

Definición de un Sistema de Aseguramiento de Calidad para Actividad de Titulación en un Curriculum de Ingeniería de Software

Eduardo G. Jara

Universidad del Bío-Bío, Departamento de Sistemas de Información,
Concepción, Chile, 4081112
ejara@ubiobio.cl

y

Salomé R. Castro

Universidad del Bío-Bío, Departamento de Sistemas de Información,
Concepción, Chile, 4081112
scastro@ubiobio.cl

RESUMEN

La solución a la crisis del software pasa por un cambio de mentalidad de sus protagonistas: programadores, analistas, jefes de proyecto, etc. Se necesita ver al software como un producto que debe ser pensado, diseñado y desarrollado por un grupo multidisciplinario dentro de un sistema que asegure su calidad. Este cambio puede ser logrado, en parte, a través de la educación de la nueva generación de profesionales del software, de modo que para ellos los conceptos de calidad y desarrollo en equipo sean naturales.

Este artículo describe la definición de un Sistema de Aseguramiento de Calidad para la actividad de titulación de un curriculum de ingeniería de software en la Universidad del Bío-Bío. El sistema está basado sobre el modelo de calidad Capability Maturity Model (CMM) y el proceso de desarrollo Rational Unified Process (RUP).

Palabras claves: Aseguramiento de calidad, Proceso de desarrollo, Enseñanza de ingeniería de software.

ABSTRACT

In order to solve the software crisis a mental change in actors (programmers, system analysts, project managers, etc.) involved in the development process is needed. A software must be viewed as a product that must be thought, designed, and developed by a multidisciplinary team whose purpose is to assure its final quality. This change may be achieved by educating the future generation of software development professionals in order to that both quality and development through team work became second nature to them.

This article describes the definition of a Quality Assurance System needed in the final stages of the software engineering curriculum at the University of Bío-Bío. The system is based on the quality model Capability Maturity Model (CMM) and the development process Rational Unified Process (RUP).

Keywords: Quality assurance, Development process, Software engineering education.

1 INTRODUCCION

La industria del software sufre del un mal endémico que se remonta a sus orígenes y que hasta ahora no parece tener una solución definitiva, es la llamada Crisis del Software. Ha habido muchas iniciativas para atenuar los efectos de esta crisis. Una de las más importantes es la aplicación de sistemas de aseguramiento de calidad al desarrollo de software, los que se basan sobre las siguientes ideas:

- El software debe ser pensado, diseñado y desarrollado como un producto sujeto a normas de calidad.
- El software es un producto desarrollado por grupos de personas cuya interacción debe ser gestionada.
- El énfasis en el proceso de desarrollo asegura un producto adecuado a los requisitos de los clientes.

La solución a la crisis del software pasa por un cambio de mentalidad de los protagonistas de su desarrollo: programadores, analistas, jefes de proyecto, usuarios, etc. Esto conduce a una idea que cada vez encuentra más fundamentos y adherentes: la crisis del software puede ser reducida a través de la educación [1] [5], es decir, educando adecuadamente la próxima generación de profesionales del software, de modo que en ellos sean naturales e inmediatos conceptos como calidad y desarrollo en equipo. Sin embargo, ha pasado el tiempo y las universidades no han logrado un consenso sobre cómo educar a sus futuros profesionales del software.

En el Departamento Sistemas de Información de la Universidad del Bío-Bío se dictan carreras de pregrado en las que se enseña a desarrollar software, pero impera un enfoque “artesanal” y las nociones más formales e ingenieriles no son apoyadas con actividades prácticas que permitan que los alumnos las internalicen y las hagan suyas. Esto ha llevado a la necesidad de intervenir el curriculum y modificar cursos para incorporar las nociones de calidad en términos teóricos y prácticos.

En este artículo se presenta el diseño de un Sistema de Aseguramiento de Calidad (Sistema de Aseguramiento de Calidad) para la actividad de titulación de ingenieros de software. En la Sección 2 se describe la situación actual de las carreras de pregrado del departamento de Sistemas de Información de la Universidad del Bío-Bío en lo concerniente a la enseñanza de desarrollo de software y las acciones que se están realizando para mejorarla. En la Sección 3 se describen las bases del Sistema de Aseguramiento de Calidad: modelo de calidad, proceso de desarrollo y estructura organizacional. La Sección 4 está dedicada a la descripción del Sistema de Aseguramiento de Calidad desde el punto de vista de sus niveles de documentación. En la Sección 5 se presentan las restricciones del sistema producto de las limitaciones organizacionales. Y, finalmente, en la Sección 6 se describe resumidamente la manera en que se implantará el sistema y las acciones futuras.

