



UNIVERSIDAD DE VALPARAÍSO  
Facultad de Ciencias  
Departamento de Computación  
Ingeniería Civil en Informática

## **DESARROLLO DE METODOLOGÍAS DE SQA Y SCM PARA LA DIRECCIÓN DE SERVICIOS DE INFORMACIÓN Y COMPUTACIÓN**

TRABAJO REALIZADO PARA OPTAR AL TITULO PROFESIONAL  
DE  
**INGENIERO CIVIL EN INFORMÁTICA**

**PALOMA MÉNDEZ SOTO**  
Profesor Guía: CARLOS BECERRA CASTRO  
Julio 2011

Certifico que he leído este documento y que, en mi opinión, es adecuado en ámbito y calidad como trabajo para optar al título de Ingeniero Civil en Informática.

---

CARLOS BECERRA CASTRO Profesor Guía

Certifico que he leído este documento y que, en mi opinión, es adecuado en ámbito y calidad como trabajo para optar al título de Ingeniero Civil en Informática.

---

ANDRES RAMOS Profesor Informante

Certifico que he leído este documento y que, en mi opinión, es adecuado en ámbito y calidad como trabajo para optar al título de Ingeniero Civil en Informática.

---

MARCO ARAVENA VIVAR Profesor Informante

Aprobado por el Departamento de Computación, UNIVERSIDAD DE VALPARAÍSO.

# **Resumen**

La Dirección de Servicios de Información y Computación (DISICO) fue creada para entregar servicios de calidad tanto a alumnos, profesores y funcionarios de la Universidad de Valparaíso, por lo cual constantemente se buscan formas de mejorar los procedimientos que se llevan a cabo para el desarrollo de cada proyecto. En la actualidad, DISICO carece de procesos que sigan el ciclo de vida de los proyectos provocando, pérdida de tiempo en identificar el estado de los recursos y ausencia de documentación formal de los cambios. El propósito del trabajo de título que se presenta en éste documento, es desarrollar metodologías para los procesos de SQA y SCM que garanticen mayor calidad y satisfacción para DISICO y sus clientes. Los principales resultados que se pretenden obtener al concretar la solución propuesta, son el desarrollo de las metodologías y su validación en un proyecto real, llevado a cabo en DISICO para tener certeza de la factibilidad de éstas, en términos de eficiencia y eficacia.

# **Agradecimientos**

Aqui pueden colocar sus agradecimientos. Si han estudiado con becas es recomendable colocar los agradecimientos a las instituciones que les otorgaron las becas.

# Índice general

<b>Resumen</b>	<b>III</b>
<b>Agradecimientos</b>	<b>IV</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Marco Conceptual</b>	<b>4</b>
2.1. Conceptos y Terminología para SQA Y SCM . . . . .	4
2.2. Estándares para SQA . . . . .	6
2.2.1. IEEE Plan de aseguramiento de calidad del software . . . . .	6
2.2.2. IEEE Verificación y Validación del software . . . . .	7
2.3. Métodos para SQA . . . . .	9
2.3.1. Auditorías para SQA de procesos y productos . . . . .	9
2.3.2. Prueba de Esfuerzo . . . . .	10
2.4. Herramientas para SQA . . . . .	11
2.4.1. OpenSTA . . . . .	11
2.4.2. Bugzilla . . . . .	12
2.4.3. Hudson . . . . .	12
2.4.4. Mantis . . . . .	13
2.5. Estándares para SCM . . . . .	14
2.5.1. IEEE Estándar para Plan de Gestión de Configuración de Software . . . . .	14
2.5.2. ISO Sistemas de Gestión de la calidad. Directrices para la Gestión de la Configuración . . . . .	15
2.6. Herramientas para SCM . . . . .	16
2.6.1. Sistema de versiones concurrentes (CVS) . . . . .	16
2.6.2. Subversion (SVN) . . . . .	17
2.6.3. ClearQuest . . . . .	17
2.7. Estándares y Métodos de Gestión de Calidad . . . . .	18
2.7.1. Familia ISO 9000 . . . . .	18
2.7.2. CMMI-DEV . . . . .	20

2.8.	Comparación entre trabajos . . . . .	22
2.8.1.	Comparación entre trabajos relacionados con SQA . . . . .	22
2.8.2.	Comparación entre trabajos relacionados con SCM . . . . .	24
2.8.3.	Comparación entre trabajos relacionados con Gestión de Calidad . .	26
2.8.4.	Análisis de los trabajos estudiados . . . . .	27
<b>3.</b>	<b>Definición del Problema</b>	<b>28</b>
3.1.	Definición del problema . . . . .	28
3.1.1.	Procesos Actuales . . . . .	28
3.1.2.	Testing . . . . .	28
3.1.3.	Control de Cambios y Versiones . . . . .	31
3.2.	Solución propuesta . . . . .	33
3.2.1.	Metodología SQA Propuesta . . . . .	33
3.2.2.	Metodología SCM Propuesta . . . . .	34
<b>4.</b>	<b>Diseño del Experimento</b>	<b>35</b>
4.1.	Selección del Contexto . . . . .	36
4.2.	Formulación Hipótesis . . . . .	37
4.2.1.	Hipótesis de Trabajo . . . . .	37
4.2.2.	Hipótesis Operacionales . . . . .	37
4.3.	Definición de Variables . . . . .	38
4.3.1.	Variables Independientes . . . . .	38
4.3.2.	Variables Dependientes . . . . .	39
4.4.	Selección Sujetos . . . . .	40
4.5.	Selección del Diseño del Experimento . . . . .	40
4.6.	Instrumentación . . . . .	41
4.6.1.	Instrumentos Metodología SQA . . . . .	41
4.6.2.	Instrumentos Metodología SCM . . . . .	42
4.6.3.	Instrumentos Metodología Ad-hoc . . . . .	42
4.7.	Fuentes de Invalidación . . . . .	43
4.7.1.	Fuentes de Invalidación Interna . . . . .	43
4.7.2.	Fuentes de Invalidación Externa . . . . .	43
4.8.	Ejecución del Caso de Estudio . . . . .	44
4.9.	Recolección de Datos . . . . .	45
4.10.	Ánalisis de Datos . . . . .	46
4.10.1.	Preparación y organización de Datos . . . . .	46
4.10.2.	Ánalisis y Conclusiones . . . . .	47

<b>5. Diseño Metodología SQA</b>	<b>48</b>
5.1. Tareas y Responsables . . . . .	49
5.2. Documentación . . . . .	50
5.2.1. Diseño y Especificación de Requerimientos de Software . . . . .	50
5.2.2. Documentación del Usuario . . . . .	51
5.2.3. Documento de Pruebas . . . . .	51
5.2.4. Verificación y Validación . . . . .	52
5.3. Recursos . . . . .	52
5.4. Formación . . . . .	53
<b>6. Diseño Metodología SCM</b>	<b>54</b>
6.1. Tareas y Responsables . . . . .	55
6.2. Actividades de SCM . . . . .	55
6.2.1. Identificación de Configuración . . . . .	56
6.2.2. Configuración de Control . . . . .	56
6.3. Recursos de SCM . . . . .	59
6.4. Formación . . . . .	60
<b>A. Encuesta Sujetos</b>	<b>61</b>
<b>B. Entrevista Sujetos</b>	<b>64</b>
<b>C. Capacitaciones</b>	<b>66</b>
C.1. Capacitación SQA . . . . .	66
C.2. Capacitación SCM . . . . .	77
<b>D. Template SQA</b>	<b>87</b>
D.1. Tareas y Responsables SQA . . . . .	87
D.2. Check-List . . . . .	90
D.3. Diseño y Especificaciones de Requerimientos . . . . .	91
D.4. Documento del Usuario . . . . .	99
D.5. Verificación y Validación . . . . .	102
D.6. Pruebas . . . . .	106
D.7. Formación . . . . .	112
<b>E. Template SCM</b>	<b>119</b>
E.1. Tareas y Responsables SCM . . . . .	120
E.2. Identificación de Configuración . . . . .	123
E.3. Configuración de Cambios . . . . .	125
E.3.1. Solicitud de Cambio . . . . .	126
E.3.2. Implementación del Cambio . . . . .	127



# Índice de cuadros

2.1.	Áreas de proceso y niveles de madurez CMMI-DEV asociados . . . . .	21
2.2.	Comparación entre Estándares de SQA . . . . .	22
2.3.	Comparación entre Métodos de SQA . . . . .	23
2.4.	Comparación entre Herramientas de SQA . . . . .	24
2.5.	Comparación de estándares de SCM . . . . .	25
2.6.	Comparación de Herramientas de SCM . . . . .	26
2.7.	Comparación de estándares de Gestión de Calidad . . . . .	26
2.8.	Comparación de Métodos y Modelos de Gestión de Calidad . . . . .	27
6.1.	Nomenclatura de urgencia de solicitudes de cambios . . . . .	58

# Índice de figuras

2.1.	Lista de Etapas propuesto en IEEE Std 730. . . . .	7
2.2.	Proceso propuesto en IEEE Std 1012. . . . .	9
2.3.	Página Oficial OpenSTA. . . . .	11
2.4.	Página Oficial Bugzilla. . . . .	12
2.5.	Página Oficial Hudson. . . . .	13
2.6.	Interfaz Login Mantiz. . . . .	14
2.7.	Proceso propuesto en IEEE Std 828. . . . .	15
2.8.	Proceso propuesto en ISO 10007. . . . .	16
2.9.	Respositorio de esquemas y esquemas . . . . .	18
3.1.	Proceso de Testing . . . . .	30
3.2.	Proceso de Control de Cambios y Versiones . . . . .	32
4.1.	Etapas Planificación Caso de Estudio . . . . .	36
4.2.	Etapas Ejecución Caso de Estudio . . . . .	45
5.1.	Etapas Metodología SQA . . . . .	48
6.1.	Etapas Metodología SCM . . . . .	54
6.2.	Proceso de Configuración de Cambio . . . . .	57

# **Capítulo 1**

## **Introducción**

La Dirección de Servicios de Información y Computación (DISICO) fue creada con la finalidad de llevar a cabo y controlar los objetivos informáticos de la Universidad de Valparaíso. En el decreto n°427 expedido por la Universidad de Valparaíso [20] se encuentra en detalle el listado de funciones de las cuales DISICO es responsable, siendo algunas de estas:

1. Administrar todo el procesamiento de datos y la documentación, que por medio de sistemas computacionales requiera la Universidad para su toma de decisiones.
2. Establecer un catastro renovable en el tiempo de los requerimientos informáticos de los usuarios de las distintas unidades de la Universidad.
3. Establecer pautas para obtener una estandarización en los sistemas computacionales.

Para cumplir con éstas funcionalidades es necesario considerar la satisfacción de aquellas personas que tienen acceso a éstos servicios, es decir el cliente. Una forma de lograr que el cliente entregue su total aprobación a un producto de software es asegurar la calidad de éste. En la actualidad existen buenas prácticas, estándares y procedimientos recomendados para lograr éste objetivo.

Uno de éstos procesos recomendados es el aseguramiento de calidad del software (SQA), el cual debe lograr minimizar los defectos en el producto entregado y para ésto es importante considerar tres puntos relevantes que menciona Roger S. Pressman [19]:

1. Los requisitos del software son la base de las medidas de la calidad. La falta de concordancia con los requisitos es una falta de calidad.
2. Los estándares especificados definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software. Si no se siguen esos criterios, casi siempre habrá falta de calidad.

3. Existen un conjunto de requisitos implícitos que a menudo no se mencionan (por ejemplo: el deseo por facilitar el uso y un buen mantenimiento). Si el software se ajusta a sus requisitos explícitos pero falla en alcanzar los requerimientos implícitos, la calidad del software queda en entredicho.

Por otra parte las especificaciones de cada proyecto se encuentran expuestas a reiterados cambios los cuales deben ser seguidos de forma constante, ya que de lo contrario se puede prestar para confusiones realizando cambios en versiones erróneas perdiendo tiempo de forma inútil que luego puede afectar en la planificación inicial. Para evitar esto surge la gestión de configuración de software (SCM), que se ha convertido un proceso muy importante al momento de construir sistemas complejos que se componen por un gran número de tareas paralelas expuestas a modificaciones, trayendo repercusiones impensables.

En la actualidad, DISICO carece de metodologías formales de SQA y SCM, que garanticen calidad y permitan gestionar de forma adecuada la planificación, los cambios y las versiones dentro de los proyectos que realizan. En consecuencia, la implementación de éstas metodologías traen consigo un gran número de mejoras para sus procesos.

El principal propósito de realizar éstas mejoras, es lograr un mayor control sobre el ciclo de vida del proyecto, gestionando de forma adecuada los recursos disponibles, monitoreando las actividades asignadas para que éstas cumplan con lo especificado en la planificación y entreguen un resultado exitoso para DISICO y el cliente. Una vez generadas las metodologías para SQA y SCM es importante realizar la validación de éstas, la cual se llevará a cabo en un proyecto real de DISICO extrayendo los datos necesario para hacer un análisis estadístico.

# **Capítulo 2**

## **Marco Conceptual**

### **2.1. Conceptos y Terminología para SQA Y SCM**

**El Aseguramiento de Calidad** o Software Assurance Quality (SQA) se encuentra definido como un patrón planificado y sistemático de todas las acciones necesarias para proporcionar la confianza adecuada de que un artículo o producto se ajusta a los requisitos técnicos [5]. Para lograr interiorizar el concepto de SQA es necesario definir un conjunto de conceptos y términos fundamentales.

- **Calidad:** En la Ingeniería de software la calidad es el grado en que un sistema, componente o proceso cumple con los requisitos especificados [5].
- **Atributos de Calidad:** Son un rasgo o característica que afecta la calidad de un artículo. También son nombrados como factores de calidad [5].
- **Análisis de Riesgo:** Una evaluación sistemática cualitativa o cuantitativa del software para los resultados deseados derivadas del desarrollo o el funcionamiento de un sistema. Métodos de detección o análisis son el eliminar, reducir o mitigar los riesgos [5].
- **Verificación y Validación:** Procesos para determinar si los requisitos para un sistema o componente se completan correctamente, los productos de cada fase de desarrollo cumplen con los requisitos o condiciones impuestos por la fase anterior, y el sistema final o componente cumple con los requisitos especificados [5].
- **Procesos del Ciclo de Vida:** Conjunto de actividades interrelacionadas que se traducen en el desarrollo o la evaluación del producto de software [5].
- **Requerimiento:** Una representación documentada de una condición o capacidad que necesita un usuario para resolver un problema o alcanzar un objetivo [5].

- Especificación de Requerimientos: Un documento que especifica los requisitos para un sistema o componente. Por lo general se incluyen los requisitos funcionales, requerimientos de rendimiento, requisitos de diseño y las normas de desarrollo [5].
- Métricas de Calidad: Una medida cuantitativa del grado en que un elemento tiene un atributo de calidad determinado [5].
- Auditoría: Un examen independiente de un producto de trabajo o conjunto de productos de trabajo para evaluar el cumplimiento de las especificaciones, normas, acuerdos contracturales, u otros criterios [5].

**La Gestión de Configuración de Software** o Software Configuration Management (SCM) es un proceso técnico y administrativo que busca identificar y documentar las características funcionales y físicas de un elemento de configuración, cambios de control a las características, registros y procesamiento de informes de cambio y estado de ejecución, y verificar el cumplimiento de los requisitos especificados [5]. Para lograr interiorizar el concepto de SCM es necesario definir un conjunto de conceptos y términos fundamentales.

- Item de Configuración: Una agregación de hardware, software, o ambos, que es designado para la gestión de configuración y tratados como una sola entidad en el proceso de gestión de la configuración [5].
- Baseline: Una especificación o producto que ha sido formalmente revisado y acordado, que después sirve de base para un mayor desarrollo, y que solo se puede cambiar a través de procedimientos formales de control de cambios [5].
- Comité de Control de Configuración: Un grupo de personas responsables de evaluar y aprobar o rechazar los cambios propuestos a los elementos de configuración, y para asegurar la aplicación de los cambios aprobados [5].
- Identificación de la Configuración: Es un elemento de la gestión de configuración, que consiste en seleccionar los elementos de configuración de un sistema y el registro de sus características funcionales y físicas en técnicas de documentación [5].
- Versión: Una versión inicial o re-lanzamiento de un elemento de configuración de software, asociadas con una compilación completa o la recopilación del elemento de configuración [5].
- Biblioteca de software: Una colección controlada de software y documentación relacionada diseñada para ayudar en el desarrollo de software, uso y mantenimiento. Los tipos incluyen la biblioteca principal, la biblioteca de producción y el repositorio de software [5].

- Proceso de control de cambios: Es un elemento de la gestión de configuración, que consiste en la evaluación, coordinación, aprobación o desaprobación, y la aplicación de los cambios en los elementos de configuración después del establecimiento formal de su identificación de configuración [5].

## 2.2. Estándares para SQA

En la actualidad existen una serie de estándares y normas para el aseguramiento de calidad. Éstas recomendaciones están dirigidas a cualquier organización independiente del rubro y tamaño de la empresa. Entregan directrices genéricas y fundamentales para alcanzar altos niveles de calidad, para lo cual deben ser adaptadas al lugar donde serán utilizadas.

Algunos de los estándares del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) se preocupan del aseguramiento de calidad específicamente del software, es decir que se enfocan directamente con el producto a desarrollar. Dos de los estándares más relevantes al momento de asegurar la calidad en el desarrollo del software se describen a continuación.

### 2.2.1. IEEE Plan de aseguramiento de calidad del software

El estándar IEEE Std. 730 [4] se enfoca principalmente en ayudar la preparación y el contenido de los planes de SQA. Se dirige hacia el desarrollo y mantenimiento crítico del software, es decir, que un fallo podría afectar a la seguridad o causar grandes pérdidas económicas o sociales. El principal objetivo de ésta norma es establecer requisitos mínimos para los planes de SQA.

En la Figura 2.1 se detallan cada una de las etapas propuestas en éste estándar para incorporar un proceso de SQA dentro de una organización y la descripción básica de ellas. Las etapas entregan recomendaciones para crear una planificación adecuada, pero no se entregan indicaciones de como implementar cada una de ellas, ya que todo depende de las características de la empresa.



Figura 2.1: Lista de Etapas propuesto en IEEE Std 730.

### 2.2.2. IEEE Verificación y Validación del software

El estándar IEEE Std. 1012 tiene como propósito principal establecer un marco común para los procesos de verificación y validación (V&V), actividades y tareas en apoyo de todos los procesos del ciclo de vida del software, incluyendo la adquisición, suministro, operación y procesos de mantenimiento, es decir se preocupa de definir el contenido del plan de V&V de software [6].

Los principales procesos que propone incorporar éste estándar se pueden observar en la Figura 2.2, la cual es descrita a continuación.

1. Gestión: Contiene las actividades genéricas y tareas, que pueden ser empleadas por cualquier parte que gestiona sus respectivos procesos. Las tareas de gestión son:
  - Preparar los planes para la ejecución del proceso.
  - Iniciar la ejecución del plan.
  - Vigilar la ejecución del plan.
  - Analizar los problemas descubiertos durante la ejecución del plan.
  - Informe sobre la marcha de procesos.
  - Asegurar que los productos cumplan necesidades.
  - Evaluar los resultados de la evaluación.
  - Determinar si una tarea se ha completado.
  - Comprobar los resultados completos.
2. Adquisición: Comienza con la definición de aquello que se desea adquirir. Le sigue la preparación y emisión de una solicitud de la propuesta de lo deseado, la selección de un proveedor y la gestión del proceso de adquisición a través de la aceptación del sistema, producto o servicio informático.
3. Suministro: Éste proceso es utilizado para verificar si la solicitud de los requisitos de la propuesta y los requisitos del contrato son consistentes y satisfacen las necesidades del usuario. La actividad de planificación utiliza los requisitos del contrato, incluidos los horario de programas de revisión y actualización de la planificación de la interfaz entre el proveedor y el comprador.
4. Desarrollo: Contiene las actividades y tareas del desarrollador. El proceso contiene las actividades de análisis de requerimientos, diseño, codificación, integración, pruebas, la instalación y aceptación relacionadas con los productos de software. La actividades son organizadas de acuerdo a los requisitos, diseño, implementación, prueba, instalación y envío de V&V.
5. Operación. Evalúa el impacto de los cambios en el entorno operativo, el efecto sobre el sistema de los cambios propuestos, los procedimientos operativos para el cumplimiento con el uso previsto, y los análisis de riesgos que afectan al usuario y el sistema.
6. Mantenimiento: Se activa cuando el producto de software sufre modificaciones en el código y la documentación asociada se debe a un problema o una necesidad de mejorar o adaptación.



Figura 2.2: Proceso propuesto en IEEE Std 1012.

Principalmente la V&V pretende asegurar no solo que se construya el producto correctamente sino también que sea el correcto. Esto se logra a través del descubrimiento de defectos en el sistema, y evaluando si el sistema es útil y utilizable. Si bien éste estándar ayuda a lograr mejores niveles de calidad no logra cubrir todas las áreas correspondientes al proceso de SQA [4].

## 2.3. Métodos para SQA

Actualmente existen una serie de métodos que intentan ayudar al aseguramiento de calidad, los cuales consisten en técnicas sencillas y al alcance de cualquier organización. Si bien cada uno de los métodos tienen un propósito en común, alcanzar la calidad, no lo logran por si solos, ya que no cubren todas las actividades que se necesitan y se consideran mínimas para obtener un buen resultado al momento de implementar un proceso de SQA. A pesar de ésto reuniendo varias de ellas se pueden lograr muy buenos resultados. A continuación se definen tres de los métodos más utilizados.

### 2.3.1. Auditorías para SQA de procesos y productos

Las auditorías para el aseguramiento de calidad, tanto para procesos como para productos, consiste en un examen sistemático e independiente con el fin de determinar si las actividades y los resultados relativos a la calidad se llevan de acuerdo según las normas pre establecidas, y si éstas se aplican de la forma correcta y escrita permitiendo alcanzar

los objetivos planteados al inicio de cada proyecto [8]. Las Auditorías de calidad se dividen en dos tipos, física y funcional.

