

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**ASEGURAMIENTO DE LA CALIDAD EN SPEM
BASADO EN SCRUM**

DANIEL ALEJANDRO OCARANZA BORNAND

INFORME FINAL DEL PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO CIVIL EN INFORMATICA

ENERO DEL 2012

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**ASEGURAMIENTO DE LA CALIDAD EN SPEM
BASADO EN SCRUM**

DANIEL ALEJANDRO OCARANZA BORNAND

PROFESOR GUÍA: RODOLFO VILLARROEL ACEVEDO

CARRERA: INGENIERÍA CIVIL INFORMÁTICA

ENERO DEL 2012

*“A mi familia por haberme apoyado y
acompañado en todo este proceso”*

AGRADECIMIENTOS

Se agradece a mi compañera Tanya por haberme apoyado incondicionalmente durante toda la carrera. A mi madre por entregarme toda su fuerza para realizar este trabajo. A Roberto por entregarme su alegría y apoyo en los momentos difíciles. A Héctor por estar siempre ahí cuando lo necesité. A Fernando por entregarme el coraje necesario para cumplir con mis obligaciones.

También se agradece al Proyecto Fondef D09I1171 ADAPTE (Adaptable Domain And Process Technology Engineering) y a sus integrantes por haber realizado mi proyecto en una temática relacionada y de interés para ADAPTE.

Para finalizar quería agradecer a mi profesor guía, Rodolfo Villarroel, por entregar su paciencia, sabiduría y conocimientos en el desarrollo de este proyecto.

Gracias a todos los mencionados, sin ellos el camino hubiera sido más complicado.

INDICE

RESUMEN	9
ABSTRACT	10
1. LISTA DE FIGURAS	11
2. LISTA DE TABLAS	12
3. INTRODUCCIÓN	13
3.1. DESCRIPCIÓN DEL PROBLEMA	13
3.2. OBJETIVOS	14
3.2.1. OBJETIVOS GENERALES	14
3.2.2. OBJETIVOS ESPECÍFICOS	14
3.3. METODOLOGÍA DE TRABAJO	14
4. ASEGURAMIENTO DE LA CALIDAD	16
4.1. PROPÓSITO	16
4.2. DEFINICIÓN	16
4.3. ATRIBUTO	17
5. SCRUM	18
5.1. MODELO	19
5.2. CONTROL DE LA EVOLUCIÓN DEL PROYECTO	19
5.2.1. REVISIÓN DE LAS ITERACIONES	20
5.2.2. DESARROLLO INCREMENTAL	20
5.2.3. DESARROLLO EVOLUTIVO	20
5.2.4. AUTO-ORGANIZACIÓN	20
5.2.5. COLABORACIÓN	21
5.3. VISIÓN GENERAL DEL PROCESO	21
5.3.1. REUNIONES	22
5.3.2. ELEMENTOS	22

5.3.3.	ROLES	22
6.	SPEM.....	23
6.1.	DEFINICIÓN.....	23
6.2.	ARQUITECTURA.....	26
6.3.	ESTRUCTURA DE PROCESOS	27
6.3.1.	ACTIVIDAD.....	27
6.3.2.	ELEMENTO DE DESGLOSE.....	28
6.3.3.	HITOS	28
6.3.4.	PROCESO ELEMENTO	28
6.3.5.	PROCESO PARÁMETRO	28
6.3.6.	EJECUTOR DE PROCESOS	28
6.3.7.	PROCESO DE ASIGNACIÓN DE RESPONSABILIDADES	29
6.3.8.	ROL DE USO.....	29
6.3.9.	ELEMENTOS DE DESGLOSE DEL TRABAJO	29
6.3.10.	PRODUCTO DE TRABAJO DE USO	29
6.3.11.	RELACIÓN PRODUCTO DE TRABAJO DE USO	29
6.3.12.	SECUENCIA DE TRABAJO	30
6.4.	COMPORTAMIENTO DE PROCESOS	30
6.4.1.	ACTIVITY_EXT.....	30
6.4.2.	CONTROL_FLOW_EXT.....	30
6.4.3.	REFERENCIA EXTERNA.....	30
6.4.4.	STATE_EXT.....	31
6.4.5.	TRANSITION_EXT.....	31
6.4.6.	PARAMETRO DE DEFINICIÓN DE TRABAJO	31
6.5.	MANEJADOR DE CONTENIDOS	31
6.5.1.	CATEGORÍA	31
6.5.2.	DESCRIPCIÓN DE CONTENIDO	32
6.5.3.	ELEMENTOS DESCRIPTIBLES	32
6.5.4.	GUÍA.....	32
6.5.5.	MÉTRICAS.....	32
6.5.6.	SECCIÓN.....	32