2 SITUACION ACTUAL

En el Departamento Sistemas de Información de la Universidad del Bío-Bío se dictan dos carreras de pregrado: Ingeniería en Computación (8 semestres) e Ingeniería Informática (12 semestres). Cada semestre los alumnos tienen un promedio de 5 cursos. En la Figura 1 se muestra, para ambas carreras, los cursos directamente relacionados con el desarrollo de software.

Ingeniería en Computación		Ingeniería Informática	
I. sem.	Informática	Informática	I. sem.
II. sem.	Diseño de algoritmos	Diseño de algoritmos	III. sem.
III. sem.	Técnicas de programación	Programación Estructurada	IV. sem.
IV. sem.	Taller de lenguaje I	Sistemas de Información I	VI. sem.
V. sem.	Taller de lenguaje II	Sistemas de Información II	VII. sem.
	Sistemas de Información I	Métodos de desarrollo I	VIII. sem.
VI. sem.	Sistemas de Información II	Métodos de desarrollo II	IX. sem.
VII. sem.	Taller de Construcción	Ingeniería de software	X. sem.
	Ingeniería de Software		
VIII. sem.	Proyecto de Título		

Fig. 1. Cursos de desarrollo de software de las carreras de pregrado dictadas por Departamento Sistemas de Información

Aunque actualmente los alumnos utilizan métodos para el desarrollo de software, no hacen de ellos un proceso de ingeniería, porque en la práctica no realizan estimaciones, ni planifican rigurosamente las actividades necesarias para construir el software deseado, y si lo hacen, no se rigen por ellas. Por lo anterior, es difícil que los alumnos incorporen prácticas de aseguramiento de calidad al proceso de desarrollo, como por ejemplo, revisiones, inspecciones, auditorías, etc.

Un curso especialmente importante en Ingeniería en Computación es Proyecto de Título (en el octavo semestre). Este curso corresponde a la actividad de titulación y consiste en la construcción de un producto de software por uno o dos alumnos bajo la dirección de un profesor guía. Los temas son variados y generalmente son presentados por los alumnos sobre la base de requerimientos reales de empresas de la región.

En particular, en el curso Proyecto de Título tanto alumnos como profesores no cuentan con un proceso de desarrollo definido, ni procedimientos, ni estándares que los orienten en sus responsabilidades en el proceso de desarrollo del software. Por ejemplo, la evaluación que los profesores realizan al proyecto se concentra en el producto final, porque no existen normas que exijan evaluaciones a los resultados intermedios, quedando esto a criterio del profesor guía a cargo del proyecto, pero aún así, no existe una forma de demostrar que se han hecho tales evaluaciones, y que al mismo tiempo asegure la calidad del software ante terceros. Por lo anterior se presentan diversos problemas, entre ellos:

- Existe un número importante de proyectos que sobrepasan en tiempo estipulado (un semestre) para su desarrollo.
- Existe demasiada dependencia del proyecto con respecto a los alumnos que comienzan el proyecto, ya que si deciden retirarse de éste, no existen los medios que faciliten a otros alumnos, entender, reutilizar y continuar con el trabajo ya iniciado. Esto también, dificulta la posibilidad de desarrollar grandes proyectos, en el que trabajen varios alumnos en forma independiente, paralela o secuencialmente.
- Es difícil para los profesores realizar una evaluación objetiva y comparativa entre proyectos, porque la mayoría se desarrolla de diferentes formas.
- Los alumnos al comenzar el desarrollo de su proyecto se sienten desorientados, ya que frecuentemente no saben por dónde comenzar.

Acciones Correctivas

El Departamento de Sistemas de Información comenzó en 1999 un proceso de revisión y modificación de los contenidos de los cursos y los métodos de enseñanza. En lo concerniente al desarrollo de software se acordó incentivar la planificación y medición mediante la incorporación de estándares y procedimientos que complementen el método de desarrollo utilizado, haciendo énfasis en:

- Definir y documentar todas las actividades que los alumnos deberán llevar a cabo cuando desarrollen un software.
- Definir e implantar, procedimientos y estándares, para asegurar que tanto las actividades del proceso de desarrollo como los resultados que de él se obtienen, se cumplen según lo planificado.

Las nociones de calidad deben ser parte de todos los cursos relacionados con el desarrollo de software de las dos carreras de pregrado (ver Figura 1). Empezando desde el primer semestre e incorporando nuevos elementos a medida que se avanza en el curriculum en un constante proceso de maduración. Para el curso Proyecto de Título, que representa la culminación de este proceso, se desarrolló un Sistema de Aseguramiento de Calidad cuyas bases se presentan en la siguiente sección.