- Auditoría Física: Tiene como objetivo principal corroborar si las especificaciones del diseño, producto y otras que se pueden haber generado para el desarrollo del software representan al software de forma fiel.
- Auditoría Funcional: Busca verificar que el desarrollo de los items de configuración existentes en la biblioteca de software cumplan con la especificación de requerimientos del software.

### 2.3.2. Prueba de Esfuerzo

El propósito principal de ésta prueba es medir el rendimiento de un sistema bajo diferentes escenarios posibles. Se busca exponer a situaciones extremas al sistema, asegurando que respondan de la forma que espera el usuario y en el caso no hacerlo, que entregue una solución adecuada al problema.

#### Revisión Técnica Formal

Actividad donde el software, su documentación y los procesos utilizados en su desarrollo son revisados por un grupo de personas. Se encargan de comprobar que se han seguido los estándares del proyecto y el software, y los documentos concuerdan con éstos estándares [3]. Existen diferentes tipos de revisiones:

- Inspección de diseño o programas: Detectar errores finos en los requerimientos, el diseño o el código.
- Revisiones del progreso: Proveer información útil sobre el progreso del proyecto, que beneficie la gestión de éste.
- Revisiones de la calidad: Llevar a cabo un análisis técnico de los componentes del producto o documentación para encontrar diferencias entre la especificación, y para asegurar que se siguen los estándares de calidad definidos [3].

## 2.4. Herramientas para SQA

Si bien no existe una herramientas para SQA que automatice éste proceso de forma completa, están disponible herramientas que benefician el desarrollo de proyectos de software dentro de una organización. En éste caso se describirán cuatro herramientas cuyo enfoque principal es encontrar los errores que pueden arrastrarse durante el ciclo de vida

del proyecto, simulan situaciones específicas realizando un seguimiento al comportamiento de la aplicación.

### 2.4.1. OpenSTA

Es una herramienta open-source que permite realizar testing de rendimiento especialmente en aplicaciones web. El conjunto de herramientas tiene la capacidad de ejecutar secuencias de comandos que proporcionan pruebas de carga y stress para de ésta forma cuantificar el rendimiento que tiene la aplicación en situaciones extremas [17]. OpenSTA se convierte en un buen aliado para aplicaciones web que necesitan un mantenimiento diario, ya que entrega la funcionalidad de programar estas pruebas de comportamiento al sistema, obteniendo un monitoreo constante.

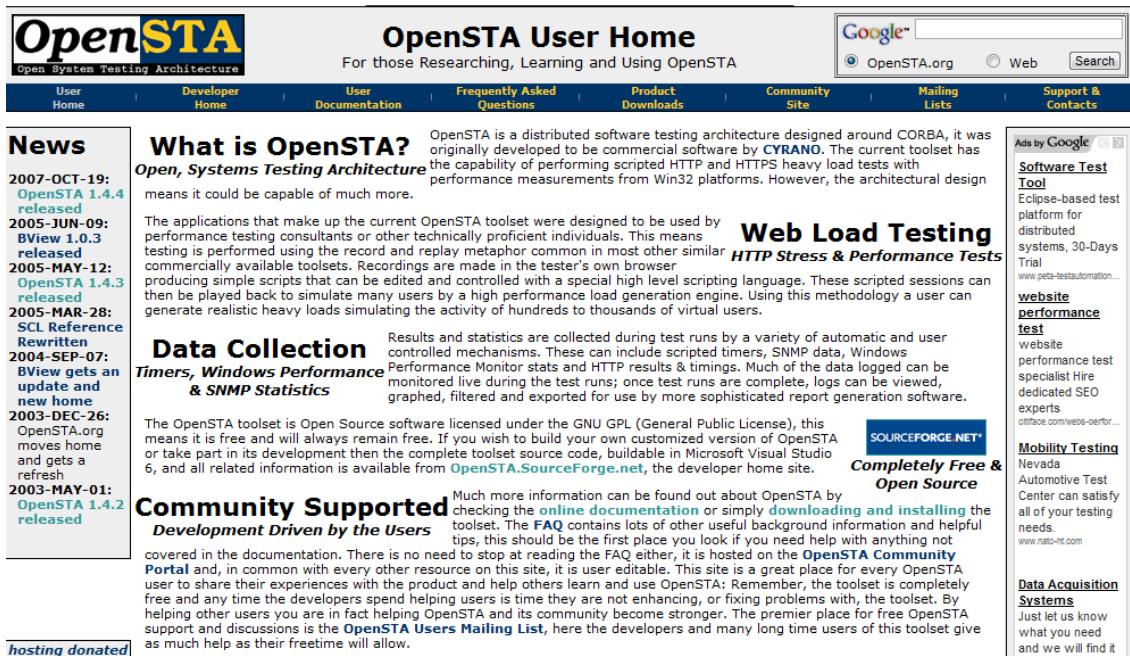


Figura 2.3: Página Oficial OpenSTA.

### 2.4.2. Bugzilla

Ésta herramienta es un sistema de seguimiento y notificación de errores. Está diseñado de tal forma que pueda ser aplicado a casi cualquier desarrollo de software. Bugzilla permite realizar un seguimiento de errores y cambios en el código, mantener comunicación entre los integrantes del equipo, enviar y revisar las soluciones y administrar la garantía de calidad [16].



The screenshot shows the Bugzilla homepage. At the top, there's a dark blue header bar with the Bugzilla logo (a cartoon ant) and the word "Bugzilla" in white. To the right of the logo is a search bar with the placeholder "search bugzilla.org:" and a "go" button. Below the header are several navigation links: About, News, Docs, Support, Download, Features, and Contribute!. The main content area has a light gray background. It starts with a section titled "Bugzilla" which describes it as server software for managing software development. There are links to "More about Bugzilla" and "News". The "News" section lists three recent releases: "Release of Bugzilla 4.1.1 [ 2011 Mar 13 ]", "Release of Bugzilla 4.0! [ 2011 Feb 15 ]", and "Release of Bugzilla 3.2.10, 3.4.10, 3.6.4, and 4.0rc2 [ 2011 Jan 24 ]". Below the news is a link to "More news". The "Features" section contains a bulleted list of benefits: optimized database structure, excellent security, advanced query tool, integrated email capabilities, editable user profiles, comprehensive email preferences, a permissions system, and being proven under fire as Mozilla's bug tracking system. There's also a link to "View complete list of features". The "Benefits" section has one bullet point: "Improve communication". On the right side of the page, there's a small image of the Bugzilla logo ant and the text "Bugzilla logo by Dave Shea".

Figura 2.4: Página Oficial Bugzilla.

### 2.4.3. Hudson

Hudson es una herramienta web que permite realizar un seguimiento del código guardado en un repositorio como SVN o CVS. Entrega constantes reportes de los errores en el código según las actualizaciones que sufra éste. Es un herramienta fácil de utilizar y muy intuitiva [18]. En la actualidad es utilizada en el Área de desarrollo de DISICO en conjunto con SVN.

The screenshot shows the Hudson official website. At the top, there is a dark blue header bar with the word "HUDSON" in large white letters and "Extensible continuous integration server" in smaller white text below it. To the right of the logo are navigation links: HOME, MAILING LISTS, BUGS, WIKI, SOURCE, and DOCUMENTATION. Below the header, there are three main sections: "Meet Hudson" (with a brief description and a cartoon character icon), "Use Hudson" (with a brief description and a book icon), and "Extend Hudson" (with a brief description and a wrench and screwdriver icon). To the right of these sections is a sidebar titled "News" which contains three items: "Hudson 2.0.0 Is Available" (with a link to download), "Revision of Hudson Versioning Scheme" (with a link for more info), and "Hudson Process Documents Published" (with a link to the documents page). At the bottom of the page, there is a footer bar with an RSS feed icon and links to Twitter Feed and JIRA Updates, along with a small "Duke's Choice AWARD" badge.

Figura 2.5: Página Oficial Hudson.

#### 2.4.4. Mantis

Mantis es una herramienta de gestión de incidencias informáticas muy flexible y eficaz, también tiene una interfaz de fácil comprensión para los usuarios. Esta herramienta permite al administrador asignar incidencias y monitorear el estado de éstas [13]. En la actualidad Mantis es usada en el Área de Desarrollo de DISICO para gestionar los errores que se generan luego de que el responsable del aseguramiento de calidad (QA) realice las pruebas unitarias.



Login		<a href="#">[ Login Anonymously ]</a>
<b>Username</b>	<input type="text"/>	
<b>Password</b>	<input type="password"/>	
<b>Remember my login in this browser</b>	<input type="checkbox"/>	
<b>Secure Session</b>	<input checked="" type="checkbox"/> Only allow your session to be used from this IP address.	
<input type="button" value="Login"/>		
<a href="#">[ Signup for a new account ]</a>   <a href="#">[ Lost your password? ]</a>		

Figura 2.6: Interfaz Login Mantiz.

## 2.5. Estándares para SCM

El proceso de SCM contiene un número significativo de actividades las cuales necesitan un planificación adecuada para asegurar su cumplimiento. Si bien es un tema bastante nuevo ya se han descrito ciertas normas con requisitos mínimos para implementar éste tipo de procedimientos. En ésta sección se detallan las características primordiales de dos de los estándares utilizados actualmente por algunas organizaciones.

### 2.5.1. IEEE Estándar para Plan de Gestión de Configuración de Software

El IEEE propone el Std. 828 para el plan de gestión de configuración de software que permite la planificación de las actividades que comprende la gestión de configuración de software, entregando los requisitos mínimos para la implementación de un proceso de SCM en el desarrollo de software [7]. La Figura 2.7 representa las cinco secciones fundamentales en las cuales se divide el estándar y se define la planificación propuesta.



Figura 2.7: Proceso propuesto en IEEE Std 828.

### 2.5.2. ISO Sistemas de Gestión de la calidad. Directrices para la Gestión de la Configuración

La ISO 10007 tiene como propósito principal mejorar la comprensión sobre el tema de gestión de la configuración, promover su uso y ayudar a las organizaciones que la aplican, a mejorar su desempeño. En particular ésta norma entrega directrices para el uso del proceso de SCM dentro de una organización. Resume responsabilidades de cada integrante de un proyecto, la planificación para SCM, la identificación de configuración, el control de cambios y las auditorías de configuración[9]. La Figura 2.8 representa el proceso de gestión de calidad propuesto por éste estándar.



Figura 2.8: Proceso propuesto en ISO 10007.

## 2.6. Herramientas para SCM

La herramientas usadas en la actualidad para procesos como el de SCM se enfocan en el control de cambio y versiones. Existen varias herramientas que colaboran para éste tipo actividades, las que en su mayoría logran su objetivo a través de un repositorio. La diferencia entre ellas es la administración del repositorio y su estructura principal. A continuación se describen tres herramientas para respaldar el proceso de SCM.

### 2.6.1. Sistema de versiones concurrentes (CVS)

Es una aplicación informática que implementa un sistema de control de versiones, mantiene el registro de todo el trabajo y los cambios en los ficheros que forman un proyecto y permite que distintos desarrolladores colaboren. Es una herramienta open-source que permite la concurrencia del trabajo, donde más de una persona puede trabajar con el mismo archivo [2]. Una desventaja de CVS, es que los archivos que existen en el repositorio no puede ser renombrados, deben agregarse con otro nombre y luego borrar el anterior.

### 2.6.2. Subversion (SVN)

Subversion es un sistema de control de versiones creado con la finalidad de reemplazar a CVS. SVN permite tener acceso al repositorio a través de redes, permitiendo a los

integrantes de un proyecto tener acceso a él desde distintos computadores sin problema, permitiendo las modificaciones de forma paralela entre los integrantes fomentando el trabajo en equipo [1].

Para el control de versiones Subversion recomienda la siguiente organización:

- Branches: Línea principal de desarrollo.
- Tags: Alberga ramas alternativas de desarrollo, las cuales generalmente se utilizan para realizar testing.
- Trunk: Directorio con versiones finales etiquetadas.

### 2.6.3. ClearQuest

Consiste en una serie de componentes que trabajan en conjunto. Las herramientas administrativas que componen ésta herramienta son las siguientes [14]:

- Herramienta de Diseño
- Herramienta de Mantenimiento
- Herramienta de lectura de e-mail
- Herramienta de administración de usuarios
- Herramienta para importar
- Herramienta para exportar
- Herramienta multisitios

ClearQuest trabaja con una variedad de objetos los cuales están definidos como base de datos, esquemas, repositorios de esquemas, conjuntos de bases de datos y conexiones. Ésta herramienta puede ser utilizada para diseñar y modificar esquemas en su respectivo repositorio. Para la creación y edición de un esquema incluye actividades como definir una distribución de datos, construir un modelo de proceso para cada tipo de petición de cambios, añadir programación para los modelos de procesos, crear formularios, tablas e informes, entre otras tareas permitidas por ClearQuest. La Figura 2.9 muestra la estructura del repositorio propuesto.

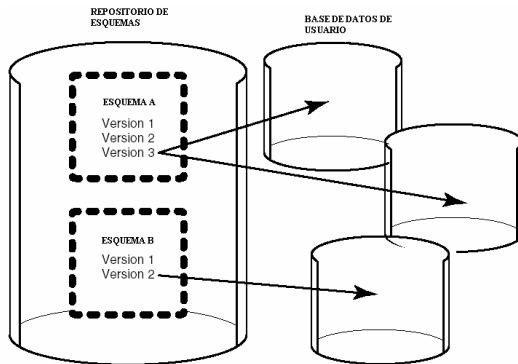


Figura 2.9: Respositorio de esquemas y esquemas

## 2.7. Estándares y Métodos de Gestión de Calidad

Así como podemos encontrar normas y prácticas específicas del proceso de SQA y SCM, también se encuentran disponibles estándares que abarcan por completo la gestión de calidad, siendo éste el concepto de donde nacen ambos procesos propuestos en éste trabajo de título. A continuación se observan una serie de estándares y herramientas con éste enfoque.

### 2.7.1. Familia ISO 9000

La Norma ISO 9000 [10] es una familia de normas preocupadas de normalizar la gestión de calidad para diferentes tipos de empresas, la cual puede ser utilizada para el desarrollo de software. Están enfocadas a cualquier tipo de organización, independiente de su rubro y tamaño.

Se identifican ocho principios de gestión de calidad fundamentales, que forman la base de la familia de Normas ISO 9000 de gestión de calidad [10]:

1. Enfoque al cliente
2. Liderazgo
3. Participación del personal
4. Enfoque basado en procesos
5. Enfoque de sistema para la gestión

6. Mejora continua
7. Enfoque basado en hechos para la toma de decisión
8. Relaciones mutuamente beneficiosas con el proveedor

La Norma ISO 9000 describe cada integrante de ésta familia, listados a continuación:

- La Norma ISO 9000, Sistemas de Gestión de Calidad - Fundamentos y Vocabulario, describe los fundamentos de los sistemas de gestión de la calidad y especifica la terminología para los sistemas de gestión de calidad [10]. La estructura de ésta norma es la siguiente:
  - Objeto y Campo de aplicación: Específica en qué tipo de organizaciones es aplicable ésta norma. Principalmente está dirigido a empresas que buscan la confianza de que los requisitos de los productos serán satisfechos.
  - Fundamentos de los sistemas de gestión de la calidad: Describe la base, los requisitos y el enfoque necesario para un sistema de gestión de calidad.
  - Términos y Definiciones: Define un conjunto de términos pertenecientes a la gestión de calidad, tanto para el proceso de SQA como de SCM.
- La Norma ISO 9001, Sistemas de Gestión de Calidad - Requisitos, especifica los requisitos para los sistemas de gestión de la calidad aplicables a toda organización que necesite demostrar su capacidad para proporcionar productos que cumplan los requisitos de sus clientes y los reglamentarios. Su objetivo es aumentar la satisfacción del cliente [11]. Los nuevos elementos definidos en ésta norma son:
  - Sistemas de gestión de calidad
  - Responsabilidad de la dirección
  - Gestión de los recursos
  - Realización del producto
  - Medición, análisis y mejora

Éstos puntos especifican los requisitos mínimos para lograr implementar un sistema de gestión de calidad dentro de una organización.

- La Norma ISO 9004, Sistemas de Gestión de Calidad - Directrices para la mejora del desempeño, proporciona directrices que consideran tanto la eficacia como la eficiencia del sistema de gestión de la calidad. El objetivo de esta norma es la mejora del desempeño de la organización y la satisfacción de los clientes y de otras partes interesadas [12]. Si bien ésta norma se estructura de la misma forma que la Norma ISO 9001, se caracteriza por perseguir la continua mejora del desempeño.

### 2.7.2. CMMI-DEV

Es un modelo de madurez que mejora los procesos del desarrollo de productos y de servicios. Propone una solución integrada y completa para mejorar las actividades de desarrollo y de mantenimiento. Existen seis niveles de madurez dentro de éste modelo, donde cada uno refleja un conjunto de metas y prácticas que pueden estar presentes en una organización. Dependiendo del conjunto utilizado por la empresa es su nivel de madurez [15]. Estos niveles son:

1. Nivel 1, Inicial: Se caracteriza por contener procesos incompletos o inexistentes provocando constantes problemas dentro de la organización, como productos defectuosos, exceso en los presupuestos e incumplimiento de los tiempos establecidos.
2. Nivel 2, Gestionado: Se realizan los procesos necesarios para lograr el éxito en cada proyecto, existe la planificación de éstos de acuerdo a las políticas de la organización. Si bien logran los objetivos mínimos no manejan una documentación de los procesos para la obtención de calidad.
3. Nivel 3, Definido: Los procesos correspondientes a la organización se encuentran claramente definidos en procedimientos y métodos en la documentación correspondiente, provocando constancia en las buenas prácticas necesarias para el éxito en un proyecto de software.
4. Nivel 4, Gestionado Cuantitativamente: La organización y los proyectos establecen objetivos cuantitativos en cuanto al rendimiento de calidad del proceso, y los utilizan como criterios en la gestión de software.
5. Nivel 5, Optimizado: Cuando los procesos que ya fueron definidos claramente y a su vez mantienen objetivos cuantitativos, se encuentran en constante mejora.

Cada uno de éstos niveles tiene un conjunto de actividades que lo caracteriza, si bien el modelo CMMI-DEV permite evaluar el nivel de madurez de una organización, también le permite alcanzar uno de éstos niveles. En el Cuadro 2.1 se describen las actividades para los niveles de madurez 2, 3, 4 y 5. Se debe recordar que el nivel 1 no contiene procesos definidos.

Área de proceso	Nivel de Madurez
Gestión de Configuración	2
Medición y Análisis	2
Monitorización y control de proyecto	2
Planificación de proyecto	2
Aseguramiento de la calidad de proceso y producto	2
Gestión de requerimientos	2
Gestión de acuerdos con proveedores	2
Análisis de decisiones y resolución	3
Gestión integrada de proyecto	3
Definición de procesos de la organización	3
Enfoque en procesos de la organización	3
Formación Organizativa	3
Integración de producto	3
Desarrollo de requerimientos	3
Gestión de riesgos	3
Solución técnica	3
Validación	3
Verificación	3
Rendimiento de procesos de la organización	4
Gestión cuantitativa de la organización	4
Ánálisis causal y resolución	5
Innovación y despliegue de la organización	5

Cuadro 2.1: Áreas de proceso y niveles de madurez CMMI-DEV asociados

## 2.8. Comparación entre trabajos

Luego del estudio de cada una de éstas prácticas y técnicas para los proceso de SQA y SCM es importante realizar un análisis de sus ventajas y desventajas, y decidir si satisfacen o no las necesidades de DISICO. Se puede observar que ninguna de ellas se ajusta a la realidad y recursos existentes en el Área de Desarrollo. Si bien en su mayoría, estas normas y herramientas definidas entregan flexibilidad para su uso en diferentes tipos de organizaciones, no contienen las facilidades para utilizarlas con éxito con un número reducido de participantes y tiempo.

### 2.8.1. Comparación entre trabajos relacionados con SQA

A pesar de que las normas y técnicas descritas para concretar el proceso de SQA no satisfacen las necesidades de DISICO, el estándar IEE Std. 730 entrega las directrices principales para la planificación de cada procedimiento, permitiendo adaptar los pasos principales y requeridos a los recursos actuales. Logran ser lineamientos de gran utilidad para un primer acercamiento al proceso de SQA. Si bien el estándar IEEE 1012 ayuda a asegurar la calidad y propone una metodología mucho mas breve que el estándar anterior, no cubre todas las actividades necesarias para una implementación correcta de un proceso de SQA (Ver Cuadro 2.2).

Estándares SQA	Ventajas	Desventajas
IEEE Plan de aseguramiento de calidad del software	<p>Involucra a cada integrante de la organización en el proceso de SQA.</p> <p>Busca establecer los responsables, etapas, técnicas y documentación para el aseguramiento de la calidad del software.</p> <p>Ayuda a alcanzar mayor calidad en el producto de software.</p> <p>Permite entregar mayor satisfacción al cliente.</p>	<p>Requiere una gran cantidad de documentación.</p> <p>Exige un mínimo de tres integrantes para su implementación.</p>
IEEE Verificación y Validación del software	<p>Se preocupa que los requisitos sean técnicamente correctos.</p> <p>Corrobora la satisfacción de las necesidades del sistema.</p>	<p>No logra cubrir todas la actividades necesarias para la correcta implementación de un proceso de SQA [4].</p>

Cuadro 2.2: Comparación entre Estándares de SQA

Los métodos planteados son de fácil implementación, pero ninguno de ellos es factible ya que para alcanzar los objetivos de un proceso de SQA es necesario utilizarlos de forma conjunta incluyendo otros tipos de técnicas, lo que no se adapta a los recursos disponibles en DISICO (Ver Cuadro 2.3).