6.6.	MÉTODO DE CONTENIDOS.....	33
6.6.1.	ASIGNACIÓN DE RESPONSABLE POR DEFECTO.....	33
6.6.2.	CALIFICACIÓN.....	33
6.6.3.	DEFINICIÓN DE ROL.....	33
6.6.4.	PASO.....	33
6.6.5.	DEFINICIÓN DE TAREA.....	34
6.6.6.	DEFINICIÓN DE HERRAMIENTA.....	34
7.	LEVANTAMIENTO DE REQUERIMIENTOS UTILIZANDO SPEM.....	35
7.1.	DEFINICIÓN DE ROLES.....	35
7.2.	DEFINICIÓN DE TAREAS.....	37
7.2.1.	IDENTIFICACIÓN EN TERRENO DE LAS NECESIDADES DEL CLIENTE.....	37
7.2.2.	IDENTIFICAR STAKEHOLDERS.....	38
7.2.3.	CREAR REQUERIMIENTOS.....	40
7.3.	MODELO COMPLETO.....	42
7.4.	MÉTRICA PARA DEFINIR LA CALIDAD DEL LEVANTAMIENTO DE REQUERIMIENTOS.....	43
7.4.1.	PARAMETROS A EVALUAR.....	43
7.4.2.	ESCALA EVALUADORA.....	44
7.4.3.	EVALUADORES.....	44
7.4.4.	MÉTRICA FINAL.....	44
8.	DESARROLLO DE SOFTWARE CON SCRUM EN SPEM.....	46
8.1.	DEFINICIÓN DE ROLES.....	46
8.1.1.	SCRUMMASTER.....	46
8.1.2.	PRODUCT OWNER.....	47
8.1.3.	EQUIPO.....	47
8.2.	DEFINICIÓN DE TAREAS.....	48
8.2.1.	DEFINIR PRODUCT OWNER.....	48
8.2.2.	REUNIÓN PLAN LANZAMIENTO.....	49
8.2.3.	DEFINICIÓN Y PLANIFICACIÓN DE SPRINTS.....	50
8.2.4.	REUNIÓN DIARIA.....	52

8.2.5.	DESARROLLO.....	53
8.2.6.	REVISIÓN DE SPRINT.....	54
8.3.	MODELO COMPLETO	55
8.4.	MÉTRICA PARA DEFINIR LA CALIDAD DEL DESARROLLO DE SOFTWARE EN SCRUM	56
8.4.1.	PARÁMETROS A EVALUAR	56
8.4.2.	ESCALA EVALUADORa.....	57
8.4.3.	EVALUADORES.....	57
8.4.4.	MÉTRICA FINAL.....	57
9.	PUESTA EN MARCHA	59
9.1.	PROYECTO	59
9.1.1.	EMPRESA	59
9.1.2.	CLIENTE	60
9.1.3.	DESCRIPCIÓN PROYECTO.....	61
9.1.4.	PARTICIPANTES Y ROLES.....	61
9.2.	CAPACITACIÓN.....	62
10.	RESULTADOS OBTENIDOS	63
10.1.	DOCUMENTO NECESIDADES.....	63
10.2.	DOCUMENTO STAKEHOLDER.....	66
10.3.	DOCUMENTO REQUERIMIENTOS	68
10.4.	PLAN DE LANZAMIENTO.....	69
10.5.	DOCUMENTO SPRINTS.....	71
10.6.	PRODUCTO SPRINT.....	75
11.	CONCLUSIÓN.....	78
12.	REFERENCIAS.....	80

RESUMEN

El Aseguramiento de la Calidad es el proceso de verificación de que los estándares de desarrollo sean aplicados de manera correcta. Es por esto que el objetivo de esta investigación es desarrollar un método para medir la calidad del software, utilizando el lenguaje de modelamiento SPEM, y la metodología de desarrollo de software Scrum.

Para realizar esta medición de calidad es necesario primero definir los productos que se evaluarán, en este caso los productos obtenidos en un proceso de levantamiento de requerimientos de software y también los productos obtenidos en un proceso de desarrollo de software utilizando Scrum. Para evaluar la calidad del producto se utilizarán dos puntos de vista, uno de parte del cliente y otro de parte de la empresa que desarrolla el producto, en base a evaluaciones ponderadas.