3 BASES DEL SISTEMA DE ASEGURAMIENTO DE CALIDAD

Para la definición del Sistema de Aseguramiento de Calidad se necesitan tres elementos: un modelo de calidad, un proceso de desarrollo y una estructura organizacional. En las siguientes subsecciones se explica cada uno de ellos.

CMM como Modelo de Calidad

El Sistema de Aseguramiento de Calidad debía realizarse a partir de un modelo de calidad ampliamente aceptado y orientado al desarrollo de productos de software, se estudiaron dos modelos: ISO 9000-3 [2] [6] y CMM (Capability Maturity Model) [4] [8]. Se optó por este último por las siguientes razones:

- CMM, es un modelo creado originalmente para mejorar procesos de desarrollo de software. En cambio ISO 9000-3, que también es un modelo para mejorar el desarrollo de software, es una adaptación de ISO 9001, que es un modelo genérico para cualquier tipo de proceso de desarrollo, y en cualquier tipo de empresa. En consecuencia, el contenido y descripción de CMM está en un lenguaje acorde con un proceso de desarrollo de software, facilitando así, su entendimiento y aplicabilidad.
- La estructura de CMM permite mejorar el proceso de software en forma evolutiva, a medida que se van incorporando los niveles de madurez al proceso de software. En cambio ISO 9000-3 tiene una estructura plana, en donde hay veinte requerimientos de calidad que deben ser definidos e implantados, sin entregar una guía que indique prioridad entre dichos requerimientos de calidad.
- CMM en su descripción identifica explícitamente actores, y los roles que dichos actores deben desempeñar para llevar a cabo las actividades que conformarán el Sistema de Aseguramiento de Calidad, facilitando así la interpretación del modelo. En cambio ISO 9000-3, sólo describe las actividades que conforman el Sistema de Aseguramiento de Calidad, sin identificar actores o roles que pudieran estar involucradas con las actividades a desarrollar.
- CMM permite seguir mejorando el proceso a través de niveles crecientes de madurez hasta aspectos como la administración cuantitativa del proceso, administración de cambios tecnológicos y administración del cambio de proceso. Sin embargo, estos elementos no están incluidos en el modelo ISO 9000-3.

Características Principales de CMM

El modelo CMM define cinco Niveles de Madurez para un proceso de software, los cuales son secuenciales y cada uno establece nuevas acciones para mejorar el proceso. Estos niveles son: *Inicial, Repetible, Definido, Administrado y Optimizado*.

Por definición, se encuentra en el Nivel 1 (Inicial) toda organización cuyo proceso de desarrollo es improvisado y sus resultados dependen de los esfuerzos personales. Por las razones explicadas en la sección anterior queda claro que, como organización desarrolladora de software, el Departamento de Sistemas de Información se encuentra en el Nivel 1 (Inicial). Por lo tanto el Sistema de Aseguramiento de Calidad debe ser definido en torno al Nivel 2 (Repetible).

Cada Nivel de Madurez, exceptuando en Nivel 1, está compuesto de Areas Claves de Proceso que dan cuenta de los elementos que deben ser logrados para certificarse en dicho nivel. El Nivel 2 (Repetible) tiene las siguientes Areas Clave de Proceso:

- Administración de Requerimientos
- Planificación del Proyecto de Software
- Seguimiento y Control del Proyecto de Software
- Administración de Subcontrato de Software
- Garantía de Calidad del Software
- Administración y Configuración del Software

A su vez, cada Area Clave de Proceso está organizada en cinco secciones llamadas *Características Comunes*, que son: Compromisos para el Desempeño; Habilidades para el Desempeño; Actividades a Realizar; Medición y Análisis; y Verificación de la Implementación.

RUP como Proceso de Desarrollo

Los modelos de calidad como CMM e ISO 9000-3 asumen que debe existir un proceso de desarrollo de software en la organización, pero no se establecen restricciones mayores, excepto que el proceso debe ser compatible con el modelo. En el presente caso se optó por Rational Unified Process (RUP) [3] por las siguientes razones

- RUP es explícito en la definición de artefactos y su trazabilidad, es decir, contempla en relación causal de los artefactos creados desde los requerimientos hasta la implementación y pruebas. Esto facilita especialmente la realización de las Areas Clave de Proceso de CMM: Seguimiento y Control del Proyecto de Software; y Administración y Configuración del Software.
- RUP identifica claramente a los profesionales (actores) involucrados en el desarrollo del software y sus responsabilidades en cada una de las actividades. Además, explícitamente indica qué actor es responsable de qué artefacto en cada actividad. Esto facilita especialmente la realización de las Areas Clave de Proceso de CMM: Planificación del Proyecto de Software; y Administración y Configuración del Software.