Métodos SQA	Ventajas	Desventajas
Auditorías para SQA de procesos y productos.	Permite una organización y coordinación interna dentro de la empresa  Entrega mayor satisfacción al usuario.  Disminuye los riesgos.	No cubre por completo las tareas mínimas para la implementación de un procesos de SQA [4].
Prueba de Esfuerzo	Pueden realizarse con el apoyo de alguna herramienta que automatice éste tipo de prueba.	No cubre por completo las tareas mínimas para la implementación de un proceso de SQA [4].
Revisión Técnica Formal	Somete al producto a una revisión muy profunda realizada de forma manual.	Requiere entre dos y cuatro personas.  No cubre por completo las tareas mínimas para la implementación de un proceso de SQA [4]. Requiere varios procedimientos gastando tiempo extra para su implementación.

Cuadro 2.3: Comparación entre Métodos de SQA

Por otra parte, en el caso de las herramientas OpenSTA y Bugzilla, a pesar de entregar información sobre los errores que existen en las aplicaciones web y ayudan a aumentar la calidad de los productos, no es factible para DISICO usar tiempo y recursos extras para aprender a utilizar nuevas herramientas. Las herramientas Hudson y Mantis son usadas en el Área de Desarrollo de DISICO con bastante éxito, por lo cual serán usadas para potenciar la metodología de SQA (Ver Cuadro 2.4).

Herramientas para SQA	Ventajas	Desventajas
OpenSTA	<p>Software Libre.</p> <p>Permite trabajar en aplicaciones web.</p> <p>Entrega gran cantidad de información sobre los errores encontrados.</p> <p>Tiene su propio lenguaje de script.</p>	No cubre por completo las tareas mínimas para la implementación de un proceso de SQA [4].
Bugzilla	<p>Software Libre.</p> <p>Funciona en diferentes plataformas.</p> <p>Control de errores para software y hardware</p>	<p>No permite guardar seguimientos y tareas realizadas.</p> <p>No permite ver disponibilidad de los recursos humanos.</p> <p>No controla tiempo en cada actividad que fue realizada.</p> <p>No cubre por completo las tareas mínimas para la implementación de un proceso de SQA [4].</p>
Hudson	<p>Es simple e intuitivo se usar.</p> <p>Permite trabajar en conjunto con herramientas como SVN y CVS.</p>	No cubre por completo las tareas mínimas para la implementación de un proceso de SQA [4].
Mantis	<p>Software Libre.</p> <p>Es de fácil aprendizaje.</p> <p>Permite gestionar las incidencias informáticas.</p>	No cubre por completo las tareas mínimas para la implementación de un proceso de SQA [4].

Cuadro 2.4: Comparación entre Herramientas de SQA

### 2.8.2. Comparación entre trabajos relacionados con SCM

Para los estándares planteados en la sección 2.5, para el proceso de SCM es necesario un gran número de personas, ya que ambos proponen conformar diferentes equipos de trabajos para tareas específicas dentro de la planificación, lo que no es posible incorporar en DISICO, porque exige recursos que no se tienen en la actualidad.

A pesar de lo alejado que se encuentra del requisito principal del Área de Desarrollo a la hora de implementar una nueva metodología, el estándar IEEE Std. 828 entrega las directrices principales para una planificación de un proceso de SCM con una base sólida para la metodología que se desea crear. Por otra parte la ISO 10007, aunque entrega lineamientos claros y completos sobre un proceso de SCM, se enfoca mayoritariamente en entregar orientación sobre el término SCM, más que en una planificación concreta del proceso (Ver Cuadro 2.5).

Estándares SCM	Ventajas	Desventajas
IEEE Plan de Gestión de Configuración de Software	Comprime todas las actividades de un procesos SCM en cinco etapas. Entrega una base sólida para realizar una buena planificación de SCM.	Requiere un mínimo de cinco personas para implementar el proceso propuesto sin dificultad y con éxito.
ISO Directrices para la Gestión de Configuración de Software	Entrega orientación sobre el término SCM.  Promueve el uso del proceso de SCM en las organizaciones. Ayuda a mejorar el proceso de SCM en organizaciones que ya lo implementen.	Exige un mínimo de cinco personas para su implementación.  Indica la necesidad de documentar un gran número de procesos.

Cuadro 2.5: Comparación de estándares de SCM

La herramienta SVN es usada actualmente en el Área de Desarrollo de DISICO, aunque es un paso muy importante para el control de versiones dentro del desarrollo de software, no han logrado usar por completo las capacidades que ésta tiene, ya que se utiliza solo el directorio Branches descrito en la sección 2.6.2. El que SVN sea una herramienta ya incorporada dentro del proceso de desarrollo es una ventaja sobre el resto de las tecnologías planteadas, siendo seleccionada como la herramienta adecuada para el control de versiones y descartando CVS y ClearQuest por motivos prácticos, como el tiempo y recursos que significa aprender a utilizar una nueva tecnología (Ver Cuadro 2.6).

Herramientas SCM	Ventajas	Desventajas
Sistema de versiones concurrentes	Se pueden comparar diferentes versiones.  Permite sacar varias copias de un proyecto al mismo tiempo.	Los archivos en el repositorio no pueden ser renombrados.
Subversion	Sigue la historia de los archivos y directorios por medio de copias y renombrados.  La creación de ramas y etiquetas es más eficiente que en CVS.	El manejo de cambio de nombre de archivos no es completo.  No facilita el conteo de cambios realizados.
ClearQuest	Permite gestionar varias tareas al mismo tiempo.  Crea formulario, tablas e informes.  Crea una base de datos test para examinar el esquema guardado antes de chequearlo.	Es una herramienta compleja con muchas funcionalidades, necesitando demasiado tiempo para su comprensión.  Entrega funcionalidades que no son necesarias para la implementación de un proceso de SCM.

Cuadro 2.6: Comparación de Herramientas de SCM

### 2.8.3. Comparación entre trabajos relacionados con Gestión de Calidad

Los estándares y modelos que se preocupan de la gestión de calidad no cumplen con los requerimiento de DISICO para éste trabajo de título. Es necesario generar metodologías independientes para cada proceso, ya que las necesidades actuales se enfocan con más fuerza sobre el proceso de SCM. También ésta independencia nos permite que las metodologías propuestas sean lo más modular posible, considerando la posibilidad de incorporarlas en las otras áreas de trabajo de DISICO (Ver Cuadro 2.7 y Cuadro 2.8).

Estándar	Ventajas	Desventajas
Familia ISO 9000	Entrega mayor nivel de satisfacción al cliente.  Reducción de Costos.  Mejor comunicación dentro del grupo de trabajo.	Combina procesos tanto de SQA y SCM en una misma norma y metodología.  Se necesita un grupo mínimo de cinco personas para ser implementada.

Cuadro 2.7: Comparación de estándares de Gestión de Calidad

Métodos y Modelos	Ventajas	Desventajas
CMMI-DEV	<p>Permite conocer el nivel de madurez de la organización.</p> <p>Sirve de base para cualquier organización que busca mejorar.</p>	<p>No es un Proceso.</p> <p>No es una metodología de desarrollo de software ni de gestión.</p> <p>Mezcla en un mismo modelo actividades de SQA Y SCM.</p>

Cuadro 2.8: Comparación de Métodos y Modelos de Gestión de Calidad

#### 2.8.4. Análisis de los trabajos estudiados

Luego de realizar el estudio de prácticas y herramientas que existen en la actualidad y que son usadas por algunas organizaciones para procesos de SQA y SCM, se han seleccionado aquellas que serán de ayuda para crear las metodologías propuestas en éste trabajo de título.

Para la metodología de SQA se usará como base principal Std. 730, el cual entrega una planificación completa para SQA, permitiendo extraer aquellas actividades que se ajustan a las necesidades de DISICO. También las herramientas Hudson y Mantis, las cuales ya se encuentran incorporadas en el Área de Desarrollo, lo que hace innecesario el aprender el uso de nuevas herramientas.

Para la metodología de SCM se utilizará el Std. 828 que entrega las directrices principales y fundamental para una planificación adecuada de un proceso de SCM. Éste estándar permitirá tener una base sólida para crear una nueva metodología de SCM. Junto con Std. 828, se empleará la herramienta SVN, la cual ya es utilizada con éxito en el Área de Desarrollo.

# **Capítulo 3**

## **Definición del Problema**

### **3.1. Definición del problema**

El problema principal es la inexistencia de metodologías para los procesos de SQA y SCM. La ausencia de éstos procesos trae consigo una serie de consecuencias al momento de realizar un proyecto de desarrollo en DISICO. Para identificar con claridad cada uno de los problemas que existen en DISICO en particular en el Área de Desarrollo, se definen a continuación los procesos utilizados en la actualidad.

#### **3.1.1. Procesos Actuales**

Para lograr un mejor análisis del problema es necesario definir algunos de los procesos que actualmente se encuentran vigentes en el Área de Desarrollo de DISICO y que son influyentes dentro de una metodología de SQA y SCM. Éste procedimiento permitirá identificar las falencias en cada uno de los procesos, obteniendo la información suficiente para generar una solución factible al problema. En los puntos siguientes se describen dos de los procedimientos claves que se relacionan con los procesos de SQA y SCM, los cuales varían según la urgencia de cada proyecto.

#### **3.1.2. Testing**

Con el fin de asegurar la calidad de los productos de software desarrollados en el Área de Desarrollo de DISICO, se realiza un procedimiento de testing liderado por el responsable de QA. Si bien existe el modelo formal de un proceso de QA que involucra pruebas unitarias, de integración y sistemas, y por otra parte la aprobación del operador de base de datos (DBO) del modelo de datos, no se aplican actualmente por factores como el tiempo y falta de recursos. En la mayoría de los casos solo se logran realizar pruebas unitarias.

A continuación se describen los pasos del proceso de testing actual, que está representado gráficamente en la Figura 3.1.

- El Desarrollador envía un documento de requerimientos al responsable de QA. Éste documento incluye los requerimientos correspondiente al módulo o sistema a evaluar, los casos de uso correspondiente y prototipos de la interfaz.
- El responsable de QA revisa el documento enviado por el Desarrollador. En el caso de existir imperfecciones en éste informe notifica al Desarrollador para que realice las correcciones pertinentes. Éste paso se realiza hasta que el responsable de QA apruebe el documento.
- Una vez aprobado el documento de requerimiento por el responsable de QA, procede a generar el documento de casos de pruebas.
- Apoyado en el documento de casos de pruebas unitarias, el responsable de QA procede a realizar las pruebas unitarias. Los errores encontrados son registrados y enviados al desarrollador para que los corrija. Éste paso se lleva a cabo hasta que el responsable de QA considere que el módulo o sistema creado por el desarrollador cumple con le documento de requerimientos correctamente.
- Luego de aprobar el módulo o sistema por el responsable de QA, el producto puede ser liberado con mutua acuerdo del Jefe del Área de Desarrollo de DISICO.

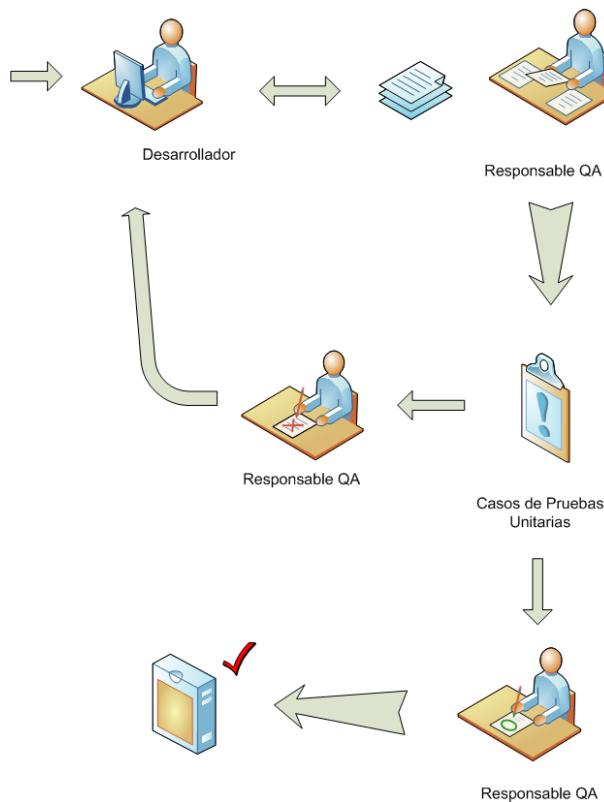


Figura 3.1: Proceso de Testing

Si bien éste proceso ayuda a asegurar cierta calidad en los productos desarrollados, tiene muchos problemas ya que no cubre el ciclo de vida del software por completo. El siguiente listado detalla las dificultades que existen en el proceso actual.

- En ocasiones los sistemas desarrollados no pasan por éste proceso debido a los problemas de estimación y la urgencia del cliente.
- El documento de requerimiento no exige la especificación de éstos.
- El responsable de QA valida los módulos de al menos ocho desarrolladores no teniendo el tiempo suficiente para pasar de las pruebas unitarias.
- Al no existir un monitoreo durante el ciclo de vida del proyecto, en ocasiones se pasan por alto ciertos requerimientos, los cuales se agregan con posterioridad cuando el responsable de QA u otro da cuenta de esto.

- El responsable de QA no participa de las reuniones de toma de requerimientos con los clientes, perdiendo la oportunidad de comprender la visión del cliente y lo que espera del producto.

### 3.1.3. Control de Cambios y Versiones

Para tener control sobre los cambios y las versiones de los sistemas desarrollados no existe un proceso formal pero se intenta llevar a cabo ciertos procedimientos. El flujo de tareas que se realiza en la actualidad se explica a continuación y se representa gráficamente en la Figura 3.2.

- Se envía un solicitud de cambio por correo electrónico o por teléfono al Jefe del Área de Desarrollo de DISICO.
- El jefe del Área de Desarrollo de DISICO aprueba el cambio solicitado e informa al desarrollador involucrado.
- El desarrollador realiza la actualización en el módulo o sistema y guarda la nueva versión en el directorio Branches de SVN.
- El jefe de proyecto busca errores en el nuevo código apoyándose en una herramienta automatizada que le notifica las fallas existentes en las líneas de códigos.
- Si el Jefe de Proyecto encuentra errores informa al Desarrollador involucrado para que realice las correcciones pertinentes. Éste paso se realizada hasta que el Jefe de Proyecto considere que el código guardado en el repositorio SVN está correcto.

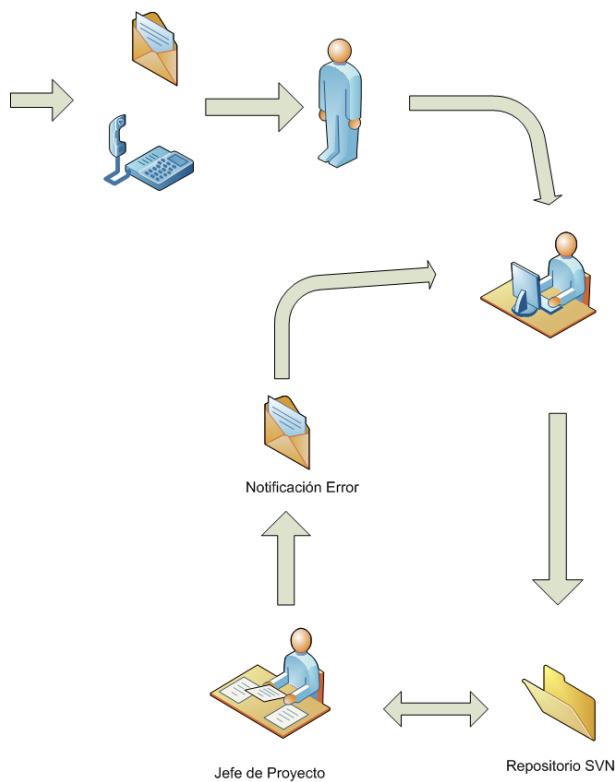


Figura 3.2: Proceso de Control de Cambios y Versiones

Los principales problemas que existen en éste proceso se detallan a continuación:

- No existe un proceso formal. La actividades descritas en la Figura 3.2 varían según las necesidades y exigencias del momento.
- No existe documentación de los cambios que se realizan.
- El repositorio SVN no es utilizado por completo por falta de normativas.
- No se guardan versiones finales.

## 3.2. Solución propuesta

El conjunto de problemas anteriormente descritos puede ser solucionado con la creación de procesos transversales que cercioren que se lleven a cabo con éxito todas las actividades definidas en la planificación, e informar los posibles cambios sujetos a ella. Esto con el objeto de garantizar calidad en el producto final entregando mayor confiabilidad al cliente.

También es importante incorporar el monitoreo de los cambios de los elementos del software que se construirá, registrando e informando constantemente de su estado. Con el fin de lograr una buena comunicación entre los participantes del proyecto.

Para concretar el conjunto de soluciones planteadas, se propone desarrollar dos metodologías que cubran las necesidades que existen en DISICO de una forma ordenada, completa y genérica para cada proyecto que se lleve a cabo. De esta forma surge la idea de crear una metodología tanto para el aseguramiento de calidad del software como para la gestión de configuración de software, es decir, SQA Y SCM.

Los procesos de SQA y SCM nos permiten asegurar la calidad del software, actuando de forma pralela durante el desarrollo del proyecto, a través de un equipo capacitado para implementar técnicas y herramientas que sigan constantemente el comportamiento de cada etapa de desarrollo controlando los cambios que se presenten.

### 3.2.1. Metodología SQA Propuesta

Para solucionar los problemas anteriormente identificados se propone una metodología para el proceso de SQA que permita a DISICO aumentar la calidad de sus productos, principalmente a través de procedimientos formales.

Considerando la situación actual del Área de Desarrollo, lo variable que son los requerimientos y las exigencias del cliente, es necesario implementar una metodología ágil y fácil de manejar y que a su vez permita asegurar que se realice el producto de forma correcta y con la menor cantidad de errores posible, todo ésto en un corto plazo.

Ésta metodología debe tener la menor cantidad de documentación posible y debe basarse en templates que exijan la información necesaria para el éxito de las tareas de un proceso de SQA. En éste caso el responsable de QA actual será quien incorpore ésta metodología en el Área de Desarrollo, para lo que será capacitado con anterioridad.

Una vez que el responsable de QA adquiera los conocimientos necesarios sobre el aseguramiento de calidad y comprenda en totalidad la metodología propuesta, la utilizará en un proyecto real de DISICO para observar si se logran beneficios o no a la calidad del producto y al ciclo de vida de su desarrollo.

### **3.2.2. Metodología SCM Propuesta**

Para solucionar los problemas identificados en el control de cambios y versiones es necesario generar un proceso formal de éste. Para la metodología de SCM propuesta se debe considerar la poca disponibilidad de tiempo y recursos para implementar los procedimientos relacionados.

Por lo anterior y para lograr un proceso de SCM de calidad es fundamental crear toda la documentación pertinente a los cambios que se realicen durante el desarrollo de un proyecto, pero siempre buscando que ésta se realice de forma rápida y precisa, apoyada de templates que exijan la información mínima para que la metodología cumpla sus objetivos.

Por otra parte motivará al equipo de trabajo del Área de Desarrollo para generar las políticas necesarias para utilizar todos los directorios propuestos por SVN, para manejar de forma correcta las versiones de cada sistema creado.

Para lograr lo descrito anteriormente se capacitará a quien se hará cargo de éste proceso, adquiriendo los conocimientos necesarios para comprender e interiorizar sus responsabilidades. Tras terminar la capacitación del responsable de SCM, se utilizará la metodología propuesta en un proyecto real de DISICO, para obtener la información que permita concluir si existieron beneficios o no tras terminar el proceso.

# **Capítulo 4**

## **Diseño del Experimento**

Para la validación de las Metodologías de SQA y SCM propuestas en éste Trabajo de Título, se llevará a cabo la investigación de un caso de estudio [21]. Ésta contiene una serie de pasos fundamentales para su correcta realización, los cuales se encuentran descritos y especificados desde la sección 4.1 a la 4.10. La Figura 4.1 representa las actividades de la planificación del método de investigación.

Para diseñar y planificar el caso de estudio, primero se debe seleccionar el contexto en el cuál será llevado a cabo. Luego se define la hipótesis sobre la cual se trabajará, con el fin de refutar o no su validez. Posteriormente es necesario definir las variables tanto dependientes como independientes. También se determina que sujetos serán observados durante el desarrollo de la investigación. En la etapa siguiente, se selecciona el diseño del caso de estudio determinando sus factores y tratamientos. Por otra parte se requiere generar la instrumentación necesaria, es decir material explicativo, capacitaciones, herramientas de medición, entre otros. Los pasos finales de ésta planificación son la ejecución del caso de estudio, la recolección de los datos y el análisis de éstos para así generar las conclusiones pertinentes.



Figura 4.1: Etapas Planificación Caso de Estudio

## 4.1. Selección del Contexto

El caso de estudio se efectuará sobre un conjunto de profesionales pertenecientes al Área de Desarrollo de DISICO, quienes tienen conocimientos básicos sobre ingeniería y gestión de calidad de software. Se utiliza éste tipo de sujetos con el fin de realizar el caso de estudio en el contexto natural donde en el futuro se usarán las metodologías propuestas.

Ésta investigación se hará en las dependencias de DISICO, el cual es el lugar físico donde cada profesional lleva a cabo las actividades asignadas por el Jefe del Área de Desarrollo en la actualidad, es decir, el escenario real.

## 4.2. Formulación Hipótesis

Un paso muy importante en la planificación de un caso de estudio es la formulación de la hipótesis, ya que ayuda a identificar aquello que se desea probar con la investigación.

En éste caso se definirán tanto la hipótesis de trabajo o de investigación y las hipótesis operacionales que nacen de ésta, es decir hipótesis nulas y alternativas. La notación a utilizar para las hipótesis a definir es la siguientes:

- Hipótesis de Trabajo o Investigación:  $H_i$
- Hipótesis Nula:  $H_o$
- Hipótesis Alternativas:  $H_a$

#### 4.2.1. Hipótesis de Trabajo

La Hipótesis de Trabajo o de Investigación es una afirmación sobre las probables relaciones que existen entre dos o más variables. Para el caso de estudio a realizar en éste trabajo de título la hipótesis de trabajo se define a continuación:

«El uso de Metodologías de SQA y SCM durante el ciclo de vida del desarrollo de un proyecto de software, aumenta la calidad del producto final y la satisfacción del cliente.»

