, los modelos aislaron las cuestiones relacionadas con la fase de análisis de sistemas, lo que facilitó la actualización de los tópicos en la sesión teórica del curso, y sirvieron de guía para rediseñar los procesos que pertenecen a dicha etapa del proyecto.

Los modelos descritos contribuyen a que los estudiantes tengan una mejor comprensión de sus responsabilidades en el proceso y les permiten enfocar sus esfuerzos hacia las actividades que son significativas en el desarrollo del proyecto, esto les brinda autonomía al reducir el número de interacciones con el instructor. Los resultados preliminares, utilizando procesos rediseñados, sugieren que los modelos son una guía poderosa para entrenar a los participantes del proyecto. Los modelos facilitan la colaboración entre los distintos miembros del grupo al determinar explícitamente los tipos de interacción que existen en cada etapa del desarrollo de un sistema de software. No es necesario que los estudiantes investiguen cuáles son las actividades que corresponden a sus puestos, ni con quién deben interaccionar durante el proceso, sólo deben preocuparse por comprender cómo realizar sus tareas y adaptarlas al proyecto de desarrollo de software en el que están trabajando.

7. Trabajo futuro

Actualmente se está trabajando en los procesos clave del nivel 2 del Modelo de maduración de capacidades del SEI (*Software Engineering Institute*), en los que se incluyen: administración de requerimientos, planificación del proyecto, seguimiento del proyecto, aseguramiento de calidad y administración de configuración (Paulk *et al.*, 1995). Estos procesos clave servirán de guía para el siguiente curso. Aunque los resultados iniciales indican que es favorable el modelado de procesos para que los estudiantes conozcan las actividades que deben ejecutar, sobre todo cuando no tienen experiencia en el área, es necesario probar este modelo en cursos del posgrado y licenciatura para corroborar los resultados preliminares descritos con anterioridad.

Un segundo paso, conociendo las virtudes de estos modelos, es extenderlos a un ambiente de desarrollo distribuido, donde los estudiantes están en localidades geográficas diferentes. En este ambiente, la investigación se enfocará a las interfaces que deberán existir en el proceso, para que funcione de manera adecuada; además de recomendar herramientas de apoyo a la comunicación y coordinación.

Referencias

Arthur, L. J. (1992). *Improving software quality: insider's guide to TQM*. New York: John Wiley & Sons.

Ayer, S. (1992). *Documenting the software development process: A handbook of structured techniques*. New York: McGraw-Hill.

Bagert, D. J., Hilburn, T. B., Hislop, G., Lutz, M., McCracken, M. y Mengel, S. (1999). *Guidelines for software engineering education version 1.0*. (Reporte técnico CMU/SEI-99-TR-032 ESC-TR-99-002). Pittsburgh, PA: Carnegie Mellon University, Software 555Engineering Institute.

Caputo, K. (1998). *CMM Implementation guide: Choreographing software process improvement*. Reading, MA: Addison Wesley-Longman.

Clark, B. K. (2000). Quantifying the effects of process improvement on effort. *IEEE Software*, 17 (6), 65-70.

Collofello, J. S., Kantipudi y M., Kanko, M. A. (1994). Assessing the software process maturity of software engineering courses. En *Proceedings of the 25th SIGCSE Technical Symposium on Computer Science Education* (pp. 16-20). Phoenix, AR: ACM Press.

Curtis, B., Kellner, M. I. y Over, J. (1992). Process modeling. *Communications of the ACM*, 35 (9), 75-90.

Donaldson, S. E. y Seigel, S. G. (1997). *Cultivating successful software development: a practitioner's view*. Upper Saddle River, NJ: Prentice-Hall PTR.

Dorfman, M. y Thayer, R. H.. (1990). *Standards, guidelines, and examples on system and software requirements engineering*. Los Alamitos, CA: IEEE Computer Society Press.

Fairley, R. (1985). *Ingeniería de software*. (Trads. A. Sánchez y P. L. Flores-Suárez). México: McGraw-Hill. (Trabajo original publicado en 1985).

García Mireles, G. A. y Rodríguez Jacobo, J. (2000). *Manual de procedimientos para la elaboración del documentos de requerimientos en el desarrollo de software* (Informe técnico CICESE CTCCT2001, serie Electrónica y Telecomunicaciones). Ensenada, B. C.: CICESE.