Características Principales de RUP

Como se muestra en la Figura 2, en RUP el proceso de desarrollo se divide en cuatro grandes fases:

- Inicio, que termina cuando se han determinado los objetivos del ciclo de vida.
- Elaboración, que termina cuando se ha implementado la arquitectura del ciclo de vida
- Construcción, que termina cuando se ha logrado la capacidad operacional inicial del software.
- Transición, que termina con la entrega del producto al cliente.

Cada fase se divide en Iteraciones que consisten en la realización secuencial de los Flujos de Trabajo Fundamentales, es decir, se parte del Modelamiento del Negocio y se finaliza en la realización de las Pruebas. Así, un proyecto puede ser visualizado como una secuencia de “microproyectos” ajustados al modelo tradicional de cascada dentro de cada iteración.

A medida que se avanza en el proyecto, es decir, cuando se va pasando de una fase a otra, la importancia relativa de cada uno de los Flujos de Trabajo va cambiando. (Véase las áreas sombreadas de la Figura 2). Así, en las iteraciones de la Fase de Inicio el trabajo se centra principalmente en el Modelamiento del Negocio y en la captura y especificación de Requisitos. Pero en la fase de Construcción el desarrollo está enfocado en la Implementación (codificación) y, en menor medida, en el Diseño.

Cada Flujo de Trabajo Fundamental está definido por: actividades, actores y artefactos. Donde cada actividad es responsabilidad de un actor y consiste en la creación y/o complementación de ciertos artefactos. Por ejemplo, en el Flujo de Trabajo *Requisitos* existe la actividad *Encontrar Actores y Casos de Uso* que es responsabilidad del actor *Analista de Sistemas*. Esta actividad tiene como entrada los artefactos: *Modelo de Dominio*, *Requisitos Adicionales* y *Lista de Características*; y como salida: *Modelo de Casos de Uso* y el *Glosario de Términos*.

Además, RUP contempla Flujos de Trabajo de Soporte que involucran actividades de administración y planificación de recursos humanos, tecnológicos y financieros.

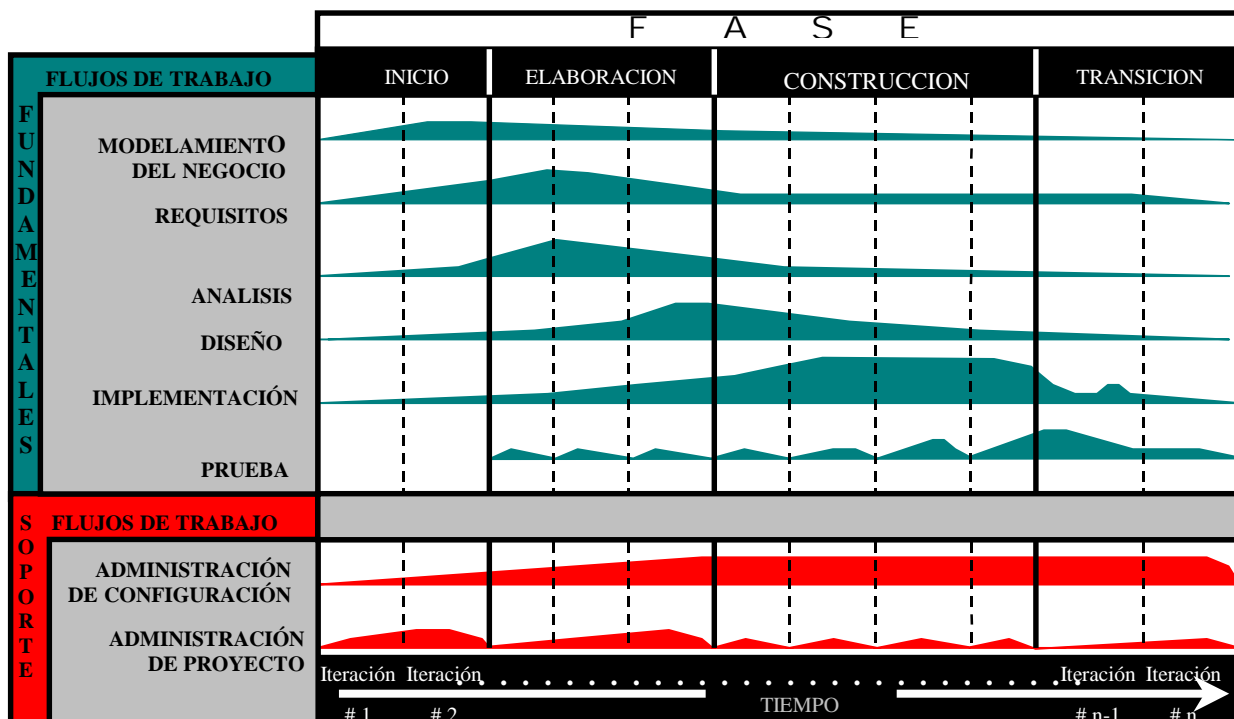


Fig. 2. Flujos de trabajo, fases e iteraciones de Rational Unified Process (RUP)[3]