Durante la fase de implantación de este método, la calidad de los productos desarrollados tuvo un aumento constante por cada iteración realizada, con lo cual el cliente, que tuvo participación durante la evaluación de calidad del producto, está de acuerdo con el nivel de calidad entregado por la empresa proveedora.

Palabras-Claves: SPEM, Scrum, calidad del software, ingeniería de procesos, ingeniería de software

ABSTRACT

The Quality Assurance is the process of verifying that development standards are applied correctly. This is why the objective of this research is to develop a method to measure software quality, using the SPEM modeling language, and software development methodology Scrum

To make this measurement of quality is necessary to first define the products to be evaluated, in this case the products obtained in the process of software requirements gathering and the products obtained in the process of software development using Scrum. To evaluate the quality of the product will use two perspectives, one from the client and a part of the company that developed the product, based on weighted evaluations.

During the implementation of this method, the quality of products developed had a constant increase for each iteration performed, whereby the customer, who had partition during evaluation of product quality, agrees with the quality delivered by the supplier company.

Keywords: SPEM, Scrum, software quality, process engineer, software engineer

1. LISTA DE FIGURAS

Figura 5.1 Estructura del desarrollo ágil.....	18
Figura 5.2 Estructura central de Scrum.....	19
Figura 5.3 Proceso completo de Scrum.	21
Figura 6.1 Marco de Trabajo de Ingeniería de Procesos.....	23
Figura 6.2 Relaciones de instanciación de los niveles MOF.....	24
Figura 6.3 Forma de representar un proceso en SPEM.	25
Figura 7.4 Arquitectura SPEM.....	26
Figura 7.1 Definición del Rol Analista en SPEM.....	36
Figura 7.2 Definición del Rol de Uso “Analista Terreno” en SPEM.....	36
Figura 7.3 Definición del Rol de Uso llamado “Analista Líder” en SPEM.....	37
Figura 7.4 Tarea Identificación en terreno de las necesidades en SPEM.....	38
Figura 7.5 Tarea identificar Stakeholders en SPEM.....	39
Figura 7.6 Tarea Crear requerimientos en SPEM.....	41
Figura 7.7 Modelo completo del proceso levantamiento de requerimientos en SPEM...	42
Figura 8.1 Definición del Rol de Uso ScrumMaster en SPEM.....	46
Figura 8.2 Definición del Rol de Uso ProductOwner en SPEM.....	47
Figura 8.3 Definición del Rol de Uso Equipo en SPEM.....	48
Figura 8.4 Tarea Definir Product Owner en SPEM.....	49
Figura 8.5 Tarea Reunión Plan de Lanzamiento en SPEM.....	50
Figura 8.6 Tarea Definición y planificación de Sprints en SPEM.....	51
Figura 8.7 Tarea Reunión Diaria en SPEM.....	52
Figura 8.8 Tarea Desarrollo en SPEM.....	53
Figura 8.9 Tarea Revisión de Sprint en SPEM.....	54
Figura 8.10 Modelo completo del proceso Desarrollo de Software en SPEM.....	55

2. LISTA DE TABLAS

Tabla 4.1 Atributos que definen la calidad del software.....	17
Tabla 7.1 Cuatro niveles conceptuales de MOF.....	24
Tabla 7.1 Ejemplo: evaluación de calidad del producto documento necesidades.....	45
Tabla 8.1 Ejemplo de evaluación de la calidad del producto “Sprint”	58
Tabla 9.1 Nombre y Rol de las personas que participan en el proyecto.	61
Tabla 10.1 Primera iteración análisis de calidad documento necesidades.....	63
Tabla 10.2 Segunda iteración análisis de calidad documento necesidades.....	64
Tabla 10.3 Tercera iteración análisis de calidad documento necesidades	65
Tabla 10.4 Primera iteración análisis de calidad documento Stakeholder.....	66
Tabla 10.5 Segunda iteración análisis de calidad documento Stakeholder.....	67
Tabla 10.6 Primera iteración análisis de calidad documento requerimientos	68
Tabla 10.7 Segunda iteración análisis de calidad documento requerimientos	69
Tabla 10.8 Primera iteración análisis de calidad plan de lanzamiento	70
Tabla 10.9 Segunda iteración análisis de calidad plan de lanzamiento	71
Tabla 10.10 Primera iteración análisis de calidad documento Sprints	72
Tabla 10.11 Segunda iteración análisis de calidad documento Sprints	73
Tabla 10.12 Tercera iteración análisis de calidad documento Sprints.....	74
Tabla 10.13 Análisis de calidad producto sprint N°1 primera iteración.....	75
Tabla 10.14 Análisis de calidad producto sprint N°2 primera iteración.....	75
Tabla 10.15 Análisis de calidad producto sprint N°2 segunda iteración	76
Tabla 10.16 Análisis de calidad producto sprint N°3 primera iteración.....	76
Tabla 10.17 Análisis de calidad producto sprint N°3 segunda iteración	77