#### 4.2.2. Hipótesis Operacionales

Las hipótesis operacionales de dividen en dos tipos.

- Hipótesis Nula: Sirve para negar lo que es afirmado en la hipótesis de trabajo.
- Hipótesis Alternativa: Sirve para dar posibilidades alternas para la hipótesis de trabajo y las hipótesis nulas que se planteen.

A continuación se definen las hipótesis nulas y alternativas, correspondientes a la hipótesis de trabajo planteada en la sección 4.2.1.

- $H_{o1}$ : El uso de una Metodología de SQA durante el ciclo de vida del desarrollo de un proyecto de software no aumenta la calidad del producto final ni la satisfacción del cliente.
- $H_{a1}$ : El uso de una Metodología de SQA durante el ciclo de vida del desarrollo de un proyecto de software aumenta la calidad del producto final y la satisfacción del cliente.
- $H_{o2}$ : El uso de una Metodología de SQA durante el ciclo de vida del desarrollo de un proyecto de software no aumenta el esfuerzo realizado por los integrantes del proyecto.

- $H_{a2}$ : El uso de una Metodología de SQA durante el ciclo de vida del desarrollo de un proyecto de software aumenta el esfuerzo realizado por los integrantes del proyecto.
- $H_{o3}$ : El uso de una Metodología de SCM durante el ciclo de vida del desarrollo de un proyecto de software no aumenta la calidad del producto final ni la satisfacción del cliente.
- $H_{a3}$ : El uso de una Metodología de SCM durante el ciclo de vida del desarrollo de un proyecto de software aumenta la calidad del producto final y la satisfacción del cliente.
- $H_{o4}$ : El uso de una Metodología de SCM durante el ciclo de vida del desarrollo de un proyecto de software no aumenta el esfuerzo realizado por los integrantes del proyecto.
- $H_{a4}$ : El uso de una Metodología de SCM durante el ciclo de vida del desarrollo de un proyecto de software aumenta el esfuerzo realizado por los integrantes del proyecto.

### 4.3. Definición de Variables

Para la investigación de un caso de estudio se definen dos tipos de variables, independientes y dependientes. En las secciones 4.3.1 y 4.3.2 se explican y describen aquellas que serán utilizadas en ésta ocasión.

#### 4.3.1. Variables Independientes

Las variables independientes sirven para describir el tratamientos a usar en la investigación, siendo éstas las que se manipularán para obtener los resultados deseados.

- Metodología SQA: Corresponda a la metodología propuesta en éste trabajo de título para incorporar el proceso de SQA en el Área de Desarrollo de DISICO. Ésta variable tiene como objetivo mejorar la calidad de los servicios entregados por DISICO a estudiantes, profesores y funcionarios de la Universidad de Valparaíso. Busca entregar un monitoreo constante durante el ciclo de vida de cada proyecto que se realice, para asegurar el cumplimiento de los requerimientos y obtener altos niveles de calidad y satisfacción para el cliente.
- Metodología SCM: Corresponde a la metodología propuesta en éste trabajo de título para incorporar el proceso de SCM en el Área de Desarrollo de DISICO. Ésta variable tiene como objetivos mejorar la gestión de los proyectos realizados durante su ciclo de vida completo y gestionar los cambios y las versiones de éstos, permitiendo

asegurar la calidad de los productos que serán entregados a estudiantes, profesores y funcionarios de la Universidad de Valparaíso.

- Metodología Ad-hoc: Corresponde a las actividades y procedimientos reales que actualmente son llevados a cabo en DISICO para el desarrollo de los proyectos de software.

#### 4.3.2. Variables Dependientes

Las variables dependientes son las respuestas de las variables independientes. Éstas variables se miden para visualizar el efecto que tuvieron las variables independientes sobre ellas al ser manipuladas.

- Esfuerzo: Corresponde a las horas-hombres necesarias para llevar a cabo el desarrollo de un proyecto de software, ya sea incorporando las actividades de las Metodologías propuestas o de la Metodología Ad-doc.

Las horas-hombres utilizadas en los proyectos que usen las Metodologías propuestas serán comparadas con las horas-hombres utilizadas en los proyectos donde se use la Metodología Ad-hoc, de éste forma identificar cual implica mayor esfuerzo.

El esfuerzo realizado en cada proyecto se obtendrá a través de la recolección de datos realizada a través de las encuestas y check-list correspondientes, obteniendo el número de horas que tardó el grupo de sujetos en aplicar las metodologías en cada proyecto.

- Calidad: Será observada en el número de requerimientos cumplidos en los proyectos de software a estudiar. Se comparará el porcentaje de cumplimiento de los requerimientos de aquellos proyectos donde fueron usadas las metodologías propuestas contra los proyectos donde se utilizó la metodología Ad-hoc, de ésta forma evaluar cual de éstas entrega más calidad al producto final desarrollado y a su vez satisfacción la cliente.

La calidad basada en el cumplimiento de los requerimientos del sistema será medida con los datos recolectados por los documentos generados, específicamente el de Verificación y Validación detallado en la sección 5.2.4. En éste se evalúa el número de requerimiento funcionales y no funcionales cumplidos correctamente, permitiendo calcular el porcentaje éstos. El calculo se realizará de la siguiente forma:

$$\% \text{ Cumplimiento Req.} = (\text{Cantidad Req. Cumplidos} / \text{Total Req.}) * 100$$

Por otra parte los errores en un sistema también indican la calidad de éste, permitiendo comparar el porcentaje de errores existentes en los proyectos donde se usan las metodologías propuestas y el porcentaje de errores presente en los proyectos desarrollados con la metodología Ad-hoc. Éste se calcula de la siguiente forma:

$$\% \text{ Errores} = (\text{Cantidad de Errores} / \text{Total líneas de Código}) * 100$$

#### 4.4. Selección Sujetos

Los sujetos seleccionados para el caso de estudio que permitirá validar las metodologías propuestas, serán aquellos que sean asignados a los proyectos que se estudiarán. Esta asignación se realiza a través de los criterios que actualmente se usan en el Área de Desarrollo de DISICO para determinar que sujetos trabajarán en determinado proyecto.

Para llevar a cabo cualquier proyecto son asociados a él, un máximo de dos desarrolladores, un responsable de QA y un Jefe de Proyecto, es decir, para cada caso a estudiar se involucrarán cuatro personas, además del Jefe del Área de Desarrollo. Se estudiarán dos casos donde se usen las metodologías de SQA y SCM propuestas y dos casos donde se utilice la metodología Ad-hoc.

#### 4.5. Selección del Diseño del Experimento

El diseño del experimento para la validación de las metodologías propuestas se basará en un caso de estudio. Éste es un diseño observacional que permite al investigador obtener información suficiente para generar conclusiones sobre determinadas acciones de los sujetos en un escenario real. El investigador tiene una participación pasiva, no se debe involucrar durante el proceso. El enfoque principal de la investigación es ordenar, a través de diferentes técnicas, la información que se obtenga durante el estudio de cada caso.

La investigación será de un factor y dos tratamientos, donde se compararán los resultados obtenidos para cada uno, en relación a las variables dependientes definidas. Cada equipo por proyecto llevará a cabo un tratamiento, es decir el uso de las metodologías propuestas o la metodología Ad-hoc.

#### 4.6. Instrumentación

Para la ejecución del caso de estudio es necesario tener acceso a la instrumentación que permite a los sujetos seleccionados entender, analizar y utilizar las metodologías propuestas

en éste trabajo de título. Para esto se definirán los instrumentos a utilizar en la Metodología de SQA, SCM y Ad-hoc.

#### **4.6.1. Instrumentos Metodología SQA**

La instrumentación para la Metodología SQA necesaria es la siguiente:

- Capacitación: Instancia donde los sujetos que participarán del caso de estudio, son instruidos en la Metodología de SQA propuesta. Se les entrega de forma expositiva el contenido de la metodología, explicando como debe ser incorporada en un proyecto de desarrollo.
- Presentación: Material de apoyo necesario para llevar a cabo la capacitación para la metodología SQA a los sujetos. La presentación tiene el contenido mínimo para poder instruir a los participantes.
- Lugar físico: El caso de estudio se realizará en las dependencias de DISICO, específicamente en el Área de Desarrollo.
- Proyecto: Para la validación de la Metodología SQA se necesita incorporar ésta en dos proyectos reales de DISICO, para poder observar si trae beneficios o no a éstos.
- Documentación: Documento de la Metodología de SQA, la cual debe estar a disposición de los sujetos que participarán en el caso de estudio, para cuando lo estimen conveniente. Ésta documentación define la Metodología de SQA propuesta, presentando cada una de sus actividades.
- Herramientas: Para llevar a cabo la correcta incorporación de la Metodología de SQA, es necesario disponer de los template adjuntos al documento de la metodología, donde se reúne el conjunto de documentación mínima para el correcto uso del proceso.

#### **4.6.2. Instrumentos Metodología SCM**

La instrumentación para la Metodología SCM necesaria es la siguiente:

- Capacitación: Instancia donde los sujetos que participarán del caso de estudio, son instruidos en la Metodología de SCM propuesta. Se les entrega de forma expositiva el contenido de la metodología, explicando como debe ser incorporada en un proyecto de desarrollo.
- Presentación: Material de apoyo necesario para llevar a cabo la capacitación para la metodología SCM a los sujetos. La presentación tiene el contenido mínimo para poder instruir a los participantes.

- Lugar físico: El caso de estudio se realizará en las dependencias de DISICO, específicamente en el Área de Desarrollo.
- Proyecto: Para la validación de la Metodología SCM se necesita incorporar ésta en dos proyectos reales de DISICO, para poder observar si trae beneficios o no a éstos.
- Documentación: Documento de la Metodología de SCM, la cual debe estar a disposición de los sujetos que participan en el caso de estudio, para cuando lo estimen conveniente. Ésta documentación define la Metodología de SCM propuesta, presentando cada una de sus actividades.
- Herramientas: Para llevar a cabo la correcta incorporación de la Metodología de SCM, es necesario disponer de los template adjuntos al documento de la metodología, donde se reúne el conjunto de documentación mínima para el correcto uso del proceso.

#### 4.6.3. Instrumentos Metodología Ad-hoc

La instrumentación para la Metodología Ad-hoc necesaria es la siguiente:

- Lugar Físico: El caso de estudio se realizará en las dependencias de DISICO, específicamente en el Área de Desarrollo.
- Proyecto: Se necesita evaluar el uso de la Metodología Ad-hoc en dos proyectos reales de DISICO, para poder observar si trae beneficios o no a éstos.
- Documentación: Consiste en la documentación de la Metodología Ad-hoc, la cual debe estar a disposición de los sujetos participantes en el caso de estudio para cuando lo estimen conveniente. Ésta documentación define los procedimientos actualmente usados en el Área de Desarrollo de DISICO.

### 4.7. Fuentes de Invalidación

#### 4.7.1. Fuentes de Invalidación Interna

Las fuentes de invalidación interna, son aquellas que atentan contra la validez interna del caso de estudio a realizar. Las principales fuentes para ésta investigación son:

- Historia: Aquellos sucesos externos que pueden ocurrir durante la ejecución del experimento, pueden provocar que los sujetos a observar, adquieran conocimientos fuera de los entregados en las capacitaciones correspondientes a las metodologías propuestas.

- Instrumentación: Pueden existir errores durante la ejecución del experimento por parte de los sujetos, a causa de inexactitud en la instrumentación utilizada para la aplicación de las metodologías propuestas en los proyectos a estudiar. También éstas fallas se pueden presentar por imprecisiones en las herramientas de recolección de datos usadas en el caso de estudio.

#### 4.7.2. Fuentes de Invalidación Externa

Las fuentes de invalidación externa es el paso de los datos del caso de estudio a otros tipos de situaciones, es decir, es el grado de posibilidad de generalizar un experimento. Las principales fuentes de éste tipo son:

- Ambiente y Tratamientos: El ambiente donde se concretará el caso de estudio y la incorporación de las metodologías propuestas y el uso de la metodología Ad-hoc, corresponde al lugar natural donde se realizan los proyectos de desarrollo de DISICO actualmente, por lo tanto no traerá resultados inesperados como lo sería si el ambiente no fuera el apropiado.
- Historia y Tratamientos: Ésta amenaza se considera controlada, ya que la planificación de las capacitaciones se realizará antes de la ejecución del experimento, permitiendo a los sujetos tener el tiempo suficiente para interiorizar de forma correcta el uso de las metodologías propuestas.

### 4.8. Ejecución del Caso de Estudio

La ejecución de la investigación del caso de estudio propuesto, está representada por la Figura 4.2, en la cual se plantean los siguientes pasos:

- Capacitación de Metodologías propuestas: Es necesario capacitar a los sujetos que utilizarán las metodologías propuestas de SQA y SCM durante el desarrollo de dos proyectos reales de DISICO, de ésta forma adquieran las herramientas intelectuales necesarias para saber realizar las actividades relacionadas. Por otra parte no es necesario realizar capacitación a los sujetos que utilizarán la metodología Ad-hoc, ya que corresponde a aquellos procedimientos que actualmente son llevados a cabo en todos los proyectos, es decir, ya tienen conocimiento y manejo sobre ellos.
- Desarrollo del proyecto: Ésta etapa se lleva a cabo durante el ciclo de vida del proyecto. Se realizará bajo dos enfoques, los cuales son:
  - Uso de Metodologías propuestas: En éste caso en el desarrollo de dos proyectos se incorporarán las metodologías propuestas para los procesos de SQA y SCM.

- Uso de Metodología Ad-hoc: En éste caso en el desarrollo de dos proyectos se utilizará la metodología Ad-hoc, la que corresponde a los procedimientos usados en la actualidad en el Área de Desarrollo de DISICO para todos los proyectos.
- Recolección de Datos: Corresponde a la recolección y clasificación de los datos que se obtengan con la ayuda de técnicas como entrevistas y encuestas a los integrantes de cada proyecto a estudiar. Las herramientas de aplican a los sujetos que utilizaron las metodologías propuestas y aquellos que usaron la metodología Ad-hoc.
- Análisis de Datos: Luego de la recolección de datos se realiza su análisis a través de métodos de categorización y herramientas, de ésta forma poder comparar los datos obtenidos para los casos de estudio e identificar las diferencias entre ellos.
- Conclusiones: Finalmente luego del análisis de datos y la obtención de la información suficiente para dar por terminada la etapa de análisis, se inicia la documentación del procedimiento, incluyendo las conclusiones correspondientes que permitan validar o refutar las hipótesis planteadas para los casos de estudio.

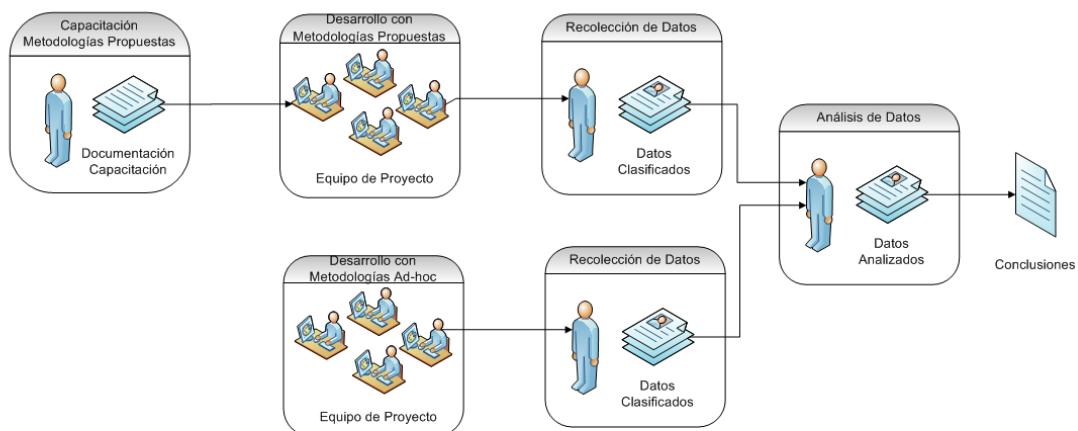


Figura 4.2: Etapas Ejecución Caso de Estudio

## 4.9. Recolección de Datos

Para la recolección de datos se utilizarán dos herramientas escritas que permitirán la obtención de la información necesaria para el análisis del caso de estudio. Las técnicas a utilizar son:

- Encuesta: Es parte de un estudio observacional que permite reunir información de acuerdo a una serie de preguntas normalizadas dirigidas a los sujetos que participan de cada caso de estudio necesario para la validación de las metodologías propuestas en éste trabajo de título (Ver Apéndice A).
- Entrevista: Se entrevista a cada sujeto participante del caso de estudio para conocer apreciaciones personales que no se puedan observar con las encuestas, como anécdotas vividas durante el desarrollo del proyecto relacionadas con el uso de las metodologías propuestas o la metodología Ad-hoc (Ver Apéndice B).

- Notas: Consiste en anotaciones del investigador sobre hechos o circunstancias que se observen dentro de los equipos de trabajo de cada proyecto a estudiar.

## 4.10. Análisis de Datos

Los datos recolectados a lo largo del caso de estudio, debido a ser parte fundamental del éxito de la investigación, deben ser cuidadosamente analizados. Es por esto que la información se debe organizar y archivar de forma coherente para lograr que éstos sean manejables y de fácil acceso, con el propósito de analizarlos y realizar las comparaciones pertinentes.

Los datos a analizar son aquellos recolectados por las encuestas y entrevistas realizadas a los sujetos que participaron en el caso de estudio y también aquellos hechos observados por el investigador, que fueron registrados. Para el correcto análisis de los datos se deben realizar dos pasos importantes: la preparación y organización de los datos, y el análisis de éstos.

### 4.10.1. Preparación y organización de Datos

Para el análisis de los datos se deben organizar de diferentes formas para observar el comportamiento del caso de estudio. La organización de la información se realizará bajo diferentes conceptos.

- Por Proyecto: Se organizarán los datos para ser analizados será por proyecto. Se compararán los datos entre los cuatro proyectos por separado, independientes del tratamiento usado en cada uno, lo que permitirá observar si existen diferencias entre los proyectos, y si estos dependen de las metodologías usadas durante el desarrollo del sistema o son por factores externos. También se organizarán en una misma categoría los dos proyectos donde se utilizaron las metodologías propuestas para ser comparadas con los datos obtenidos de los dos proyectos donde se uso la metodología Ad-doc, lo que permitirá ver si existentes diferencias positivas o negativas entre la incorporación de determinada metodología en el desarrollo de software de cada proyectos.
- Por Persona y Cargo: Se organizará la información obtenida del caso de estudio por cada sujeto que participó de éste. De ésta forma se podrá observar si existe algún comportamiento reiterativo en alguno de los sujetos y determinar si la información entregada por éste es de calidad y fidedigna. Por otra parte la información se clasificará por el cargo que tenga cada uno de éstos, ya sea desarrollador, jefe de proyecto o responsable de SQA y SCM, para identificar si existe alguna tendencia basada en las responsabilidades asignadas a cada uno.

- Por Metodología: La información se organizará según las metodologías usadas en cada proyecto. Primero se compararán los datos de las metodologías de SQA y SCM combinadas, contra la información de la metodología Ad-hoc. Luego se analizará la información de cada una de las metodologías por separado. Ésto permitirá concluir si las diferencias entre cada proyecto están influenciadas por la metodología usada en cada uno de ellos.

#### **4.10.2. Análisis y Conclusiones**

Luego de la organización de la información, según las tres categorías descritas en la sección 4.10.1, ésta debe ser respaldada en una base de datos de fácil acceso. Una vez que la información se encuentre ordenada por categorías será estudiada en detalle por cada uno de sus atributos, por ejemplo el tiempo total que tardó el desarrollo de los proyectos, los errores que existieron en él y las situaciones que afectaron el desempeño de los sujetos, entre otros.

Luego de haber identificado la información relevante de los datos obtenidos, tras realizar y analizar el caso de estudio, se deben elaborar las conclusiones referentes a éstos. Las conclusiones deben permitir refutar o no la hipótesis de investigación y a su vez las hipótesis operacionales definidas en la sección 4.2.2, entregando como resultado la validación o no de la metodologías propuestas en éste trabajo de título.

# Capítulo 5

## Diseño Metodología SQA

El propósito de la Metodología de SQA es principalmente establecer los requisitos mínimos que se deben cumplir en el proceso de aseguramiento de calidad [4]. En particular ésta metodología busca entregar un proceso formal al Área de Desarrollo de DISICO para asegurar la calidad de los servicios entregados a estudiantes, profesor y funcionarios de la Universidad de Valparaíso.

Tiene la particularidad de adaptarse a los recursos y necesidades que actualmente existentes en el área, especialmente la falta de recurso humano. El uso de éste procedimiento está enfocado a todos los proyectos de software a desarrollar, siempre y cuando su urgencia y limitaciones lo permita. Las etapas a definir para ésta metodología se representan en la Figura 5.1.



Figura 5.1: Etapas Metodología SQA

## 5.1. Tareas y Responsables

En el Área de Desarrollo de DISICO, las entidades encargadas del aseguramiento de calidad de los software desarrollados, son específicamente el Jefe del Área de Desarrollo, el Jefe de Proyecto, el responsable de QA y los desarrolladores involucrados.

Cada uno de ellos evalúa la validez y el cumplimiento de los requerimientos de cada proyecto a través de diferentes actividades. Es importante incorporar en las tareas de la Metodología de SQA, las revisiones de éstas y sus entregables (Ver sección 5.2.4).

- **Tareas:** Una vez identificados quienes son los involucrados en un proyecto determinado, se deben listar las tareas que se realizarán durante el ciclo de vida del proyecto. Ésto debe definir no solo la actividad si no también el tiempo máximo en el cual ésta se debe llevar a cabo.
- **Responsables:** Las tareas definidas para el aseguramiento de calidad de determinado servicio a desarrollar, deben ser asignadas a los participantes de éste, según corresponda. Basándose en las habilidades profesionales del sujeto éste debe concretar en el tiempo estimado la actividad que le sea asignada según la función que cumple dentro del proyecto.