Estructura Organizacional

El proceso de desarrollo (en este caso RUP) permite materializar los requerimientos de calidad declarados en el modelo de calidad (en este caso CMM), pero se requiere de una estructura organizacional que defina claramente quiénes estarán involucrados (los actores) en el sistema y cuáles serán sus roles. En el Sistema de Aseguramiento de Calidad para el curso Proyecto de Título participan cuatro actores:

- Comisión de Control: Está formada por el Director de la Carrera Ingeniería en Computación y por otros dos profesores del Area de Ingeniería de Software. Su labor es coordinar las actividades del curso Proyecto de Título y cumplir el rol de Gerencia Superior definido por el modelo CMM. Esto último implica la revisión y control permanente del Sistema de Aseguramiento de Calidad.
- Alumno: Es quien debe llevar a cabo las actividades planificadas, y las acciones correctivas cuando el plan de proyecto no está siendo logrado, ya sea ajustando el plan o su desempeño.
- Profesor Guía: Es quien evalúa los artefactos realizados por el Alumno durante todo el desarrollo del Proyecto, con el fin de que éste último no se desvíe del plan de proyecto. Además, sirve de asesor técnico permanente al alumno en lo referido al modelo CMM y al proceso RUP. Participa en la calificación final del proyecto.
- Profesor Informante: Es quien verifica que tanto el Alumno como el Profesor Guía, cumplen con las especificaciones del Proceso de Desarrollo y del Sistema de Calidad. Participa en la calificación final del proyecto.

En la Figura 3 se muestra la relación entre los actores del Sistema de Aseguramiento de Calidad y los roles requeridos por el modelo CMM

ACTORES DEL MODELO CMM	ACTORES DEL SISTEMA DE CALIDAD			
	Alumno	Profesor Guía	Profesor Informante	Comisión de Control
Grupo de Ingeniería de Software	R	AT		
Grupo de estimación de Software	R	AT		
Grupo de Ingeniería de Sistemas	R	AT		
Gerente de Proyecto de Software	R			
Gerente de Proyecto		R		
Gerencia Superior				R
Grupo de Administración de Configuración del Software	R	AT		
Soporte de Documentación	R	AT		
Grupo de Garantía de Calidad del Software			R	
Grupo de Pruebas del Sistema	R	AT		

R: Responsable **AT:** Asesor Técnico

Fig. 3. Actores del Sistema de Aseguramiento de Calidad y Actores del modelo CMM

4 EL SISTEMA DE ASEGURAMIENTO DE CALIDAD

Resumidamente, el Sistema de Aseguramiento de Calidad funciona de la siguiente manera:

1. Los Alumnos presentan las propuestas de proyecto.
2. La Comisión de Control estudia las propuestas y las aprueba o rechaza.
3. La Comisión de Control designa a lo Profesores Guía e Informante para los proyectos aprobados.
4. Se procede con el desarrollo del proyecto, para ello Alumnos, Profesores (Guías e Informantes) y Comisión de Control realizan las actividades definidas por el Sistema de Aseguramiento de Calidad.
5. Al finalizar el proyecto:
 - 5.1.El alumno es calificado por los Profesores Guía e Informante sobre la base del cumplimiento de las normas de calidad impuestas por el Sistema de Aseguramiento de Calidad.

5.2. La Comisión de Control evalúa la labor de los Profesores Guía e Informante.

5.3. La Comisión de Control evalúa el Sistema de Aseguramiento de Calidad y determina cambios en él.

Cada uno de los pasos descritos debe realizarse con apego a los procedimientos operativos y estándares definidos en la documentación del sistema. Para cualquier tipo de organización, al definir un Sistema de Aseguramiento de Calidad, se deben desarrollar tres niveles de documentación [7] [9], que se describen en las siguientes subsecciones.

Primer Nivel de Documentación

Especifica en términos generales qué actividades deberán integrar el Sistema de Aseguramiento de Calidad, que será implantado en la organización. Este nivel contiene los siguientes elementos:

- Declaración de Visión: Proyecciones de la administración sobre el lugar que ocupará la organización en el futuro.
- Declaración de Misión: Compromiso de la administración para alcanzar la Visión.
- Política de Calidad: Posición de la organización, en cuanto a la manera en que la calidad afectará la manera de cumplir con la Misión.
- Requerimientos de Calidad: Conjunto de actividades que la organización debe llevar a cabo, para asegurar la calidad tanto del proceso como el producto que desarrolla

La Visión, Misión y Políticas de Calidad fueron desarrollados a partir de los lineamientos estratégicos del Departamento de Sistemas de Información.