3. INTRODUCCIÓN

En la actualidad la calidad de software es un factor fundamental para el desarrollo del negocio de una empresa, por ende es importante que las empresas de tecnología, en específico las empresas desarrolladoras de software fijen métricas, procesos y métodos para mejorar la calidad de sus productos desarrollados.

Al no establecer medidas para mejorar la calidad del software, lleva a tener problemas tales como: no terminar el producto software a tiempo, elevar el costo del producto, encontrar errores luego de haber realizado una entrega y ver empíricamente el progreso del desarrollo del sistema.

Por otra parte está Scrum, que es una metodología para mejorar la eficacia relativa de las prácticas de desarrollo, de modo que se puedan mejorar, mientras el mismo tiempo sirve como un marco en el que los productos complejos pueden ser desarrollados.

Si no se posee de una herramienta y un estándar para modelar el proceso de ingeniería del desarrollo de sistemas, no es posible tener una visión macro del proceso completo. Para esto nace SPEM (System & Software Process Engineering Metamodel), el cual pretende ser el estándar industrial para la representación de modelos de procesos de ingeniería de software e ingeniería de sistemas.

3.1. DESCRIPCIÓN DEL PROBLEMA

SPEM es el estándar aprobado por Object Management Group (OMG) institución creadora de UML (Unified Modeling Language) el cual tiene como propósito ser el estándar industrial para la representación de modelos de procesos de ingeniería del software e ingeniería de sistemas.

Por otra parte, el aseguramiento de la calidad (Quality Assurance), según la norma ISO 8402(UNE 66-001-92), se define como “El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas”. Esto se puede definir como el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto (software) requiere para satisfacer los requerimientos de calidad entregados por el cliente.

En la actualidad no existen parámetros, indicadores o criterios de medición para controlar la calidad del software en SPEM.

3.2. OBJETIVOS

3.2.1. OBJETIVOS GENERALES

Desarrollar un método para asegurar la calidad del software en SPEM basándose en Scrum.

3.2.2. OBJETIVOS ESPECÍFICOS

- Definir parámetros de medición de la calidad del software.
- Definir un método para cumplir los parámetros de medición de la calidad del software en SPEM.
- Implementar este método desarrollado en alguna empresa del rubro.
- Analizar resultados de la implementación.

3.3. METODOLOGÍA DE TRABAJO

La metodología de trabajo será separada en las siguientes etapas:

- **Formulación del problema de investigación:** En esta etapa se define el tema a investigar
- **Fase exploratoria:** Esta etapa, consiste en revisión de la literatura y la construcción del marco teórico.
- **Diseño de la investigación:** Se diseñará como lograr los objetivos específicos de la investigación y como se obtendrán los datos.
- **Extracción de la muestra:** Se selecciona la empresa donde se aplica la investigación.
- **Procedimiento para la recopilación de datos:** Se obtienen los datos de la empresa donde se aplica la investigación.

- **Trabajo de gabinete:** Se ordenan los datos, se realiza el análisis correspondiente y se interpreta la información recopilada.
- **Elaboración de informe:** Se realiza el informe completo, con todos los pasos de la investigación.

4. ASEGURAMIENTO DE LA CALIDAD

4.1. PROPÓSITO

Una de las principales etapas dentro del proceso de desarrollo de software es el Aseguramiento de la Calidad, esto es un marco de trabajo sistemático y planeado de todas las acciones necesarias para proveer al cliente una confianza sobre el desarrollo de un sistema según los requerimientos solicitados sobre un producto de software [8].

Esta actividad es el proceso de verificación de que los estándares de desarrollo sean aplicados de manera correcta. En proyectos de menor envergadura esta actividad la puede realizar el mismo equipo de desarrollo, pero en proyectos grandes, un grupo específico de personas debe realizar esta tarea.