Una vez definidas las tareas de aseguramiento de calidad correspondiente a las secciones 5.2 a la 5.4 y asignado un responsable para cada una, se debe realizar la documentación y formalización del suceso. Ésto se realiza completando el template de tareas y responsabilidades que se encuentra en el apéndice D.1. El documento debe ser aprobado por el Jefe del Área de Desarrollo.

Por otro lado se debe usar en ésta metodología un **Check-list** que permitirá corroborar el avance de las tareas de SQA mínimas a realizar. Cada semana, desde el comienzo del proyecto, se deberá completar un Check-list, lo que ayudará a identificar si los integrantes del proyecto se encuentran o no ociosos. El template de éste documento se encuentra en el apéndice D.2.

El proceso anteriormente definido, si bien corresponde a la metodología de SQA propuesta, también puede ser utilizado para monitorear otras tareas relacionadas con el desarrollo de un proyecto de software, si se estima conveniente.

## 5.2. Documentación

La documentación de la Metodología de SQA es una de las actividades principales a realizar para incorporar éste proceso. Desde la sección 5.2.1 a la 5.2.4 se detallan las características de cada documento a realizar y los pasos necesarios para ser elaborado. También cada uno cuenta con un template que se encuentra en el apéndice D, que permitirá agilizar el proceso. Los documentos definidos es el conjunto mínimo que se debe concretar para la correcta incorporación de la Metodología de SQA, por lo cual la ausencia de alguno de los documentos debe encontrarse claramente justificada. Por otra parte, si el proyecto lo amerita, se puede agregar otra documentación a éste conjunto.

### 5.2.1. Diseño y Especificación de Requerimientos de Software

El Documento de Diseño y Especificación de Requerimientos de Software, contiene la información necesaria para saber que es lo que hará el sistema. Se detallan cada uno de los requisitos involucrados y se define como el sistema será capaz de satisfacerlos.

Por un lado la especificación de los requerimientos deben ser claros, concretos y completos. Es importante que cada uno de ellos se encuentre claramente justificado, ya que se debe saber porque es importante su existencia. También se deben generar los casos de uso correspondientes a los requerimientos descritos, indicando el modo en que un actor interactúa con el sistema, éstos permiten explicar su funcionamiento.

El template que permitirá agilizar el generar éste documento, se encuentra en el apéndice D.3, su estructura es la siguiente:

- Descripción general del Proyecto
- Descripción general del Requerimiento
- Funciones y Descripción de éstos
- Casos de Uso
- Modelo Conceptual
- Modelo de Datos
- Listado y breve descripción de procedimientos almacenados
- Modelo de Componentes
- Diagrama de Clases

### 5.2.2. Documentación del Usuario

El Documento del Usuario, entrega a los usuarios una guía completa y precisa del uso del sistema desarrollado. Ésto implica describir las funcionalidades que entrega la aplicación, indicando los pasos de cada una de ellas. También es importante detallar las restricciones que existan para un uso correcto del software, y otros elementos que se consideren necesarios para lograr la satisfacción del cliente durante su experiencia con el servicio entregado. La información sobre el comportamiento del software, se debe entregar de forma clara, indicando posibles errores y sus respectivas acciones correctivas.

La estructura del Documento del Usuario o Manual, que se encuentra en el apéndice D.4 ,es la siguiente:

- Descripción del Sistema
- Requisitos para el uso del sistema
- Funcionalidades
- Observaciones

### 5.2.3. Documento de Pruebas

El Documento de Prueba permite estructurar y definir las pruebas a realizar y como se llevarán a cabo. Entrega información sobre las entradas que se le entregarán al sistema y cuales son las salidas esperadas de éstas. Una vez realizado el documento se debe realizar la ejecución de dichas pruebas documentadas. Al obtener los resultados y errores encontrados, éstos deben ser agregados al informe.

Independiente de la urgencia del proyecto, se deben documentar los resultados arrojados por cada prueba realizada a lo largo del proceso. La ejecución de las pruebas se concretará con la ayuda de la herramienta Mantis, que permite gestionar el estado de las incidencias encontradas durante la ejecución de las pruebas.

La estructura del documento de pruebas, que se encuentra en el apéndice D.6, es la siguiente:

- Diseño de los Casos de Prueba manuales
- Ejecución de los Casos de Prueba
- Reporte de Resultados Finales
- Resultado de las pruebas de Aceptación

#### 5.2.4. Verificación y Validación

Ésta actividad se debe realizar por el Jefe de Proyecto en conjunto con el Jefe del Área de Desarrollo, basándose en el documento ERS y en los resultados de las pruebas realizadas.

El principal objetivo del documento de verificación y validación es documentar el comportamiento del proyecto durante su ciclo de vida, presentando así un análisis de las tareas realizadas y los problemas que se presentaron durante el proceso.

Éstas verificaciones permitirán a su vez identificar que se hayan realizado todas las tareas asignadas a cada sujeto. Una vez aprobados los documentos, se debe generar un informe detallado con los resultados de éste proceso siguiendo el template que se encuentra en el apéndice D.5, cuya estructura principal es la siguiente:

- Descripción del Problema
- Requerimientos Cumplidos
- Requerimientos Ausentes y Justificación
- Check-List Asociados
- Resumen Check-List
- Conclusión

### 5.3. Recursos

Para apoyar el trabajo que significa realizar un proceso de SQA son necesarios ciertos recursos. Éstos son principalmente la herramienta Mantis, los template correspondientes a la Metodología de SQA y personal perteneciente al Área de Desarrollo de DISICO.

Para la realización de las pruebas se utilizará **Mantis**, siendo ésta una herramienta de gestión de incidencias muy fácil de usar [13]. A través de ella se hará saber a los desarrolladores los errores encontrados tras la ejecución de las pruebas, para que den una solución a cada uno de ellos. El responsable de SQA seguirá el estado de la incidencia, verificando a través de Mantis si se efectuó el cambio solicitado.

Se necesita la **documentación** propuesta por la Metodología de SQA, es decir, cada uno de los template que se encuentran adjuntos a éste documento, en el apéndice D.

Por otra parte se requiere de **personal** para incorporar la Metodología de SQA con éxito en un proyecto de desarrollo. Particularmente del Área de Desarrollo de DISICO se necesita la participación del responsable de SQA, Jefe de Proyecto, Jefe del Área de Desarrollo y los desarrolladores involucrados en dicho proyecto.

## 5.4. Formación

Es necesario realizar ciertas actividades que ayuden a entregar una justificación del uso de una metodología que apoye el aseguramiento de calidad de los productos. Ésto permite que los integrantes de DISICO comprendan la necesidad de incorporar un proceso de aseguramiento de calidad en el desarrollo de los proyectos que se realicen, tomando conciencia de su importancia para entregar servicios de calidad.

Para realizar la formación de los integrantes del Área de Desarrollo, ésta metodología entrega el material de apoyo mínimo necesario. Es importante considerar el hecho de realizar modificaciones a éste según cambien las necesidades en el Área de Desarrollo. El material se encuentra en el apéndice D.7.

# Capítulo 6

## Diseño Metodología SCM

La Metodología de SCM propuesta entrega los lineamientos mínimos para llevar a cabo éste proceso en el Área de Desarrollo de DISICO. En particular busca dar un mejor manejo de los cambios durante todo el ciclo de vida del software de cada proyecto que se lleve a cabo [7], de ésta forma colaborar con el aseguramiento de calidad de los servicios entregados a estudiantes, profesor y funcionarios de la Universidad de Valparaíso.

Permite una mejor gestión de las versiones finales de cada sistema que sea desarrollado. El uso de ésta metodología se encuentra enfocado a todos los proyectos de software a desarrollar, siempre y cuando su urgencia y limitaciones lo permita. Las etapas principales que componen éste proceso son representadas en la Figura 6.1.



Figura 6.1: Etapas Metodología SCM

## 6.1. Tareas y Responsables

Las entidades encargadas de la gestión de cambios y versiones de los software desarrollados, son específicamente el Jefe del Área de Desarrollo, el Jefe de Proyecto y los desarrolladores involucrados. Son ellos quienes tienen que concretar las actividades de la Metodología de SCM.

En conjunto deben identificar las tareas mínimas a realizar durante el proceso y a su vez, determinar un responsable para cada una de ellas. De ésta forma contribuyen al cumplimiento y validación de las tareas asignadas. El grupo de tareas y asignaciones debe estar claramente documentado, para lo que se debe utilizar el template que se encuentra en el apéndice E.1.

- **Tareas:** La identificación de tareas es importante, ya que de ellas depende la profundidad del seguimiento que se va a realizar al proceso de SCM durante el desarrollo de un proyecto de software. Se debe definir no solo la actividad si no también el tiempo máximo en el cual ésta se debe ejecutar.
- **Responsables:** Las tareas definidas, para la gestión de configuración de software, de cada proyecto deben ser asignadas a los participantes de éste, según corresponda. Ésto quiere decir que basándose en las habilidades profesionales del sujeto éste debe concretar en el tiempo estimado la actividad que le sea asignada según la función que cumple dentro del proyecto.

Una vez definidas las tareas de gestión de configuración de software correspondiente a las secciones 6.2 a la 6.4 y asignado un responsable para cada una, se debe realizar la documentación y formalización del suceso. Se debe completar el template adjunto en el apéndice E.1.

Por otro lado se debe usar un **Check-list** que permitirá corroborar el avance de las tareas mínimas de SCM a realizar. Cada semana, desde el comienzo del proyecto, se deberá completar un Check-list, lo que ayudará a identificar si los integrantes del proyecto se encuentran o no ociosos. El template de éste Check-List se encuentra en el apéndice D.2.

## 6.2. Actividades de SCM

Las actividades de SCM presentan las funciones y tareas necesarias para gestionar la configuración de un sistema de software. Las tareas de la Metodología de SCM son: la identificación de configuración y control de configuración. La definición de cada una de ellas y los requisitos mínimos se encuentran desde la sección 6.2.1 a la 6.2.4.

### 6.2.1. Identificación de Configuración

La Identificación de Configuración permite determinar cuales serán los items que se seguirán durante el ciclo de vida de proyecto. Para realizar éste procedimiento es necesario considerar dos puntos importantes descritos a continuación:

- **Identificación de Items de Configuración:** El responsable de SCM en conjunto con el responsable de SQA deben determinar que entregables serán objeto de control según las exigencias de cada proyecto. Los el item de configuración mínimos a considerar son aquellos presentes en la Metodología de SQA. Es recomendable considerar la mayor cantidad de entregables a controlar.
- **Nombrar Items de Configuración:** Los items de configuración deben tener un identificador único. Los nombre o identificador puede ser entregado de forma manual o utilizar el entregado por defecto por SVN.

Para la formalización de ésta tarea se debe utilizar el template que se encuentra en el apéndice E.2, el cual permite registrar cuales son los items de configuración que se controlarán y sus atributos.

### 6.2.2. Configuración de Control

La Configuración de Control tiene como objeto gestionar los cambios que se realicen a lo largo del ciclo de vida de cada proyecto. Para ésto es necesario seguir una serie de pasos como evaluar el cambio, aprobar o desaprobar y aplicar el cambio al item de configuración.

El responsable de SCM, el Jefe de Proyecto y el Jefe del Área de Desarrollo, son los encargados de evaluar cada uno de los casos que se presenten y determinar si es factible para el proyecto que se esté desarrollando. Para el éxito de ésta actividad es necesario documentar los cambios y su gestión utilizando los template que se encuentran en el apéndice E.3. La Figura 6.2 presenta el flujo del proceso de configuración de cambio.

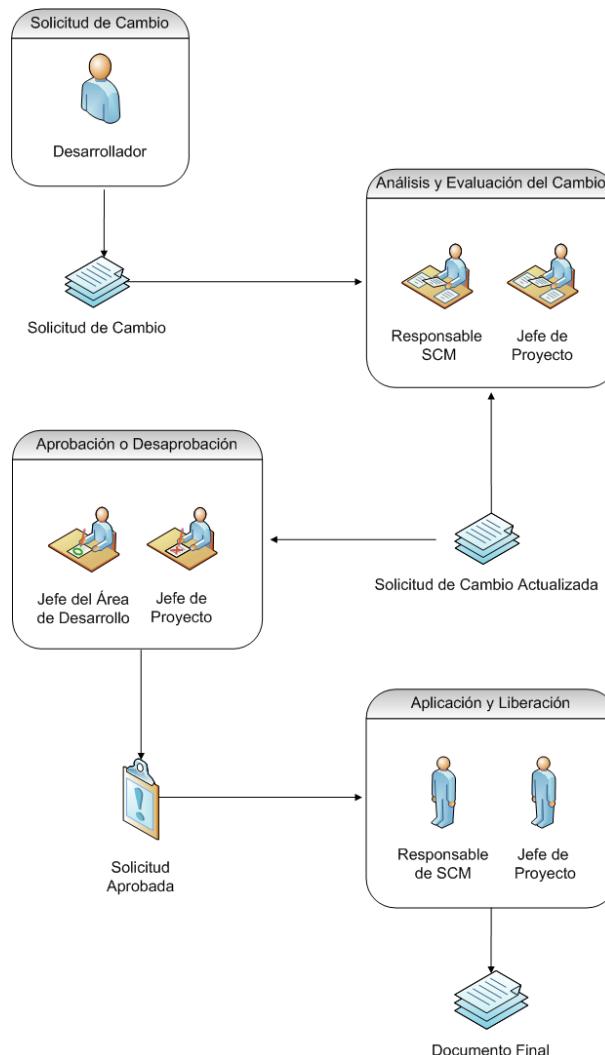


Figura 6.2: Proceso de Configuración de Cambio

La secuencia de pasos es la siguiente:

### 1. Identificación y documentación de la necesidad de un cambio

Se deben indicar los principales datos de las modificaciones que se desean realizar. Para ésto se debe enviar una solicitud de cambio al responsable de SCM usando el template que se encuentra en el apéndice E.3.1.

La información que debe contener como mínimo el documento es la siguiente:

- **Nombre y Versión:** Se debe entregar el nombre del ítem de configuración donde se ha identificado un problema y la versión de éste.
- **Datos Solicitante:** Entregar los datos del solicitante como el nombre y la organización a la cual pertenece, ya sea el cliente o parte del equipos del Área de Desarrollo.
- **Fechas de solicitud:** Se debe ingresar la fecha en la cual se ha ingresado la solicitud de cambio.
- **Urgencia:** Indicar la urgencia que tiene el cambio solicitado, para dar la prioridad que corresponde a su aprobación e implementación. La urgencia debe ser indicada con la nomenclatura definida en el Cuadro 6.1.

Tipo de Urgencia	Definición
Baja	El cambio no necesita ser priorizada sobre el resto de los cambios
Media	El cambio debe ser priorizado sobre los cambios de prioridad baja
Alta	El cambio debe ser priorizado el resto de los cambios

Cuadro 6.1: Nomenclatura de urgencia de solicitudes de cambios

- **Necesidad del cambio:** Especificar cual es el motivo que por el cual se solicita realizar el cambio en el ítem de configuración, ya sea por error o variación en las necesidades del cliente.
- **Descripción del cambio solicitado:** Definir cual es el cambio que se debe realizar en el ítem de configuración.

## 2. Análisis y evaluación de una solicitud de cambio

El responsable de SCM y el Jefe de Proyecto deben evaluar la solicitud de cambio y determinar cual es el impacto que éste puede tener sobre el desarrollo del proyecto, sobre el producto final y los recursos disponibles, ésto en común acuerdo con el Jefe del Área de Desarrollo.

Si los evaluadores consideran que la solicitud de cambio presenta errores, se debe informar al autor de ésta, para que realice las correcciones pertinentes.

El análisis final debe ser incluido en el documento de solicitud de cambio, indicando cuales son las repercusiones que puede provocar en el desarrollo del proyecto y el éxito de éste.

### 3. La aprobación o desaprobación de una solicitud

Una vez realizado el análisis correspondiente a la solicitud de cambio el Jefe de Proyecto y el Jefe del Área de Desarrollo deben determinar si la petición se aprueba o se desaprueba. Ésto debe reflejarse en el documento de solicitud dándose por concluida.

Si la solicitud de cambio es desaprobada por el Jefe de Proyecto y el Jefe de Área del Área de Desarrollo el procedimiento termina con ésta tarea, indicando en el documento que no fue autorizada la realización del cambio y cual es la justificación de ésta decisión.

### 4. La aplicación y liberación del cambio

Una vez realizada la solicitud de cambio, su análisis y aprobación, el responsable de SCM debe generar un documento de verificación donde indique cual fue el procedimiento que se efectuó.

La información registrada para la implementación de un cambio deberá contener como mínimo los siguiente puntos:

- La solicitud de cambio asociado
- Los nombres y las versiones de los elementos afectados
- Fecha de Verificación y el responsable
- Fecha de la versión y parte responsable
- El identificador de la nueva versión

Posterior a ésto se debe indicar al responsable del ítem de configuración asociado a la solicitud, que puede proceder a realizar el cambio según su prioridad y la descripción entregada. El template del documento de implementación del cambio se encuentra en el apéndice E.3.2.

### 6.3. Recursos de SCM

Para apoyar el trabajo que significa realizar un proceso de SCM son necesarios ciertos recursos. Éstos son la herramienta Subversion, los template correspondientes a la Metodología de SCM y personal perteneciente al Área de Desarrollo de DISICO.

Para concretar el uso de una biblioteca de software que contenga los items de configuración y baseline descritos durante el ciclo de vida de cada proyecto de software se utilizará la herramienta **Subversion**, la cual permite gestionar la documentación y codificación de un proyecto de forma organizada [1].

También se necesita la **documentación** propuesta en la Metodología de SCM, es decir, cada uno de los template que se encuentran adjuntos a éste documento, en el apéndice E.

Por otra parte se requiere de **personal** para incorporar la Metodología de SCM con éxito en cada proyecto de software. Particularmente del Área de Desarrollo de DISICO se necesita la participación del responsable de SCM, Jefe del Área de Desarrollo, Jefe de Proyecto y los desarrolladores involucrados en dicho proyecto.

### 6.4. Formación

Es necesario realizar ciertas actividades que ayuden a entregar una justificación del uso de una metodología que apoye la gestión de configuración de los productos. Esto permite que los integrante de DISICO comprendan la necesidad de incorporar un proceso de control en el desarrollo de los servicios a entregar, tomando conciencia de su importancia y se desempeñen de la mejor forma en las actividades que le sean asignadas.

Para realizar la formación de los integrantes del Área de Desarrollo en su totalidad, ésta metodología entrega el material de apoyo mínimo necesario. Es importante considerar el hecho de realizar modificaciones a éste según cambien las necesidades en el Área de Desarrollo. El material se encuentra en el apéndice D.7.



# Apéndice A

## Encuesta Sujetos

### Encuesta

Permite reunir información de acuerdo a una serie de preguntas normalizadas dirigidas a los sujetos que participan de cada caso de estudio necesario para la validación de las metodologías propuestas en éste trabajo de título.

**Instrucciones:** La encuesta debe ser respondida una vez leído el documento de la metodología a evaluar y una vez terminado su uso en un proyecto de desarrollo de software. Las respuestas pueden variar entre los siguientes términos:

- ✓ **Si:** El ítem se ha cumplido satisfactoriamente.
- ✓ **No:** El ítem no se ha cumplido satisfactoriamente.
- ✓ **Ocasional:** El ítem se cumple parcialmente.
- ✓ **No Aplica:** El ítem no aplica.

Encuesta: Evaluación de las Metodologías				
1.Datos Encuestado				
Nombre Encuestado:	Fecha Encuesta:			
2.Metodología a Evaluar				
<input type="radio"/> SQA	<input type="radio"/> SCM	<input type="radio"/> Ad-Hoc		
3.Preguntas	Si	No	Ocasional	No Aplica
Compleitud				
¿Promueve la definición de tareas y responsables de las tareas que componen los procesos de gestión de calidad?				
¿Apoya la planificación del proyecto de software?				
¿Promueve el cumplimiento del proceso de documentación?				
¿Funciona como apoyo al proceso de desarrollo?				
¿Permite seguir el estado de las tareas dentro del proyecto de software?				
¿Promueve la formación de los integrantes de los proyectos de software?				
¿Los procesos definidos se relacionan con la gestión de calidad se proyectos de software?				
¿Los recursos propuestos en la metodología resultan un apoyo al proceso?				
Consistencia				
¿Es consistente de acuerdo a los objetivos que tiene el Área de Desarrollo?				
¿Los procesos incluidos son consistentes con el proceso desarrollo de software?				
¿Entrega una experiencia práctica para el desarrollo de proyectos de software?				
Correctitud				

¿Los procesos contenidos en la Metodología son necesarios para el desarrollo de proyectos de software en el Área de Desarrollo?				
¿Es un aporte para el Área de Desarrollo de DISICO?				
¿Los procesos son trazables al proceso de desarrollo de proyectos de software?				
¿Los objetivos de la Metodología se encuentran claramente definidos?				
¿Ayuda a entregar productos de software de calidad?				
<b>Adaptación al Cambio</b>				
¿Los integrantes de DISICO tienen los conocimientos para implementar la metodología?				
¿Los procesos definidos permiten realizar cambios en las tareas?				
¿Los procesos definidos permiten realizar cambios en la documentación?				
¿Es posible eliminar o modificar procesos de la metodología?				
¿Permite agregar nuevos procedimientos o tareas al proceso?				



# Apéndice B

## Entrevista Sujetos

### Entrevista

Se entrevistará a cada sujeto participante del caso de estudio para conocer apreciaciones personales que no se puedan observar con las encuestas, como anécdotas vividas durante el desarrollo proyecto relacionado con el uso de las metodologías propuestas o la metodología Ad-hoc.

**Instrucciones:** El investigador debe completar el documento de la entrevista con la información entregada por el entrevistado. Las preguntas pueden variar según la experiencia obtenida con el uso de la metodología por cada sujeto a entrevistar.