Los Requerimientos de Calidad se identifican en modelos de calidad como ISO 9000 o CMM; o la organización define sus propios Requerimientos de Calidad, basándose en sus necesidades. Por lo indicado en la sección anterior se optó por tomar los requerimientos del modelo CMM, en particular del Nivel 2 de Madurez. En la Figura 4 se muestran los Requerimientos de Calidad correspondientes al Area Clave de Proceso: Administración de Requerimientos.

Segundo Nivel de Documentación

Este nivel incluye especificaciones detalladas, orientadas a la administración, para explicar cómo se llevarán a cabo las actividades que integran el Sistema de Aseguramiento de Calidad. Este nivel está compuesto básicamente por Procedimientos Administrativos, que son declaraciones de direcciones sistemáticas, sobre cómo la organización debe llevar a cabo cada uno de los Requerimientos de Calidad, definidos en el Primer Nivel de Documentación. En la Figura 5 se muestran los Procedimientos Administrativos correspondientes a algunos Requerimientos de Calidad de la Administración de Requerimientos. Obsérvese que los Requerimientos de Calidad (columna izquierda) están expresados en términos de los actores del modelo CMM (Gerencia Superior, Gerente de Proyecto y Grupo de Garantía de Calidad), mientras que los Procedimientos Administrativos (columna derecha) lo están en términos de los actores del Sistema de Aseguramiento de Calidad (Comisión de Control, Profesor Informante, Profesor Guía y Alumno).

Tercer Nivel de Documentación

Este nivel incluye especificaciones punto a punto, explícitas y concisas para llevar a cabo cualquier tarea en la organización. Está compuesto básicamente por Procedimientos Operativos que describen cada paso que se debe realizar para concretar una tarea o actividad; y Estándares que se utilizan con el fin de registrar datos o información de algo específico. Estos procedimientos y estándares han sido divididos en tres grupos:

1. Los relacionados con el desarrollo del curso Proyecto de Título.
2. Los relacionados con el desarrollo de producto de software.
3. Los que guían la implantación y mejoramiento del Sistema de Aseguramiento de Calidad.

Esta división facilita el uso y mantención del sistema. Por ejemplo, si hay cambios en las normas administrativas que afecten el desarrollo de los cursos en general, entonces sólo se verán afectados los procedimientos y estándares relacionados con el desarrollo del curso Proyecto de Título.

En la Figura 6 se muestra el procedimiento operativo Revisión de Artefactos al Final de Flujo de Trabajo, cuyo código es PPT-07.00. Este procedimiento está relacionado con el desarrollo del producto de software. Obsérvese que se hace referencia a varios estándares por medio de sus códigos (EPT-01.00, EPT-05.00 y EPT-06.00).

AREA CLAVE DE PROCESO: ADMINISTRACION DE REQUERIMIENTOS	
Metas:	
<ol style="list-style-type: none"> 1. Los requerimientos del sistema, asignados al software, son controlados para establecer una línea base que será utilizada en la administración e ingeniería de software. 2. Los planes, artefactos y actividades de software se mantienen consistentes con los requerimientos del sistema asignados al software. 	
CARACTERISTICA COMUN	REQUERIMIENTOS DE CALIDAD
Compromisos para el desempeño	1. El Proyecto sigue una política organizacional escrita, para la administración de los requerimientos del sistema asignados al SW.
Habilidades para el desempeño	<ol style="list-style-type: none"> 1. Para cada proyecto, se establece la responsabilidad de analizar los requerimientos del sistema y asociarlos al hardware, software y otros componentes del sistema 2. Los requerimientos asignados deben estar documentados. 3. Se proporcionan los recursos y el financiamiento necesario para la administración de requerimientos asignados. 4. Los miembros del grupo de ingeniería de software, y de otros grupos relacionados, son capacitados para realizar sus actividades de administración de requerimientos.
Actividades a Realizar	<ol style="list-style-type: none"> 1. El grupo de ingeniería de software revisa los requerimientos antes de ser incorporados al proyecto. 2. El grupo de ingeniería de software usa los requerimientos asignados como base para establecer los planes, artefactos y actividades de SW. 3. Los cambios a los Requerimientos asignados son revisados e incorporados en el proyecto de Software.
Medición y Análisis	1. Se realizan mediciones para determinar el estado de las actividades de administración de requerimientos.
Verificación de la Implementación	<ol style="list-style-type: none"> 1. Las actividades de administración de Requerimientos asignados son revisadas periódicamente con la Gerencia Superior. 2. Las actividades de administración de Requerimientos asignados, son revisadas periódicamente con el Gerente de Proyecto y como respuesta a eventos. 3. El Grupo de Garantía de Calidad revisa o audita las actividades de administración de requerimientos e informa los resultados.