El plan de aseguramiento de la calidad de software define las actividades específicas que tiene que realizar un proyecto en particular. Para cada fase de un proyecto es necesario crear un plan de monitoreo.

4.2. DEFINICIÓN

Existen diferentes puntos de vistas para definir la calidad del software. Desde el punto de vista de cumplir los requerimientos Roger Pressman la define como [15]:

“El cumplimiento de los requerimientos funcionales y performance explícitamente definidos, de los estándares de desarrollo explícitamente documentados y de las características implícitas esperadas del desarrollo de software profesional”.

Desde el punto de vista del cliente o usuario Watts Humphrey dice [16]:

“El foco principal de cualquier definición de calidad de software debería ser las necesidades del cliente. Crosby al igual que Pressman define la calidad como conformidad con los requerimientos. Mientras uno puede discutir la diferencia entre requerimientos, necesidades y deseos, la definición de calidad debe considerar la perspectiva de los usuarios. Entonces las preguntas claves son ¿Quiénes son los usuarios?, ¿Qué es importante para ellos? Y ¿Cómo sus prioridades se relacionan con la manera en que se construye, empaqueta y se da soporte al producto?”.

Al Davis define la calidad de software como [17]:

“La calidad no se trata de tener cero defectos o una mejora medible de la proporción de defectos, no se trata de tener los requerimientos documentados. No es más ni menos que satisfacer las necesidades del cliente (por más que las necesidades estén o no correctamente documentadas).

Luego desde estas perspectivas el glosario de la IEEE para la ingeniería de software define la calidad como [18]:

“El grado con el cual un sistema, componente o proceso cumple con los requerimientos y con las necesidades y expectativas del usuario”.

Por lo tanto, para que estas definiciones tengan sentido, la calidad debe ser medible según parámetros e indicadores, ya que como dice Tom DeMarco [19], “No se puede controlar lo que no se puede medir”.

4.3. ATRIBUTO

Para conseguir cumplir con los estándares del aseguramiento de la calidad se debe medir cuantitativamente los aspectos de calidad usando métricas bien establecidas.

Según Craig Larman [13]: Se definen las siguientes características las cuales sirven para clasificar los atributos de calidad del software.

Sigla	Tipo de Requerimiento	Descripción
F	Funcional	Características, capacidades y aspectos de seguridad
U	Usabilidad (Facilidad de uso)	Factores humanos (interacción), ayuda, documentación.
R	Fiabilidad (Reliability)	Frecuencia de Fallos, capacidad de recuperación de un fallo y grado de previsión
P	Rendimiento (Performance)	Tiempos de Respuesta, productividad, precisión, disponibilidad, uso de los recursos
S	Soporte	Adaptabilidad, facilidad de mantenimiento, internacionalización, facilidad de configuración
	Implementación	Limitación de recursos, lenguaje y herramientas, hardware
	Interfaz	Restricciones impuestas para la interacción con sistemas externos
+	Operaciones	Gestión del sistema, pautas administrativas, puesta en marcha
	Empaquetamiento	Forma de distribución
	Legales	Licencia, derechos de autor, etc.

Tabla 4.1 Atributos que definen la calidad del software

5. SCRUM

Scrum es una metodología ágil de desarrollo de proyectos que toma su nombre y principios de los estudios realizados sobre nuevas prácticas de producción por Hirotaka Takeuchi e Ikujiro Nonaka a mediados de los 80 [1].

Al principio surgió como modelo de desarrollo de productos tecnológicos, también se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad, situaciones frecuentes en el desarrollo de determinados sistemas de software.

Jeff Sutherland aplicó el modelo Scrum al desarrollo de software en 1993 en Easel Corporation. En 1996 lo presentó junto con Ken Schwaber como proceso formal, también para gestión del desarrollo de software en OOPSLA 96. Posteriormente, en el 2001 serían dos de los promulgadores del Manifiesto Ágil [3].

Scrum es una metodología de desarrollo muy simple, que requiere trabajo duro porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto.

Como metodología ágil Scrum es un modo de desarrollo de carácter adaptable más que predictivo, es orientado a las personas más que a los procesos, emplea la estructura de desarrollo ágil (ver Figura 5.1), es decir, incremental basada en iteraciones y revisiones.