Entrevista: Evaluación de las Metodologías		
1.Datos Entrevistado		
Nombre Entrevistado:	Cargo Entrevistado:	Fecha Entrevista:
2.Metodología a Evaluar		
<input type="radio"/> SQA <input type="radio"/> SCM <input type="radio"/> Ad-Hoc		
2.Entrevista		
Preguntas	Datos Obtenidos	

# Apéndice C

## Capacitaciones

### C.1. Capacitación SQA



#### Metodología de SQA

Capacitación

Paloma Méndez Soto

## ¿Qué es la Calidad de Software?

- La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario. [IEEE, Std 610-1900]



## Objetivos

- Comprobar si un producto posee una determinada característica de calidad en el grado requerido.
- Identificar defectos y corregirlos.



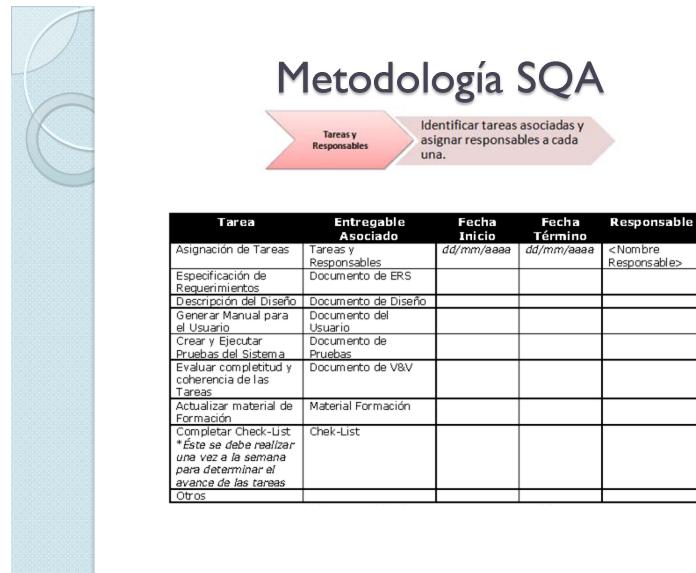
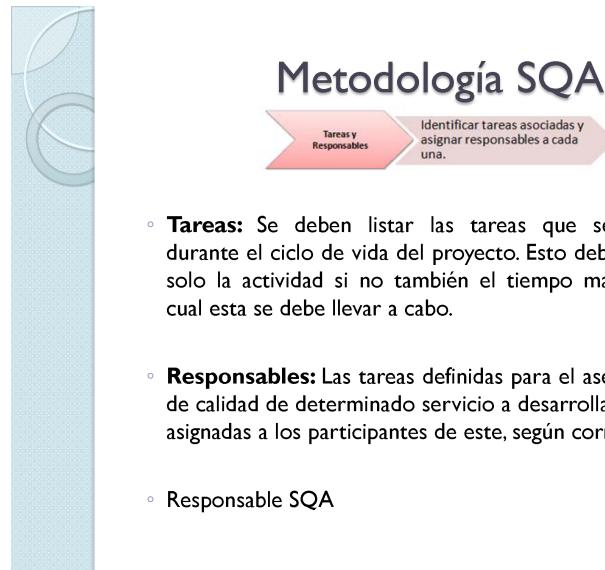
## Procesos de Calidad

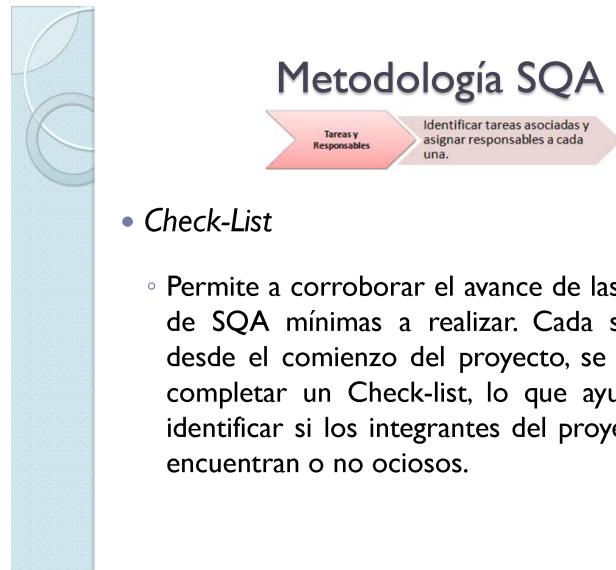
- Un proceso implica el uso de los recursos de una organización, para obtener algo de valor. Así, ningún producto puede fabricarse y ningún servicio puede suministrarse sin un proceso, y ningún proceso puede existir sin un producto o servicio. [Krajewski y Ritzman,]



## Metodología SQA







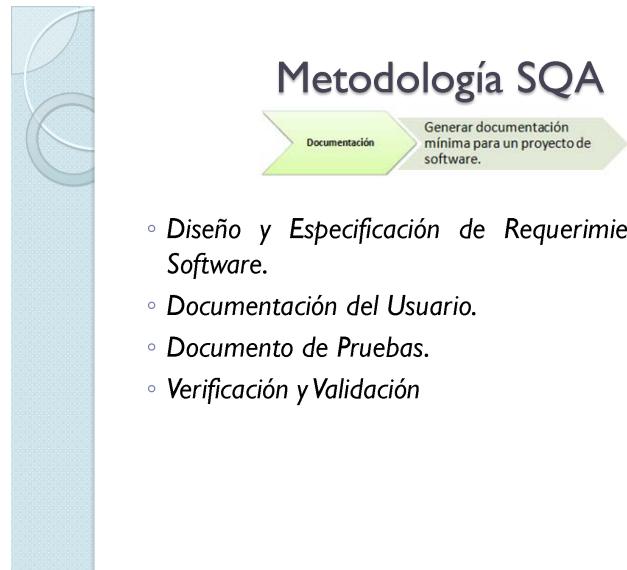
Metodología SQA

```

graph LR
    A[Tareas y Responsables] --> B[Identificar tareas asociadas y asignar responsables a cada una.]

```

Check-List: Evaluación Estado de Avance Tareas Asignadas			
1.Datos Evaluador			
Nombre Evaluador:	Fecha:		
2.Datos Tarea			
Responsable:	<input type="radio"/> SQA <input type="radio"/> SCM <input type="radio"/> Ad-hoc <input type="radio"/> Otro		
Detalle Tarea:	Porcentaje Completitud Tarea:		
3.Check-List	Si	No	Ocasional
Completitud			
¿La tarea asignada ha sido comenzada por el responsable asignado?			
¿El avance de la tarea es coherente con los plazos fijados?			
¿La tarea asignada ha sido terminada?			
Corrección			
¿La tarea se realizó según la descripción inicial de ésta?			
¿La tarea asignada presenta errores?			
¿La tarea asignada presenta inconsistencias de acuerdo al proyecto?			
¿La tarea asignada se realiza según los parámetros entregados?			
¿La tarea se realiza correctamente?			
4.Observaciones			





## Metodología SQA



- *Documentación del Usuario*
  - *Descripción del Sistema*
  - *Requisitos para el uso del Sistema*
  - *Funcionalidades*
  - *Observaciones*



## Metodología SQA



- *Documento de Pruebas*
  - *Diseño de los Casos de Prueba Manuales*
  - *Ejecución de los Casos de Prueba*
  - *Reporte de Resultados Finales*
  - *Resultados de las Pruebas de Aceptación*



## Metodología SQA



- **Verificación y Validación**

- **Descripción del Problema:**

Definir brevemente cual es el problema que se pretende solucionar con el desarrollo del sistema.



## Metodología SQA



- **Verificación y Validación**

- **Requerimientos Cumplidos:**

Indicar todos aquellos requerimientos del sistema que fueron completados de forma correcta.



## Metodología SQA



- **Verificación y Validación**

- Requerimientos ausentes o incompletos.  
Indicar aquellos requerimientos del sistema que no fueron completados y entregar una breve justificación de su ausencia.



## Metodología SQA



- **Verificación y Validación**

- **Check-List Asociados:**  
Adjuntar Check-List realizados semana a semana durante el desarrollo del proyecto.
    - Check-List SQA: Adjuntar Check-List asociados a las tareas de SQA planificadas.
    - Check-List SCM: Adjuntar Check-List asociados a las áreas de SCM planificadas.

The diagram illustrates the SQA methodology process. It begins with a light blue vertical bar on the left featuring a circular logo with a stylized 'C' or leaf pattern. To its right, the title 'Metodología SQA' is centered above a green double-headed arrow. The arrow points from left to right, with the word 'Documentación' on the left side and the descriptive text 'Generar documentación mínima para un proyecto de software.' on the right side.

- **Verificación y Validación**
  - Resumen Check-List SQA y SCM

Tarea	Completitud	Responsable	Fecha Estimada de Entrega	Fecha Real Entrega

The diagram illustrates the SQA methodology process. It begins with a light blue vertical bar on the left featuring a circular logo with a stylized 'C' or leaf pattern. To its right, the title 'Metodología SQA' is centered above a green double-headed arrow. The arrow points from left to right, with the word 'Documentación' on the left side and the descriptive text 'Generar documentación mínima para un proyecto de software.' on the right side.

- **Verificación y Validación**
  - Resumen y Análisis de la Completitud de Tareas.

Enlistar las tareas controladas por las metodologías de SQA y SCM durante el proceso de desarrollo. Entregue información sobre la completitud alcanzada por cada una y la correspondencia con la fecha de entrega estimada con la real.



## Metodología SQA

Recursos

Herramientas de apoyo a la metodología.

- *Mantis*
- *Documentación*
- *Personal*



## Metodología SQA

Formación

Material visual para educar a los integrantes de cada proyecto en cuanto a calidad.

- Es necesario realizar ciertas actividades que ayuden a entregar una justificación del uso de una metodología que apoye el aseguramiento de calidad de los productos.
- Es importante considerar el hecho de realizar modificaciones a este según cambien las necesidades en el Área de Desarrollo.



## C.2. Capacitación SCM



## ¿Qué es la Calidad de Software?

- La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario. [IEEE, Std 610-1900]



## Objetivos

- Comprobar si un producto posee una determinada característica de calidad en el grado requerido.
- Identificar defectos y corregirlos.



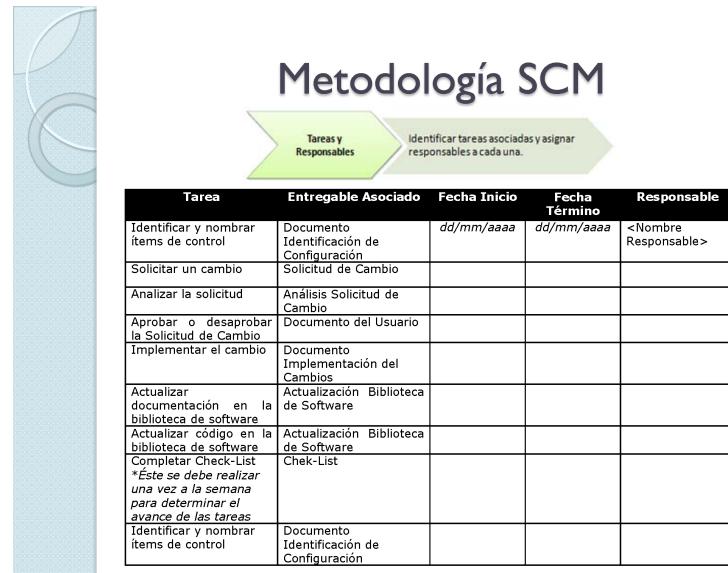
## Procesos de Calidad

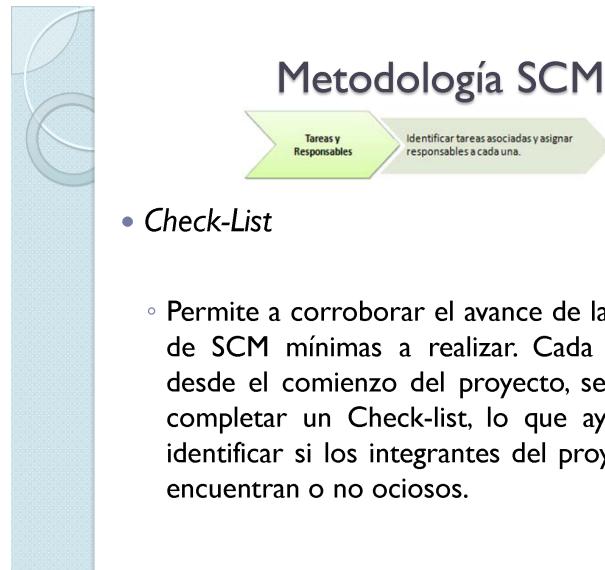
- Un proceso implica el uso de los recursos de una organización, para obtener algo de valor. Así, ningún producto puede fabricarse y ningún servicio puede suministrarse sin un proceso, y ningún proceso puede existir sin un producto o servicio. [Krajewski y Ritzman,]



## Metodología SCM



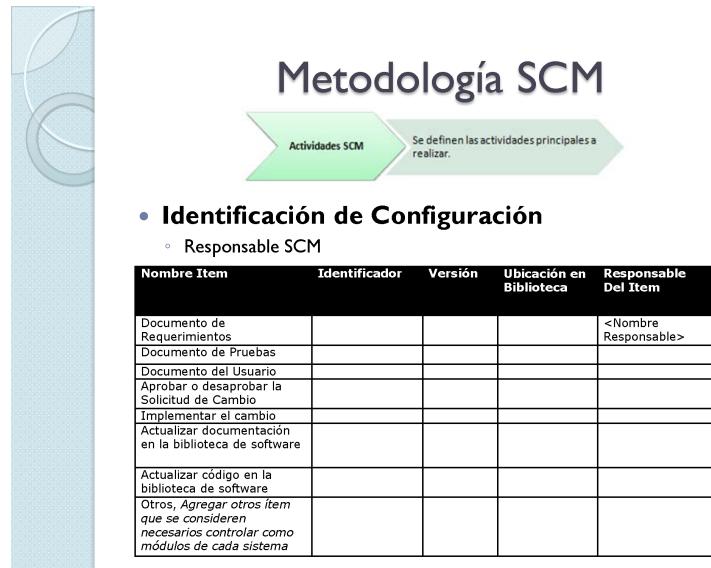


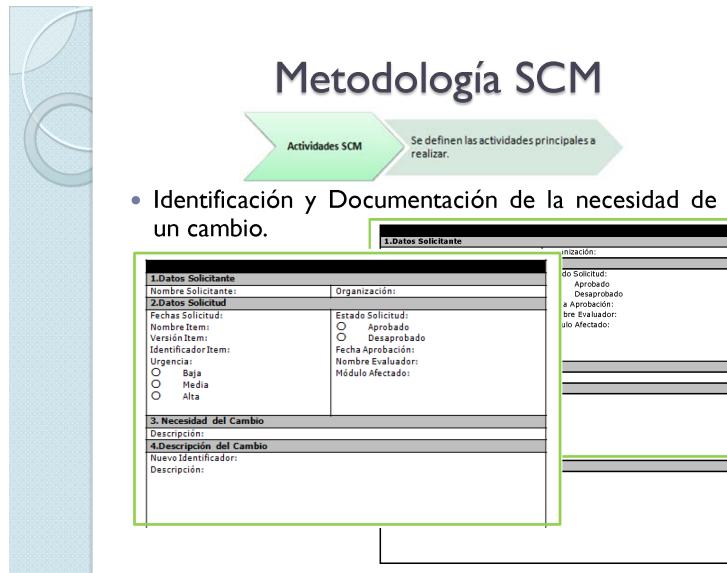
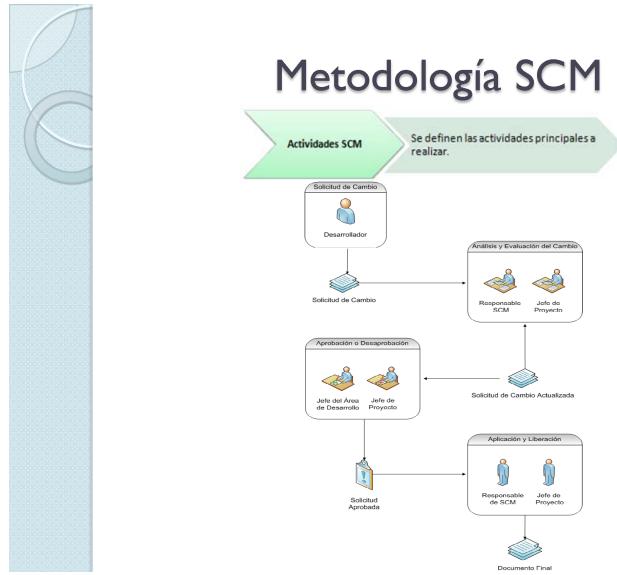


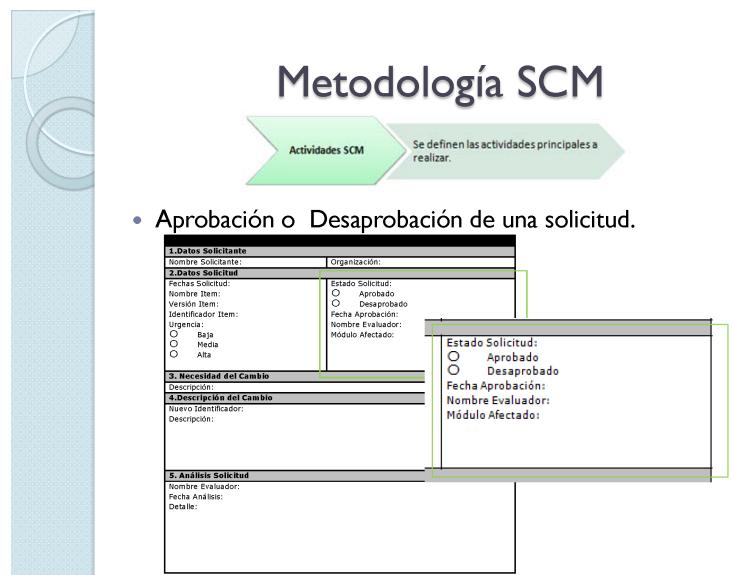
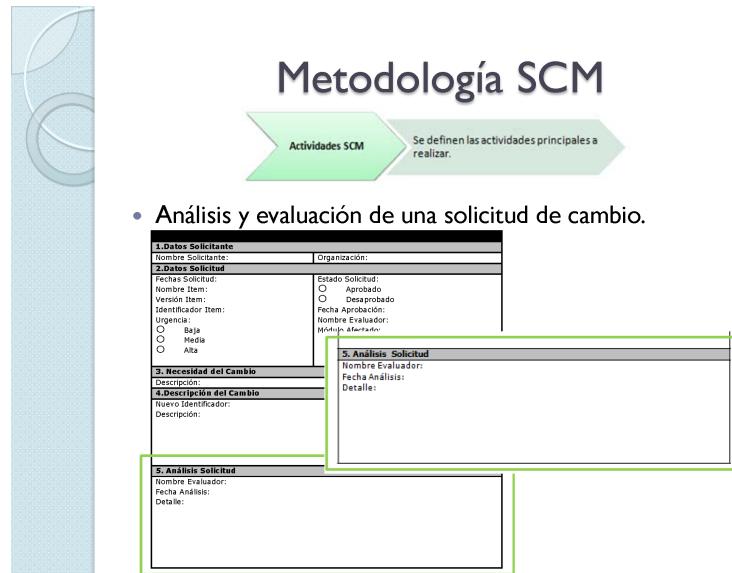
**Metodología SCM**

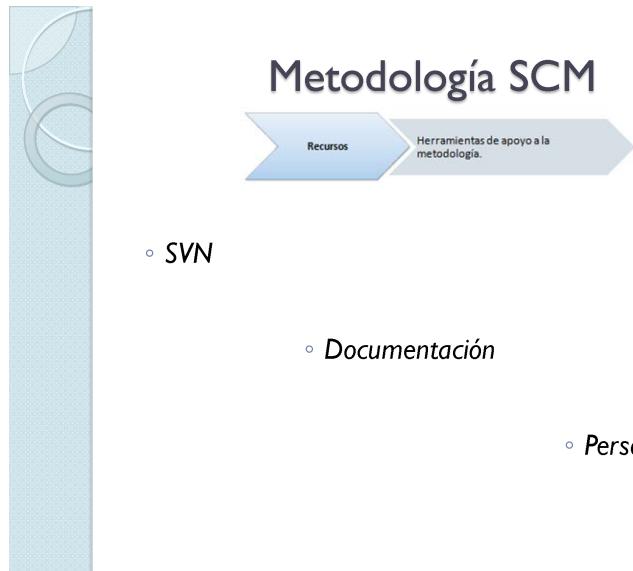
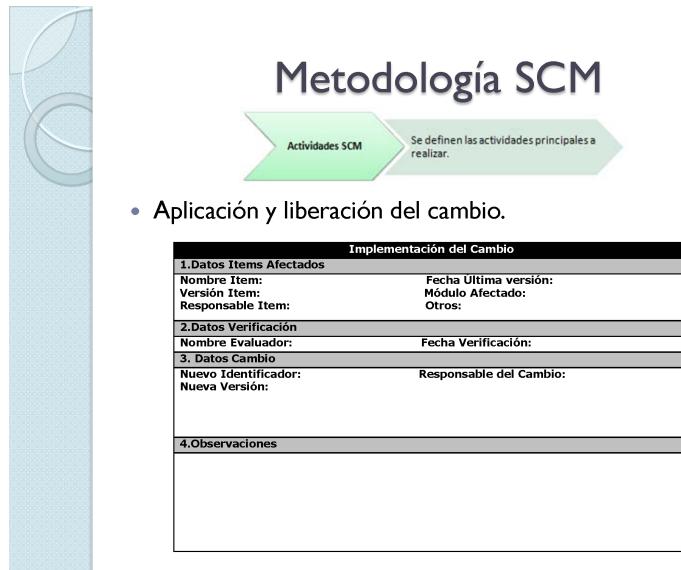
Tareas y Responsables → Identificar tareas asociadas y asignar responsables a cada una.