García Mireles, G. A. y Rodríguez Jacobo, J. (2000, junio). Propuesta para mejorar la enseñanza de la ingeniería del software basada en proyectos. Cartel presentado en el *Congreso de Educación Abierta y a Distancia* (CEAD 2000). Ensenada, B. C.: ANUIES-UABC-CICESE.

Herbsleb, J., Zubrow, D., Goldenson, D., Hayes, W. y Paulk, M. (1997). Software quality and the capability maturity model. *Communications of the ACM*, 40 (6), 30-40.

Humphrey, W. S. (1989). *Managing the software process*. Reading, MA: Addison-Wesley.

Jaccheri, M. L. y Lago, P. (1997). Applying software process modeling and improvement in academic setting. En *Proceedings of the 10th Conference on Software Engineering Education and Training* (pp. 13-27). Virginia Beach, VA: IEEE Computer Society Press.

Kawalek, P. (1994). Comments on the use of RADs in case studies. Disponible en FTP: <ftp://ftp.cs.man.ac.uk/pub/IPG/pk94.zip> (10 de julio de 2000).

Kraut, R. L. y Streeter, L. A. (1995). Coordination in software development. *Communications of the ACM*, 38 (3), 63-81.

Licea, G., Rodríguez, J. y Favela, J. (1996). Evolución de procesos de desarrollo en la enseñanza de ingeniería de software. En *Memorias del V Congreso Iberoamericano de Educación Superior en Computación* (pp. 223-231). México: Facultad de Ciencias, UNAM.

Maurer, F. y Kaiser, G. (1998). Software engineering in the internet age. *IEEE Internet Computing*, 2 (5), 22-24.

Mayr, H. (1997). Teaching software engineering by means of a “virtual enterprise”. En *Proceedings of the 10th Conference on Software Engineering Education and Training*, (pp. 176-184). Virginia Beach, VA: IEEE Computer Society Press.

Miers, D. (1996). Use of tools and technology within a BPR initiative. En C. Coulson-Thomas (Ed.), *Process re-engineering: Myth and reality*, (pp.142-165). London: Kogan Page Limited.

Oktaba, H. e Ibarguengoitia, G. (1998). Software process modeled with objects: Static view. *Computación y Sistemas*, 1 (4), 228-238.

Ould, M. A. (1995). *Business processes: Modeling an analysis for re-engineering and improvement*. Chichester: John Wiley and Sons.

Paulk, M. C., Weber, C. V., Curtis, B. y Chrissis, M. B. (Eds.). (1995). *The capability maturity model: Guidelines for improving the software process*. Reading, MA : Addison-Wesley.

Pressman, R. (1993). *Ingeniería del software. Un enfoque práctico* (3^a. edición) (Trads. C. Cervigon y L. Hernández Yáñez). Madrid: McGraw-Hill. (Trabajo original publicado en 1992).

Robillard, P. N. (1998). Measuring team activities in a process-oriented software engineering course. En *Proceedings of the 11th Conference on Software Engineering Education and Training*. Atlanta: IEEE Computer Society Press.

Rombach, H. D. (1990). Software specifications: A framework. En M. Dorfman y R. H. Thayer (Eds.), *Standards, guidelines, and examples on system and software requirements engineering* (pp. 368-407). sLos Alamitos, CA: IEEE Computer Society Press.

Sommerville, I. (1995). *Software engineering* (5ª edición). Harlow: Addison-Wesley.

Tomayko, J. E. (1987). *Teaching a project-intensive introduction to software engineering* (Reporte técnico CMU/SEI-87-TR-20 ESD-TR-87-171). Pittsburgh, PA: Carnegie Mellon University, Software Engineering Institute.

Upchurch, R. L. y Sims-Knight, J. E. (1997). Designing process-based software curriculum. En *Proceedings of the 10th Conference on Software Engineering Education and Training* (pp. 28-38). Virginia Beach, VA: IEEE Computer Society Press.

Upchurch, R. L. y Sims-Knight J. E. (1998). In support of student process improvement. En *Proceedings of the 11th Conference on Software Engineering Education and Training*. Atlanta: IEEE Computer Society Press.

Warboys, B., Kawalek, P., Robertson, I. y Greenwood, M. (1999). *Business information systems: A process approach*. Londres: McGraw-Hill.

Wastell, D. G., White, P. y Kawalek, P. (1994). A methodology for business process redesign: Experiences and issues. *Journal of Strategic Information Systems*, 3 (1), 23-40.