Fig. 4. Primer Nivel de Documentación: Requerimientos de Calidad

AREA CLAVE DE PROCESO: ADMINISTRACION DE REQUERIMIENTOS CARACTERÍSTICA COMÚN: VERIFICACIÓN DE LA IMPLEMENTACIÓN	
REQUERIMIENTOS DE CALIDAD	PROCEDIMIENTOS ADMINISTRATIVOS
Las actividades de administración de Requerimientos asignados son revisadas periódicamente con la <u>Gerencia Superior</u> .	Con el fin de evaluar el resultado de la implantación del Sistema de Aseguramiento de Calidad, y poder incorporar mejoras al Proceso de Desarrollo de SW en el curso Proyecto de Título, es necesario efectuar Revisiones a los proyectos en desarrollo. Dichas Revisiones son semestrales y orientadas al conjunto de proyectos realizados y son responsabilidad de la <u>Comisión Control (CC)</u> . Las revisiones estarán basadas en la información entregada por los <u>Profesores Informantes</u> , producto de las auditorías realizadas a los proyectos en desarrollo.
Las actividades de administración de Requerimientos asignados, son revisadas periódicamente con el <u>Gerente de Proyecto</u> y como respuesta a eventos.	Con el fin de evitar desviaciones de los objetivos del proyecto, y llevar a cabo acciones correctivas, el <u>Profesor Guía</u> debe reunirse periódicamente con el <u>Alumno</u> para revisar las actividades y artefactos desarrollados.
El <u>Grupo de Garantía de Calidad</u> revisa o audita las actividades de administración de requerimientos e informa los resultados.	Para evaluar si el <u>Alumno</u> , ha cumplido con las especificaciones del proceso, el <u>Profesor Informante</u> debe llevar a cabo auditorías, en las cuales se debe verificar que: <ol style="list-style-type: none"> 1. Los requerimientos son revisados antes de su implementación, y los problemas son resueltos antes de que el <u>Alumno</u> se comprometa con ellos. 2. Se revisan apropiadamente los planes, artefactos y actividades cuando los requerimientos cambian. 3. Se revisan los cambios en los compromisos acordados, productos de los cambios en los requerimientos.

Fig. 5. Segundo Nivel de Documentación: Procedimientos Administrativos

DEPARTAMENTO SISTEMAS DE INFORMACIÓN UNIVERSIDAD DEL BIO-BIO		
N° de Procedimiento: PPT-07.00		Número de Hoja: 2 de 02
Elaboró: M.Lagos		
TÍTULO: Revisión de Artefactos al Final de Flujo de Trabajo		
<p>Objetivo: Establecer las actividades que deben llevarse a cabo para revisar los Artefactos una vez terminado un Flujo de Trabajo.</p> <p>Alcance: Este procedimiento será utilizado en:</p> <ul style="list-style-type: none"> • Todos aquellos proyectos TIPO A, TIPO B y TIPO AB. • En todas y cada una de las Fases del Proceso RUP (Inicio, Elaboración, Construcción y Transición). <p>Políticas:</p> <ul style="list-style-type: none"> • Durante todo el transcurso de esta fase, el Profesor Guía debe asesorar al Alumno en el desarrollo de los Artefactos, sin embargo, en las revisiones de éstos se espera que el Profesor Guía sea lo más objetivo posible. • Tanto Alumnos como Profesores deben registrar en la Hoja de Esfuerzo, el tiempo utilizado en el desarrollo de cada Actividad o artefacto. (Usar EPT-05.00 y EPT-06.00). 		
PASO	DESCRIPCIÓN	ACTOR
10	Entregar al Profesor Guía los artefactos desarrollados durante el flujo de trabajo actual.	Alumno
20	Registrar en la Hoja de Esfuerzo el tiempo utilizado en el desarrollo del flujo de trabajo. Usar EPT-05.00.	Alumno
30	Revisar los artefactos entregados por el Alumno. Usar EPT-01.00	P. Guía
40	Evaluar la comprensión y el desempeño del Alumno en las diferentes actividades que debió realizar durante el flujo de trabajo. Usar EPT-01.00	P. Guía
50	Registrar en la Hoja de Esfuerzo el tiempo utilizado en la revisión de los artefactos. Usar EPT-06.00.	P. Guía
60	Archivar en Carpeta de Proyectos los Artefactos revisados y las respectivas Hojas de Esfuerzo.	Alumno

Fig. 6. Tercer Nivel de Documentación: Procedimientos Operativos

5 RESTRICCIONES

El Sistema de Aseguramiento de Calidad definido no pretende cumplir totalmente con CMM y RUP, sino que corresponde a una adaptación de estos últimos a la realidad de una organización (Departamento de Sistemas de Información) cuya misión no es construir software, sino la educación de profesionales que, entre otras cosas, participarán en el desarrollo de software.