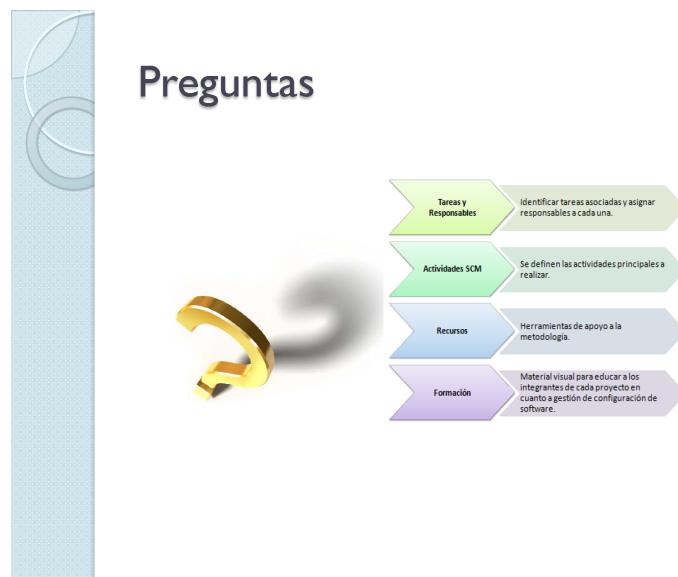
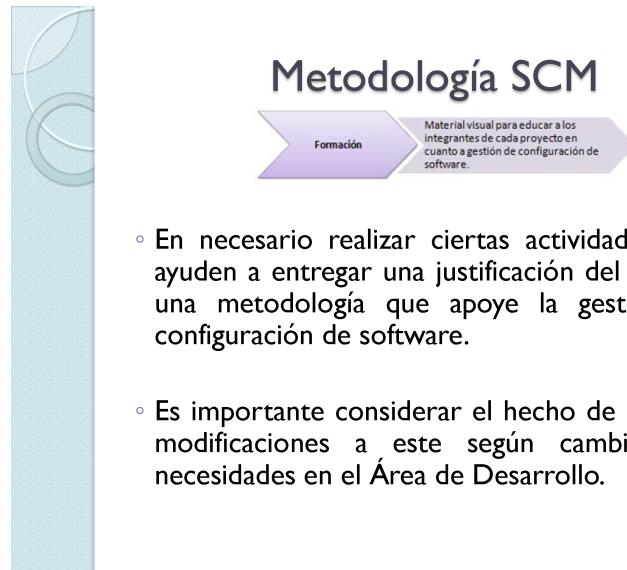
Check-List: Evaluación Estado de Avance Tareas Asignadas			
1.Datos Evaluador		Nombre Evaluador: _____ Fecha: _____	
2.Datos Tarea			
Responsable:	<input type="radio"/> SQA <input type="radio"/> SCM <input type="radio"/> Ad-hoc <input type="radio"/> Otro		
Detalle Tarea:	Porcentaje Completitud Tarea:		
3.Check-List	Si	No	Ocasional
Comprobidad			
„La tarea asignada ha sido comenzada por el responsable asignado?			
„El avance de la tarea es coherente con los plazos fijados?			
„La tarea asignada ha sido terminada?			
Correctitud			
„La tarea se realizó según la descripción inicial de ésta?			
„La tarea asignada presenta errores?			
„La tarea asignada presenta inconsistencias de acuerdo al proyecto?			
„La tarea asignada se realiza según los parámetros entregados?			
„La tarea se realiza correctamente?			
4.Observaciones			











# Apéndice D

## Template SQA

### D.1. Tareas y Responsables SQA



Área Desarrollo – Dirección  
Informática y Comunicaciones

---

“Tareas y Responsables Metodología SQA”

---

Dirección de Informática y Comunicaciones, Universidad de Valparaíso.



**Hoja de Control**

<b>Resumen Documento (máximo 2 líneas)</b>	Breve introducción al documento		
<b>Autor</b>	Quién realizó el documento (Responsable Documento)		
<b>Versión</b>	Ej: 1.0	<b>Fecha Versión</b>	
<b>Revisado/Validado por:</b>	Quién Revisa	<b>Fecha Revisión/Validación</b>	
<b>Aprobado por:</b>	Quién Valida	<b>Fecha Aprobación</b>	

**REGISTRO DE CAMBIOS**

<b>Versión</b>	<b>Causa del Cambio</b>	<b>Responsable del Cambio</b>	<b>Fecha del Cambio</b>
Ej: 2.0	Razón del cambio	Responsable de realizar el cambio	Ej: 10-09-2009

**Contactos**

<b>Nombre</b>	<b>Cargo en Proyecto</b>	<b>Dirección</b>	<b>anexo</b>	<b>Correo</b>
Eduardo Godoy	Encargado del proyecto	DISICO	5585	<a href="mailto:Eduardo.godoy@uv.cl">Eduardo.godoy@uv.cl</a>
Virginia Sanchez	Contacto	DIVACAD	7219	<a href="mailto:Virginia.samchez@uv.cl">Virginia.samchez@uv.cl</a>



**Área Desarrollo – Dirección  
Informática y Comunicaciones**

**1. Tareas y Responsables**

- Para definir las tareas y los responsables de cada una de ellas, que se realizarán para concretar la Metodología de SQA, es necesario completar la Tabla 1 con la fecha de inicio y término de la tarea, el entregable asociado y el responsable a cargo de cada una de ellas. Si se considera necesario agregar nuevas tareas con sus responsables, puede realizarse con previa autorización del Jefe de Proyecto.

Tarea	Entregable Asociado	Fecha Inicio	Fecha Término	Responsable
Asignación de Tareas	Tareas y Responsables	dd/mm/aaaa	dd/mm/aaaa	<Nombre Responsable>
Especificación de Requerimientos	Documento de ERS			
Descripción del Diseño	Documento de Diseño			
Generar Manual para el Usuario	Documento del Usuario			
Crear y Ejecutar Pruebas del Sistema	Documento de Pruebas			
Evaluar completitud y coherencia de las Tareas	Documento de V&V			
Actualizar material de Formación	Material Formación			
Completar Check-List <i>*Este se debe realizar una vez a la semana para determinar el avance de las tareas</i>	Chek-List			
Otros				

**Tabla 1 – Tabla de Tareas y Responsables**

**2. Observaciones**

- Agregar comentarios que se consideren necesarios, como alcances y explicaciones necesarias sobre las tareas y responsables asignados.

## D.2. Check-List

### Check-List

El Check-List debe ser completado por el responsable de SQA para cada tarea asignada para determinado proceso. La evaluación se debe realizar una vez a la semana según lo indique la calendarización.

- ✓ **Si:** El ítem se ha cumplido satisfactoriamente.
- ✓ **No:** El ítem no se ha cumplido satisfactoriamente.
- ✓ **Ocasional:** El ítem se cumple parcialmente.
- ✓ **No Aplica:** El ítem no aplica.

Check-List: Evaluación Estado de Avance Tareas Asignadas				
<b>1.Datos Evaluador</b>				
Nombre Evaluador:	Fecha:			
<b>2.Datos Tarea</b>				
Responsable:	<input type="radio"/> SQA <input type="radio"/> SCM <input type="radio"/> Ad-hoc <input type="radio"/> Otro			
Detalle Tarea:	Porcentaje Completitud Tarea:			
<b>3.Check-List</b>	Si	No	Ocasional	No Aplica
<b>Completitud</b>				
¿La tarea asignada ha sido comenzada por el responsable asignado?				
¿El avance de la tarea es coherente con los plazos fijados?				
¿La tarea asignada ha sido terminada?				
<b>Correctitud</b>				
¿La tarea se realizó según la descripción inicial de ésta?				
¿La tarea asignada presenta errores?				
¿La tarea asignada presenta inconsistencias de acuerdo al proyecto?				
¿La tarea asignada se realiza según los parámetros entregados?				
¿La tarea se realiza correctamente?				
<b>4.Observaciones</b>				

## **D.3. Diseño y Especificaciones de Requerimientos**



**Área Desarrollo – Dirección  
Informática y Comunicaciones**

---

**“Esquema Documentación Desarrollos  
Internos DISICO”**



**Área Desarrollo – Dirección  
Informática y Comunicaciones**

---

**Hoja de Control**

<b>Resumen Documento (máximo 2 líneas)</b>	Breve introducción al documento		
<b>Autor</b>	Quién realizó el documento (Responsable Documento)		
<b>Versión</b>	Ej: 1.0	<b>Fecha Versión</b>	
<b>Revisado/Validado por:</b>	Quién Revisa	<b>Fecha Revisión/Validación</b>	
<b>Aprobado por:</b>	Quién Valida	<b>Fecha Aprobación</b>	

**REGISTRO DE CAMBIOS**

<b>Versión</b>	<b>Causa del Cambio</b>	<b>Responsable del Cambio</b>	<b>Fecha del Cambio</b>
Ej: 2.0	Razón del cambio	Responsable de realizar el cambio	Ej: 10-09-2009

**Contactos**

<b>Nombre</b>	<b>Cargo en Proyecto</b>	<b>Dirección</b>	<b>anexo</b>	<b>Correo</b>
Eduardo Godoy	Encargado del proyecto	DISICO	5585	<a href="mailto:Eduardo.godoy@uv.cl">Eduardo.godoy@uv.cl</a>
Virginia Sanchez	Contacto	DIVACAD	7219	<a href="mailto:Virginia.samchez@uv.cl">Virginia.samchez@uv.cl</a>

---

Dirección de Informática y Comunicaciones, Universidad de Valparaíso.



**Área Desarrollo – Dirección  
Informática y Comunicaciones**

**1. Descripción General del Proyecto.**

- En este punto del documento se tiene que redactar una descripción genérica del proyecto que se va a implementar.

**2. Descripción General del Requerimiento.**

- En este punto se tiene que realizar la descripción del requerimiento que se va a implementar, además se tiene que anexar ya sea el correo electrónico que define el requerimiento o el acta en donde se tomo el acuerdo para realizar dicha implementación.

**3. Funciones y Descripción de éstas.**

- Acá se tiene que realizar el listado y descripción de cada una de las funciones que posea el sistema a implementar dado el requerimiento.

**4. Casos de uso.**

- Acá se tienen que realizar los casos de uso que describen el sistema. A continuación se muestra un ejemplo de casos de uso.

**Ejemplo:** Como parte del desarrollo del sistema de organización del próximo mundial de fútbol, a ud. le corresponde desarrollar el o los casos de usos asociados al ingreso de resultados de cada partido. Esto incluye ingresar el resultado, las estadísticas del juego y actualizar la ficha de cada jugador (tarjetas). Asumir que la planificación del mundial ya ha sido ingresada.

En esta pregunta vamos a trabajar bajo el supuesto que un usuario es el operador del sistema.

Además cabe mencionar que en esta pregunta identificamos tres casos de uso, los cuales se van a explicar detalladamente a continuación.

Identificación de los casos de uso del sistema:

- Ingresar los resultados



**Área Desarrollo – Dirección  
Informática y Comunicaciones**

**Caso de uso: Ingresar resultados.**  
**Descripción expandida:**

Caso de uso: Ingresar resultados  
 Actores: usuario (iniciador)  
 Propósito: Ingresar los resultados de cada equipo y de cada partido.  
 Precondición: el sistema de organización debe estar en funcionamiento  
 Resumen: El usuario solicita el registro de los resultados de cada partido jugado, para actualizar dicho registro de cada equipo, en cada partido jugado.

**Curso normal de eventos:**

<b>Acciones de los actores</b>	<b>Respuesta del sistema</b>
1º Este caso de uso comienza cuando el usuario accede a ingresar los resultados de los partidos 3º El usuario ingresa los nombres de cada equipo, para su posterior registro del resultado final del partido 6º El usuario ingresa los resultados correspondientes del partido ya finalizado 8º El usuario verifica que los datos ingresados sean los correctos	2º El sistema solicita los nombres de los equipos correspondientes al partido jugado 4º El sistema captura y valida los datos de cada equipo 5º El sistema accede a ingresar los resultados 7º El sistema captura y valida los datos ingresados del partido 9º Registrar los datos del partido

**Líneas alternativas:**

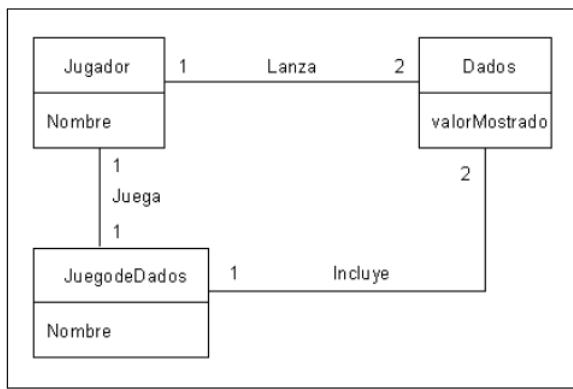
- L.3: El usuario ingresa nombre de equipo inválido, indicar error.
- L.6: El usuario ingresa datos erróneos al sistema, indicar error.
- L.8: El usuario verifica que algún dato es incorrecto, ingresar nuevamente los datos.

- **Primer hito: Documento de Aprobación del cliente.**



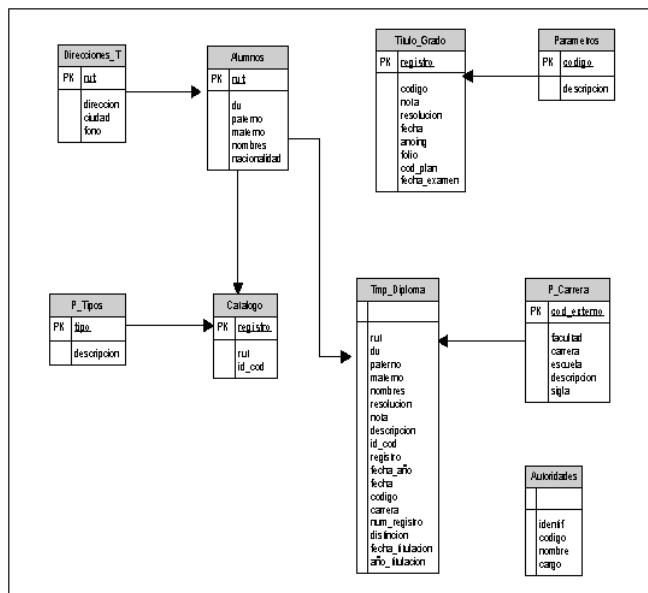
**5. Modelo Conceptual.**

- Acá se tiene que realizar el modelo conceptual que abarque todo el proyecto.  
Un modelo conceptual muestra gráficamente los conceptos (clases de objetos), los atributos y las asociaciones más importantes del dominio del problema. A continuación se muestra un ejemplo:



**6. Modelo de Datos (Particular).**

- Acá se tiene que realizar el modelo de datos que se utilizó para la realización de la implementación (Si no se tiene el modelo completo, incluir las tablas que se utilizan para el proyecto). A continuación se muestra un ejemplo de modelo de datos.

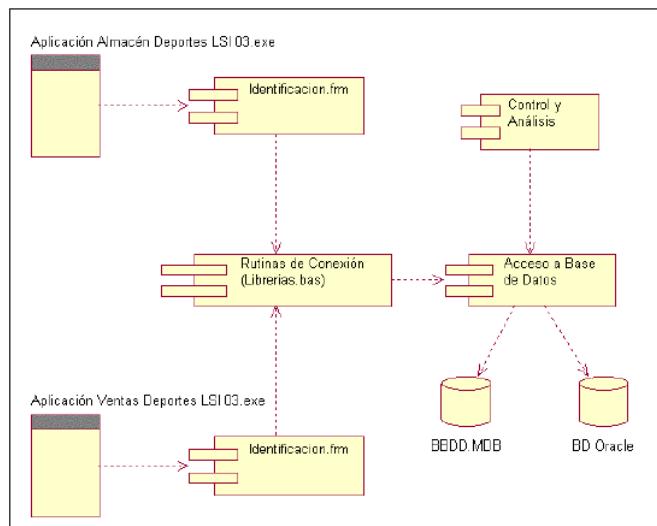


**7. Listado y breve descripción de Procedimientos almacenados.**

- En este punto se tiene que realizar el listado de los procedimientos almacenados que se utilizan para la realización del proyecto y en qué instantes son usados. Además se tiene que realizar una breve descripción de éste.

**8. Modelo de componentes.**

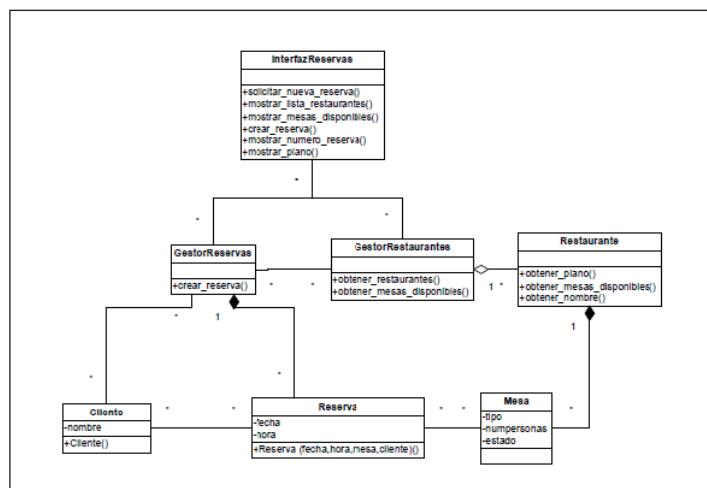
- En este punto se tiene que identificar cada uno de los elementos físicos que intervienen en el sistema.  
Este modelo describe los elementos físicos del sistema y sus relaciones, además los componentes se definen como la representación de todo tipo de elemento de software que intervienen en la creación del proyecto. A continuación se muestra un ejemplo.



Dirección de Informática y Comunicaciones, Universidad de Valparaíso.

**9. Diagrama de clases.**

- Acá se tiene que realizar el diagrama con las principales clases que se utilizan en el proyecto realizado.  
Un diagrama de clases describe gráficamente las especificaciones de las clases, los atributos de la clase y los métodos de ésta. A continuación se muestra un ejemplo de un diagrama de clases.



- Segundo hito: Documento final diseño.

## D.4. Documento del Usuario



Área Desarrollo – Dirección  
Informática y Comunicaciones

---

"Documento del Usuario"



### Hoja de Control

<b>Resumen Documento (máximo 2 líneas)</b>	Breve introducción al documento		
<b>Autor</b>	Quién realizó el documento (Responsable Documento)		
<b>Versión</b>	Ej: 1.0	<b>Fecha Versión</b>	
<b>Revisado/Validado por:</b>	Quién Revisa	<b>Fecha Revisión/Validación</b>	
<b>Aprobado por:</b>	Quién Valida	<b>Fecha Aprobación</b>	

### REGISTRO DE CAMBIOS

<b>Versión</b>	<b>Causa del Cambio</b>	<b>Responsable del Cambio</b>	<b>Fecha del Cambio</b>
Ej: 2.0	Razón del cambio	Responsable de realizar el cambio	Ej: 10-09-2009

### Contactos

<b>Nombre</b>	<b>Cargo en Proyecto</b>	<b>Dirección</b>	<b>anexo</b>	<b>Correo</b>
Eduardo Godoy	Encargado del proyecto	DISICO	5585	<a href="mailto:Eduardo.godoy@uv.cl">Eduardo.godoy@uv.cl</a>
Virginia Sanchez	Contacto	DIVACAD	7219	<a href="mailto:Virginia.samchez@uv.cl">Virginia.samchez@uv.cl</a>



**Área Desarrollo – Dirección  
Informática y Comunicaciones**

---

**1. Introducción**

- Indicar el objetivo principal del sistema y cuál es el problema que resuelve.

**2. Requisitos para el uso del Sistema**

- Si corresponde, indicar cuáles son los requisitos fundamentales para el uso exitoso y correcto del sistema, tales como sistema operativo, navegador, entre otros.

**3. Instalación**

- Si corresponde, indicar cuál es el procedimiento para realizar la instalación del sistema. Se recomienda apoyar con imágenes.

**4. Funcionalidades**

- Explicar de forma breve las principales funcionalidades del sistema, indicando el procedimiento a seguir. Es importante apoyar con las interfaces principales del sistema.

**5. Ayuda**

- Indicar al usuario como acceder a la ayuda del sistema. Si éste no la tiene indicar en ésta sección la ayuda que se crea necesaria para entregar al usuario.

**6. Observaciones**

- Agregar comentarios que se consideren necesarios, como alcances y explicaciones necesarias sobre el funcionamiento del sistema

## D.5. Verificación y Validación



Área Desarrollo – Dirección  
Informática y Comunicaciones

---

“Verificación y Validación”



**Área Desarrollo – Dirección  
Informática y Comunicaciones**

---

**Hoja de Control**

<b>Resumen Documento (máximo 2 líneas)</b>	Breve introducción al documento		
<b>Autor</b>	Quién realizó el documento (Responsable Documento)		
<b>Versión</b>	Ej: 1.0	<b>Fecha Versión</b>	
<b>Revisado/Validado por:</b>	Quién Revisa	<b>Fecha Revisión/Validación</b>	
<b>Aprobado por:</b>	Quién Valida	<b>Fecha Aprobación</b>	

**REGISTRO DE CAMBIOS**

<b>Versión</b>	<b>Causa del Cambio</b>	<b>Responsable del Cambio</b>	<b>Fecha del Cambio</b>
Ej: 2.0	Razón del cambio	Responsable de realizar el cambio	Ej: 10-09-2009

**Contactos**

<b>Nombre</b>	<b>Cargo en Proyecto</b>	<b>Dirección</b>	<b>anexo</b>	<b>Correo</b>
Eduardo Godoy	Encargado del proyecto	DISICO	5585	<a href="mailto:Eduardo.godoy@uv.cl">Eduardo.godoy@uv.cl</a>
Virginia Sanchez	Contacto	DIVACAD	7219	<a href="mailto:Virginia.samchez@uv.cl">Virginia.samchez@uv.cl</a>



**1. Definición del Problema.**

- Definir brevemente cual es el problema que se pretende solucionar con el desarrollo del sistema.

**2. Requerimientos Cumplidos.**

- Indicar todos aquellos requerimientos del sistema que fueron completados de forma correcta

**3. Requerimientos ausentes o incompletos.**

- Indicar aquellos requerimientos del sistema que no fueron completados y entregar una breve justificación de su ausencia.