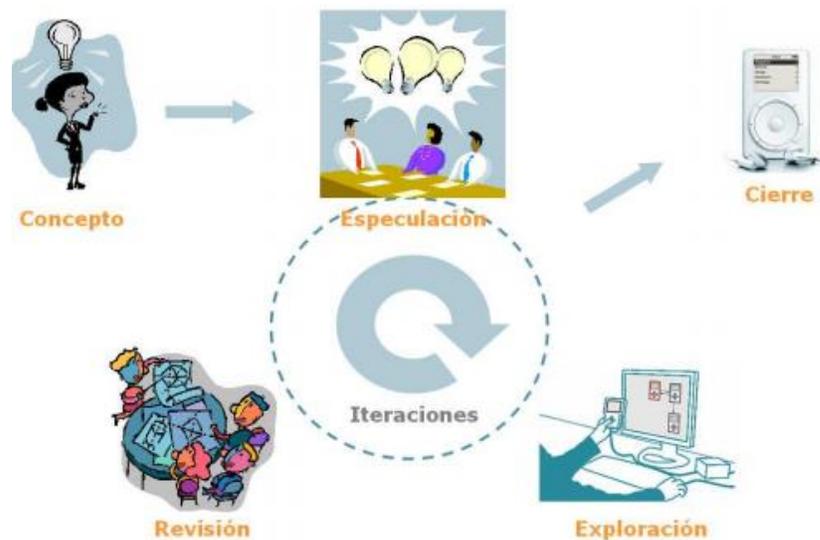


Figura 5.1 Estructura del desarrollo ágil.

5.1. MODELO

Scrum comienza dando una visión general del producto, especificando y dando detalles de las funcionalidades o partes que tienen mayor prioridad de desarrollo y que pueden llevarse a cabo en un período de tiempo breve (30 días).

Cada uno de estos períodos de desarrollo es una iteración que finaliza en la producción de un incremento operativo del producto.

Estas iteraciones son la base del desarrollo ágil, y Scrum gestiona su evolución a través de reuniones breves diarias en las que todo el equipo revisa el trabajo realizado el día anterior y el previsto para el día siguiente, esto se puede visualizar gráficamente en la figura 6.2.

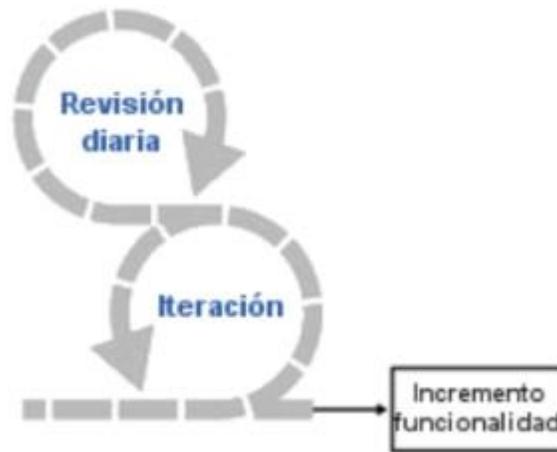


Figura 5.2 Estructura central de Scrum.

5.2. CONTROL DE LA EVOLUCIÓN DEL PROYECTO

Scrum controla de forma empírica y adaptable la evolución del proyecto, empleando las siguientes prácticas que se definirán a continuación de la gestión ágil.

5.2.1. REVISIÓN DE LAS ITERACIONES

Al finalizar cada iteración, se lleva a cabo una revisión con todas las personas implicadas en el proyecto. Este es el período máximo que se tarda en reconducir una desviación en el proyecto o en las circunstancias del producto.

5.2.2. DESARROLLO INCREMENTAL

Durante el proyecto, las personas implicadas no trabajan con diseños o abstracciones. El desarrollo incremental implica que al final de cada iteración se dispone de una parte del producto operativa que se puede inspeccionar y evaluar.

5.2.3. DESARROLLO EVOLUTIVO

Los modelos de gestión ágil se emplean para trabajar en entornos de incertidumbre e inestabilidad de requisitos. Intentar predecir en las fases iniciales como será el producto final, y sobre dicha predicción desarrollar el diseño y la arquitectura del producto no es realista, porque las circunstancias obligarán a remodelarlo muchas veces.

Para qué predecir los estados finales de la arquitectura o del diseño si van a estar cambiando. En Scrum se toma la inestabilidad como una premisa, y se adoptan técnicas de trabajo para permitir esa evolución sin degradar la calidad de la arquitectura que se irá generando durante el desarrollo.