Como se explica en la Sección 3, el sistema definido contempla cuatro actores (Alumno, Profesor Guía, Profesor Informante y Comisión de Control), los que deben realizar las actividades contempladas en CMM y RUP (que definen alrededor de 12 actores cada uno). Esta restricción de la estructura organizacional hace que no se pueda cumplir fielmente con la independencia entre ciertos actores como exigen CMM y RUP. Por ejemplo, las pruebas son definidas e implementadas por el alumno, que es quien realiza el diseño y la implementación de lo que se va a probar.

La Comisión de Control, que realiza a la Gerencia Superior definida en CMM (ver Figura 3), no está ubicada en la más posición gerencial del Departamento de Sistemas de Información como lo exige CMM.

La Administración de la Configuración es muy importante dentro del Desarrollo de Software. Sin embargo, en el sistema definido no es totalmente aplicable, ya que los desarrolladores son a lo más dos Alumnos, lo que implica que el manejo de versiones para este caso no es complejo. Por lo anterior, no se implementó toda esta Área Clave de Proceso, sino que sólo se rescataron aquellas actividades relacionadas con la identificación y control de artefactos.

El Área Clave de Proceso de CMM Administración de Subcontrato, no es aplicable en el curso Proyecto de Título, ya que el subcontrato en sí, no existe en él. Sin embargo, esto no es ningún impedimento para que a futuro, si surge la necesidad se agregue esta Área Clave de Proceso al Sistema de Aseguramiento de Calidad.

6 IMPLANTACION Y TRABAJO FUTURO

Durante el año 2001 se actualizarán los contenidos de las asignaturas de desarrollo de software, para preparar a los Alumnos y Profesores en conceptos y actividades relacionados con el Proceso de Desarrollo RUP y Sistema de Calidad definido para el curso Proyecto de Título.

Durante el primer semestre académico del 2001 se utilizará el Sistema de Aseguramiento de Calidad en un proyecto a cuyo término se realizará una revisión exhaustiva del sistema y se le realizarán las mejoras necesarias. En el segundo semestre académico del año 2001 se utilizará el sistema en al menos dos proyectos.

Desde el primer semestre académico del año 2002 se incrementará gradualmente el número de proyectos desarrollados bajo el Sistema de Aseguramiento de Calidad hasta llegar a una situación de equilibrio donde todos los proyectos, sin excepción, sean desarrollados bajo este esquema. (Semestralmente se presenta un promedio de 13 nuevos proyectos).

El Sistema de Aseguramiento de Calidad es un objeto dinámico que debe estar constantemente sometido a revisiones y mejoras. Entre estas mejoras está el pasar del Nivel 2 CMM (Repetible) al Nivel 3 CMM (Definido).

REFERENCIAS

1. Bagert, D.J. A model for the software engineering component of a computer science curriculum. *Information and Software Technology*. Vol. 40, No. 4, (April, 1998), pp. 191-193.
2. Instituto Nacional de Normalización (Chile). *NCh-ISO 9000/33.Of95: Normas de gestión de calidad y de aseguramiento de la calidad – Parte 3: Guías para la aplicación de Nch-ISO 9001 en el desarrollo, suministro y mantenimiento de software*. 1995.
3. Jacobson, I. Booch G. Rumbaugh J. *The Unified Software Development Process*. First Edition: Addison Wesley. 1999.
4. Jalote, P. *CMM in Practice: Processes for Executing Software Project at Infosys*. First Edition. Addison Wesley. 2000.
5. Ludewing J. Reibing R. Teaching what they need instead of teaching what we like – the new software engineering curriculum at the University of Stuttgart. *Information and Software Technology*. Vol 40, No 4, (April, 1998), pp. 239-244.
6. Schmauch, C.H. *ISO 9000 for software developers*. Second Edition, Milwaukee: Quality Press. 1995.
7. MacLean, G.E. *Documentación de Calidad para ISO 9000*. Primera Edición: McGraw-Hill. 1996.
8. Set Consultores. *CMM en Español I (traducción del modelo CMM 1.1.)* . <http://www.sets.cl>. (2000).
9. Tablada, G. *Guía para implementar la norma ISO 9000*. Primera Edición: McGraw-Hill. 1998.