**4. Check-List Asociados.**

- Adjuntar Check-List realizados semana a semana durante el desarrollo del proyecto.

**4.1. Check-List SQA.**

- Adjuntar Check-List asociados a las tareas de SQA planificadas.

**4.2. Check-List SCM.**

- Adjuntar Check-List asociados a las áreas de SCM planificadas.

**5. Resumen y Análisis de la Completitud de Tareas.**

- Enlistar las tareas controladas por las metodologías de SQA y SCM durante el proceso de desarrollo. Entregue información sobre la completitud alcanzada por cada una y la correspondencia con la fecha de entrega estimada con la real. La fecha de creación de éste documento se encuentra planificada en el documento de Tareas y Responsable. Esta fecha debe ser respetada aunque exista atrasos en el proyecto.



**Área Desarrollo – Dirección  
Informática y Comunicaciones**

**5.1. Resumen Check-List SQA.**

- Posterior a completar la tabla resumen de las tareas realizadas, incorpore un análisis final del avance del proyecto y cumplimiento de las tareas realizadas asociadas a las tareas de SQA planificadas.

Tarea	Completitud	Responsable	Fecha Estimada de Entrega	Fecha Real Entrega

**5.2. Resumen Check-List SCM**

- Posterior a completar la tabla resumen de las tareas realizadas, incorpore un análisis final del avance del proyecto y cumplimiento de las tareas realizadas asociadas a las tareas de SCM planificadas.

Tarea	Completitud	Responsable	Fecha Estimada de Entrega	Fecha Real Entrega

**6. Conclusión**

- Entregar una conclusión general del desempeño entregado en el proyecto, asociado al documento, por cada uno de los integrantes de éste. Indique cual es el análisis que se puede realizar con la información reunida.

## D.6. Pruebas



**Área Desarrollo – Dirección  
Informática y Comunicaciones**

---

**“Documento de Pruebas”**



**Área Desarrollo – Dirección  
Informática y Comunicaciones**

**Hoja de Control**

<b>Resumen Documento (máximo 2 líneas)</b>	Breve introducción al documento		
<b>Autor</b>	Quién realizó el documento (Responsable Documento)		
<b>Versión</b>	Ej: 1.0	<b>Fecha Versión</b>	
<b>Revisado/Validado por:</b>	Quién Revisa	<b>Fecha Revisión/Validación</b>	
<b>Aprobado por:</b>	Quién Valida	<b>Fecha Aprobación</b>	

**REGISTRO DE CAMBIOS**

<b>Versión</b>	<b>Causa del Cambio</b>	<b>Responsable del Cambio</b>	<b>Fecha del Cambio</b>
Ej: 2.0	Razón del cambio	Responsable de realizar el cambio	Ej: 10-09-2009

**Contactos**

<b>Nombre</b>	<b>Cargo en Proyecto</b>	<b>Dirección</b>	<b>anexo</b>	<b>Correo</b>
Eduardo Godoy	Encargado del proyecto	DISICO	5585	<a href="mailto:Eduardo.godoy@uv.cl">Eduardo.godoy@uv.cl</a>
Virginia Sanchez	Contacto	DIVACAD	7219	<a href="mailto:Virginia.samchez@uv.cl">Virginia.samchez@uv.cl</a>



**Área Desarrollo – Dirección  
Informática y Comunicaciones**

**1. Diseño de casos de prueba manuales.**

- El objetivo del diseño de las pruebas manuales, es que tengan la mayor probabilidad de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y tiempo posible.

- **Identificar Casos de prueba**

Los pasos a seguir al momento de diseñar los casos de prueba:

- a. Definir las entradas de los casos de prueba.
- b. Identificar las salidas de los casos de prueba.

Cada uno de los casos de prueba se deberá especificar en forma detallada con la siguiente tabla:

<b>Casos de Prueba</b>			
<b>Datos iniciales</b>			
Fecha		Número del caso de prueba	
Característica			
<b>Información del caso de prueba</b>			
Descripción		Objetivo	
Datos de entrada		Resultados esperados	
<b>Procedimiento del caso de prueba</b>			
Pasos a seguir	Nº	Secuencia	
	1		
	2		
	3		
Condiciones externas			
<b>Resultados</b>			
<b>Resultado Obtenidos</b>		<b>Complejidad</b>	
		Aprobado:	
		No aprobado:	
<b>Seriedad de la falla</b>			
		Grave	
		Menor	
		Sin errores identificados	
<b>Observaciones</b>			

**Dirección de Informática y Comunicaciones, Universidad de Valparaíso.**



- **Procedimiento de prueba.**

Para esta sección se deberá definir lo siguiente:  
a. Definir la secuencia de los casos de pruebas.

**1. Diseño de casos de prueba automáticos.**

- El objetivo del diseño de las pruebas automáticas, es automatizar los casos de prueba definidos en el punto anterior. En esta sección se deberá especificar la implementación de cada uno de los casos de prueba automatizados, es decir los métodos setUp () y tearDown () por cada uno de los módulos de prueba.

**2. Ejecución de los casos de prueba.**

- Concluidos los dos primeros procesos de diseño de casos de prueba manual y automáticos, además de la definición de la secuencia a realizar los casos de pruebas, corresponde ejecutar cada uno de estos casos de prueba. Se deberá realizar:

- **Pruebas de Unidad.**

El objetivo es encontrar errores a nivel modular del sistema, es decir probar cada módulo de manera independiente.

- **Pruebas de integración.**

El objetivo es comprobar que las interfaces entre módulos funcionen correctamente.

- **Pruebas de aceptación.**

El objetivo es validar el software con el cliente, por medio de una reunión, en la cual se presentaran los requerimientos, escenarios y pantallas al cliente, finalizando el proceso con la aceptación del software.

Es importante mencionar que se deberán ejecutar en una primera instancia la totalidad de las pruebas unitarias para luego realizar las pruebas de integración y finalizando el proceso con las pruebas de aceptación.

---



**Área Desarrollo – Dirección  
Informática y Comunicaciones**

**3. Reporte de cambios.**

- El objetivo principal de esta sección del documento es especificar cuáles fueron los principales cambios realizados tanto en la implementación como en la arquitectura. Se deben especificar los cambios por medio de la siguiente tabla:

Reporte de cambios. Cambios Arquitectónicos.			
Subsistemas	Nº	Nombre	Descripción
Módulos	Nº	Nombre	Descripción
Componentes	Nº	Nombre	Descripción
Clases	Nº	Nombre	Descripción

**4. Reporte de resultados finales.**

- Esta actividad de la fase de pruebas que se realiza en un producto de software se refiere a documentar los resultados de la comparación de ejecución de las pruebas manuales y automáticas. Se deberán documentar los resultados por medio de la siguiente tabla:

Reporte de resultados.				
Aprobación.				
Nº del caso de prueba	Subsistema	Módulo	Componente	Clases
Reprobados.				
Nº del caso de prueba	Subsistema	Módulo	Componente	Clases

Dirección de Informática y Comunicaciones, Universidad de Valparaíso.



**Área Desarrollo – Dirección  
Informática y Comunicaciones**

---

Se define como subsistema, módulo, componente y clase aprobadas al conjunto de funcionalidades que tienen el mismo resultado como salida de los casos de prueba manual y automáticos. Los demás llámese reprobados.

**5. Resultado de las pruebas de aceptación.**

- En esta última actividad de la fase de pruebas se presenta al cliente cada una de las funcionalidades del software para su posterior aceptación por parte del cliente. Los resultados de este tipo de pruebas se documentan completando la siguiente tabla.

Pruebas de Aceptación			Usuario final	
Equipo de desarrollo	Requerimientos	Escenario	Cliente	Aprobado
1				
2				
N				
1				
2				
N				
1				
2				
N				
<b>Firma del Cliente</b>				

## D.7. Formación



### • Gestión de Calidad de Software

Formación Metodologías de SQA y SCM



### ¿Qué es la Calidad de Software?

- La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario. [IEEE, Std 610-1900]



## Objetivos

- Comprobar si un producto posee una determinada característica de calidad en el grado requerido.
- Identificar defectos y corregirlo.

## Clave de la Calidad

- Para convertir una empresa en una organización con calidad, debe partirse desde lo más primordial de su ser: su equipo humano.





### Clave de la Calidad

- El trabajar con valores y principios, que cumplan lo que comprometen tanto los empleados como las empresas, respetando el medio ambiente y la comunidad, asegurando que todos los productos y servicios funcionen desde el primer intento.



### Procesos de Calidad

- Un proceso implica el uso de los recursos de una organización, para obtener algo de valor. Así, ningún producto puede fabricarse y ningún servicio puede suministrarse sin un proceso, y ningún proceso puede existir sin un producto o servicio. [Krajewski y Ritzman.]



## Aseguramiento de Calidad

- Conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto (software) satisfará los requisitos dados de calidad.



## Beneficios de SQA

- Se detectan problemas rápidamente.
- Se crean y se siguen estándares de trabajo.
- Se recomiendan métodos para realizar el trabajo.
- Se evita incurrir en costos innecesarios.
- Se planea la calidad.



## Gestión de Configuración de Software

- Conjunto de técnicas y actividades de carácter operativo, utilizadas para verificar los requisitos relativos a la calidad, centradas en mantener bajo control el proceso de desarrollo y eliminar las causas de los defectos en las diferentes fases del ciclo de vida.



## Beneficios de SCM

- ◎ Mejora la comunicación, evita duplicar esfuerzos.
- ◎ Asegura que se construya el sistema correcto.
- ◎ Agiliza la identificación de problemas y corrección de errores.
- ◎ Ayuda a identificar de manera precisa la versión sobre que se realizarán los cambios.
- ◎ Mejora el aseguramiento de calidad.



### Certificación de Calidad

- Procesos de calidad como SQA y SCM son un paso importante para alcanzar una certificación de calidad.



### Certificación de Calidad

- La Certificación de Producto es un proceso mediante el cual se garantiza la calidad y/o las características de un producto final según lo establecido en una norma específica u otros documentos preestablecidos.



## Certificación de Calidad

### ¿Para que sirve?

- Comunicar la Calidad de los Productos y Servicios certificados.
- Aumentar la confianza de nuestros consumidores finales.
- Diferenciarse de la competencia.
- Garantizar a nuestros clientes una seguridad.

## Preguntas



## **Apéndice E**

### **Template SCM**

La Metodología de SCM propone una serie de documentos para su correcto uso.

## E.1. Tareas y Responsables SCM



Universidad  
de Valparaíso  
CHILE

Área Desarrollo – Dirección  
Informática y Comunicaciones

---

"Tareas y Responsables Metodología SCM"

---

Dirección de Informática y Comunicaciones, Universidad de Valparaíso.



**Área Desarrollo – Dirección  
Informática y Comunicaciones**

**Hoja de Control**

<b>Resumen Documento (máximo 2 líneas)</b>	Breve introducción al documento		
<b>Autor</b>	Quién realizó el documento (Responsable Documento)		
<b>Versión</b>	Ej: 1.0	<b>Fecha Versión</b>	
<b>Revisado/Validado por:</b>	Quién Revisa	<b>Fecha Revisión/Validación</b>	
<b>Aprobado por:</b>	Quién Valida	<b>Fecha Aprobación</b>	

**REGISTRO DE CAMBIOS**

<b>Versión</b>	<b>Causa del Cambio</b>	<b>Responsable del Cambio</b>	<b>Fecha del Cambio</b>
Ej: 2.0	Razón del cambio	Responsable de realizar el cambio	Ej: 10-09-2009

**Contactos**

<b>Nombre</b>	<b>Cargo en Proyecto</b>	<b>Dirección</b>	<b>anexo</b>	<b>Correo</b>
Eduardo Godoy	Encargado del proyecto	DISICO	5585	<a href="mailto:Eduardo.godoy@uv.cl">Eduardo.godoy@uv.cl</a>
Virginia Sanchez	Contacto	DIVACAD	7219	<a href="mailto:Virginia.samchez@uv.cl">Virginia.samchez@uv.cl</a>



**Área Desarrollo – Dirección  
Informática y Comunicaciones**

**1. Tareas y Responsables**

- Para definir las tareas y los responsables de cada una de ellas, que se realizarán para concretar la Metodología de SCM, es necesario completar la Tabla 1 con la fecha de inicio y término de la tarea, el entregable asociado y el responsable a cargo de cada una de ellas. Si se considera necesario agregar nuevas tareas con sus responsables, puede realizarse con previa autorización del Jefe de Proyecto.

Tarea	Entregable Asociado	Fecha Inicio	Fecha Término	Responsable
Identificar y nombrar ítems de control	Documento Identificación de Configuración	dd/mm/aaaa	dd/mm/aaaa	<Nombre Responsable>
Solicitar un cambio	Solicitud de Cambio			
Analizar la solicitud	Ánalisis Solicitud de Cambio			
Aprobar o desaprobar la Solicitud de Cambio	Documento del Usuario			
Implementar el cambio	Documento Implementación del Cambios			
Actualizar documentación en la biblioteca de software	Actualización Biblioteca de Software			
Actualizar código en la biblioteca de software	Actualización Biblioteca de Software			
Completar Check-List <i>*Este se debe realizar una vez a la semana para determinar el avance de las tareas</i>	Chek-List			
Identificar y nombrar ítems de control	Documento Identificación de Configuración			

**Tabla 1 – Tabla de Tareas y Responsables**

**2. Observaciones**

- Agregar comentarios que se consideren necesarios, como alcances y explicaciones necesarias sobre las tareas y responsables asignados.

## **E.2. Identificación de Configuración**



**Área Desarrollo – Dirección  
Informática y Comunicaciones**

---

**"Identificación de Configuración"**



**Área Desarrollo – Dirección  
Informática y Comunicaciones**

**Hoja de Control**

<b>Resumen Documento (máximo 2 líneas)</b>	Breve introducción al documento		
<b>Autor</b>	Quién realizó el documento (Responsable Documento)		
<b>Versión</b>	Ej: 1.0	<b>Fecha Versión</b>	
<b>Revisado/Validado por:</b>	Quién Revisa	<b>Fecha Revisión/Validación</b>	
<b>Aprobado por:</b>	Quién Valida	<b>Fecha Aprobación</b>	

**REGISTRO DE CAMBIOS**

<b>Versión</b>	<b>Causa del Cambio</b>	<b>Responsable del Cambio</b>	<b>Fecha del Cambio</b>
Ej: 2.0	Razón del cambio	Responsable de realizar el cambio	Ej: 10-09-2009

**Contactos**

<b>Nombre</b>	<b>Cargo en Proyecto</b>	<b>Dirección</b>	<b>anexo</b>	<b>Correo</b>
Eduardo Godoy	Encargado del proyecto	DISICO	5585	<a href="mailto:Eduardo.godoy@uv.cl">Eduardo.godoy@uv.cl</a>
Virginia Sanchez	Contacto	DIVACAD	7219	<a href="mailto:Virginia.samchez@uv.cl">Virginia.samchez@uv.cl</a>



**Área Desarrollo – Dirección  
Informática y Comunicaciones**

**1. Identificación de Configuración.**

- Para identificar cada ítem de control se deben describir en la Tabla 1 llenando cada uno de los campos solicitados.

Nombre Item	Identificador	Versión	Ubicación en Biblioteca	Responsable Del Item
Documento de Requerimientos				<Nombre Responsable>
Documento de Pruebas				
Documento del Usuario				
Aprobar o desaprobar la Solicitud de Cambio				
Implementar el cambio				
Actualizar documentación en la biblioteca de software				
Actualizar código en la biblioteca de software				
Otros, Agregar otros ítem que se consideren necesarios controlar como módulos de cada sistema				

**Tabla 1 – Ítems de Configuración.**

**2. Observaciones.**

- Agregar comentarios que se consideren necesarios, como alcances y explicaciones necesarias sobre los ítems definidos.

---

Dirección de Informática y Comunicaciones, Universidad de Valparaíso.

## E.3. Configuración de Control

Para la configuración de Control se necesitan dos documentos principales descritos en la sección 7.1 y 7.2.

### E.3.1. Solicitud de Cambio

#### Solicitud de Cambio

La solicitud de cambio debe ser llenada por quien haya recibido la inquietud de modificar algún ítem del proyecto. Se debe completar cada campo presente en la tabla. Posterior a esto se debe realizar el análisis de la solicitud por el responsable de SCM y el Jefe de Proyecto completando el último campo de la solicitud.

<b>Solicitud de Cambio</b>	
<b>1. Datos Solicitud</b>	
Nombre Solicitante:	Fechas Solicitud:
Organización:	Nombre Item:
Urgencia: <input type="radio"/> Baja <input type="radio"/> Media <input type="radio"/> Alta	Versión Item:  Identificador Item:
<b>2. Necesidad del Cambio</b>	
Descripción:	
<b>4. Descripción del Cambio</b>	
Nuevo Identificador:  Descripción:	
<b>5. Análisis Solicitud</b>	
Nombre Evaluador:	Fecha Análisis:
Detalle:	
<b>6. Aprobación</b>	
Estado Solicitud: <input type="radio"/> Aprobado <input type="radio"/> Desaprobado	Nombre Evaluador:  Módulo Afectado:
Fecha Aprobación:	

### **E.3.2. Implementación del Cambio**



**Área Desarrollo – Dirección  
Informática y Comunicaciones**

---

**“Implementación del Cambio”**



**Área Desarrollo – Dirección  
Informática y Comunicaciones**

**Hoja de Control**

<b>Resumen Documento (máximo 2 líneas)</b>	Breve introducción al documento		
<b>Autor</b>	Quién realizó el documento (Responsable Documento)		
<b>Versión</b>	Ej: 1.0	<b>Fecha Versión</b>	
<b>Revisado/Validado por:</b>	Quién Revisa	<b>Fecha Revisión/Validación</b>	
<b>Aprobado por:</b>	Quién Valida	<b>Fecha Aprobación</b>	

**REGISTRO DE CAMBIOS**

<b>Versión</b>	<b>Causa del Cambio</b>	<b>Responsable del Cambio</b>	<b>Fecha del Cambio</b>
Ej: 2.0	Razón del cambio	Responsable de realizar el cambio	Ej: 10-09-2009

**Contactos**

<b>Nombre</b>	<b>Cargo en Proyecto</b>	<b>Dirección</b>	<b>anexo</b>	<b>Correo</b>
Eduardo Godoy	Encargado del proyecto	DISICO	5585	<a href="mailto:Eduardo.godoy@uv.cl">Eduardo.godoy@uv.cl</a>
Virginia Sanchez	Contacto	DIVACAD	7219	<a href="mailto:Virginia.samchez@uv.cl">Virginia.samchez@uv.cl</a>



**Área Desarrollo – Dirección  
Informática y Comunicaciones**

**1. Verificación e Implementación del Cambio.**

- Para la implementación del cambio solicitado se deben llenar los siguientes campos.

<b>Implementación del Cambio</b>	
<b>1.Datos Items Afectados</b>	
Nombre Item:	Fecha Última versión:
Versión Item:	Módulo Afectado:
Responsable Item:	Otros:
<b>2.Datos Verificación</b>	
Nombre Evaluador:	Fecha Verificación:
<b>3. Datos Cambio</b>	
Nuevo Identificador:	Responsable del Cambio:
Nueva Versión:	
<b>4.Observaciones</b>	

**2. Solicitud de Cambio.**

- Para completar el documento es necesario adjuntar la solicitud de cambio asociada.

# Bibliografía

- [1] Apache - Subversion. Último acceso 01 de Abril 2011. <http://subversion.apache.org/>.
- [2] Concurrent Versions System. Último acceso 10 de Abril 2011. <http://www.cvshome.org/>.
- [3] Ian Sommerville. *Ingeniería del Software*. Pearson Addison Wesley, 2005.
- [4] IEEE. *Plan de Aseguramiento de Calidad de Software*. Instituto de Ingenieros Eléctricos y Electrónicos, 1989.
- [5] IEEE. *Glosario de Terminología de Ingeniería de Software*. Instituto de Ingenieros Eléctricos y Electrónicos, 1990.
- [6] IEEE. *Verificación y Validación del Software*. Instituto de Ingenieros Eléctricos y Electrónicos, 1998.
- [7] IEEE. *Plan de Gestión de Configuración de Software*. Instituto de Ingenieros Eléctricos y Electrónicos, 2005.
- [8] ISO. *Terminología Gestión de Calidad*. International Organization for Standardization, 1986.
- [9] ISO. *Sistemas de Gestión de Calidad. Directrices para la Gestión de la Configuración*. International Organization for Standardization, 2003.
- [10] ISO. *Sistemas de Gestión de Calidad - Fundamentos y Vocabulario*. International Organization for Standardization, 2008.
- [11] ISO. *Sistemas de Gestión de Calidad - Requisitos*. International Organization for Standardization, 2008.
- [12] ISO. *Sistemas de Gestión de Calidad - Directrices para la mejora del desempeño*. International Organization for Standardization, 2009.
- [13] Mantis. Último acceso 14 de marzo 2011. <http://www.mantisbt.org>.

- [14] Maria Cruz Canaveras Tribaldos. Herramienta para la gestión de cambios. Rational ClearQuest.
- [15] Mary Beth Chrissis, Mike Konrad y Sandy Shrum. *Guía para la integración de procesos y la mejora de productos*, 2 edition, 2009.
- [16] Mozilla - Bugzilla. Último acceso 10 de Abril 2011. <http://www.bugzilla.org/about/>.
- [17] Open System Testing Architecture. Último acceso 10 de Abril 2011. <http://opensta.org/>.
- [18] Oracle - Hudson. Último acceso 14 de marzo 2011. <http://hudson-ci.org/>.
- [19] Roger S. Pressman. *Software Engineering*. McGraw Hil.
- [20] Universidad de Valparaíso - Decreto 427. Último acceso 14 de marzo 2011. <http://disico.uv.cl/images/docs/decreto427.pdf/>.
- [21] Villarreal Larrinaga, Landate Rodríguez. El estudio de casos como metodología de investigación científica. Investigaciones Europeas de Dirección y Economía de la Empresa.