El desarrollo Scrum va generando el diseño y la arquitectura final de forma evolutiva durante todo el proyecto. No los considera como productos que deben realizarse en la primera fase del proyecto.

5.2.4. AUTO-ORGANIZACIÓN

Durante el desarrollo de un proyecto son muchos los factores impredecibles que surgen en todas las áreas y niveles. La gestión predictiva confía la responsabilidad de su resolución al gestor de proyectos. En Scrum los equipos son auto-organizados (no auto-dirigidos), con margen de decisión suficiente para tomar las decisiones que consideren oportunas.

5.2.5. COLABORACIÓN

Las prácticas y el entorno de trabajo ágiles facilitan la colaboración del equipo. Ésta es necesaria, porque para que funcione la auto-organización como un control eficaz, cada miembro del equipo debe colaborar de forma abierta con los demás, según sus capacidades y no según su rol o su puesto.

5.3. VISION GENERAL DEL PROCESO

Scrum denomina “sprint” a cada iteración de desarrollo y recomienda realizarlas con duración de 30 días (ver figura 5.3). El sprint es por lo tanto el núcleo central que proporciona la base de desarrollo iterativo e incremental.

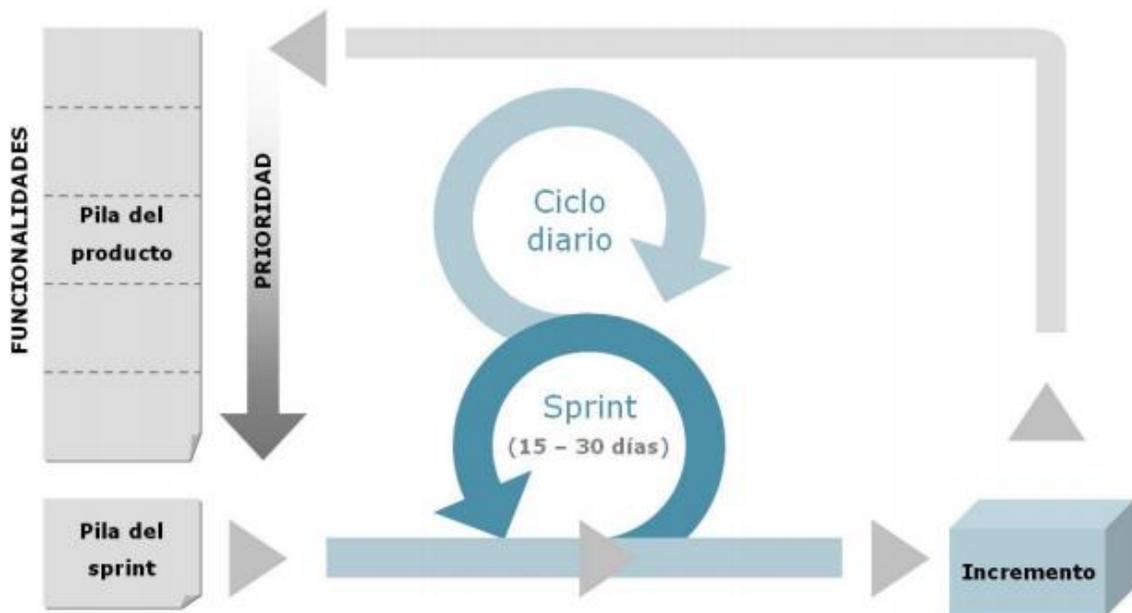


Figura 5.3 Proceso completo de Scrum.

Los elementos que conforman el desarrollo de Scrum se detallan a continuación.

5.3.1. REUNIONES

Planificación de sprint: Jornada de trabajo previa al inicio de cada sprint en la que se determina cual va a ser el trabajo y los objetivos que se deben cumplir en esa iteración.

Reunión diaria: Breve revisión del equipo del trabajo realizado hasta la fecha y la previsión para el día siguiente.

Revisión de sprint: Análisis y revisión del incremento generado.

5.3.2. ELEMENTOS

Pila del producto: lista de requisitos de usuario que se origina con la visión inicial del producto y va creciendo y evolucionando durante el desarrollo.

Pila del sprint: lista de los trabajos que debe realizar el equipo durante el sprint para generar el incremento previsto.

Incremento: Resultado de cada sprint.

5.3.3. ROLES

Scrum clasifica a todas las personas que intervienen o tienen interés en el desarrollo del proyecto en, propietario del producto, equipo, gestor de Scrum (también llamado Scrum Manager o Scrum Master) y otros interesados [7].

Propietario del producto: El responsable de obtener el mayor valor de producto para los clientes, usuarios y resto de los implicados.

Equipo de desarrollo: grupo o grupos de trabajo que desarrollan el producto.

Scrum Manager: Gestor de los equipos que es responsable del funcionamiento de la metodología Scrum y de la productividad del equipo de desarrollo.

6. SPEM

6.1. DEFINICIÓN

SPEM (Software & System Process Engineering Metamodel) es un estándar desarrollado y aprobado por la OMG (Object Management Group), el cual es un consorcio dedicado al cuidado y al establecimiento de diversos estándares de tecnologías orientadas a objetos. SPEM pretende ser el estándar industrial para la representación de modelos de procesos de ingeniería del software (ver Figura 6.1) e ingeniería de sistemas [6].

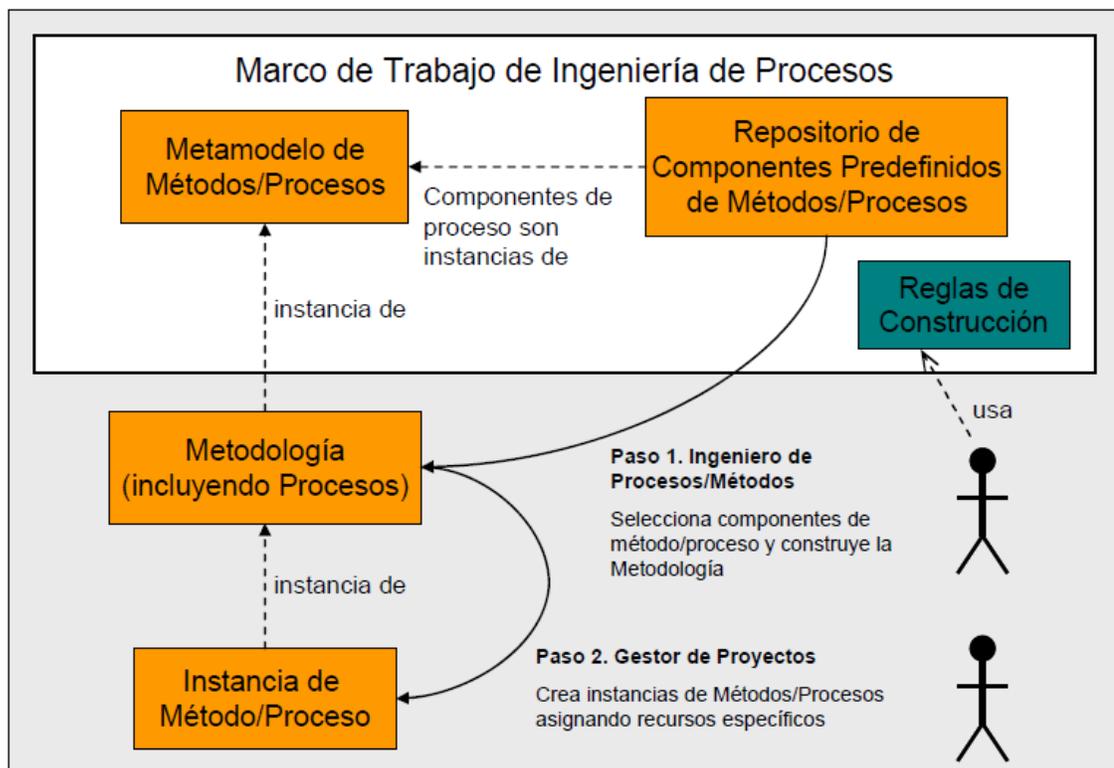


Figura 6.1 Marco de Trabajo de Ingeniería de Procesos [10].

SPEM está basado en MOF (Meta Object Facility), un estándar previo definido por la OMG que define una arquitectura de modelado de cuatro niveles conceptuales [11].

Tabla 6.1 Cuatro niveles conceptuales de MOF.

Nivel	MOF	Ejemplo
M3	Modelo MOF (Meta-Metamodelado)	
M2	Meta –Modelo	UML, SPEM, Entidad-Interrelación
M1	Modelo	Modelo en UML
M0	Datos	Datos de un proyecto concreto

Entre estos niveles existen relaciones de instanciación, de forma que un nivel más pequeño es una instancia de un nivel más alto. (ver figura 6.2).

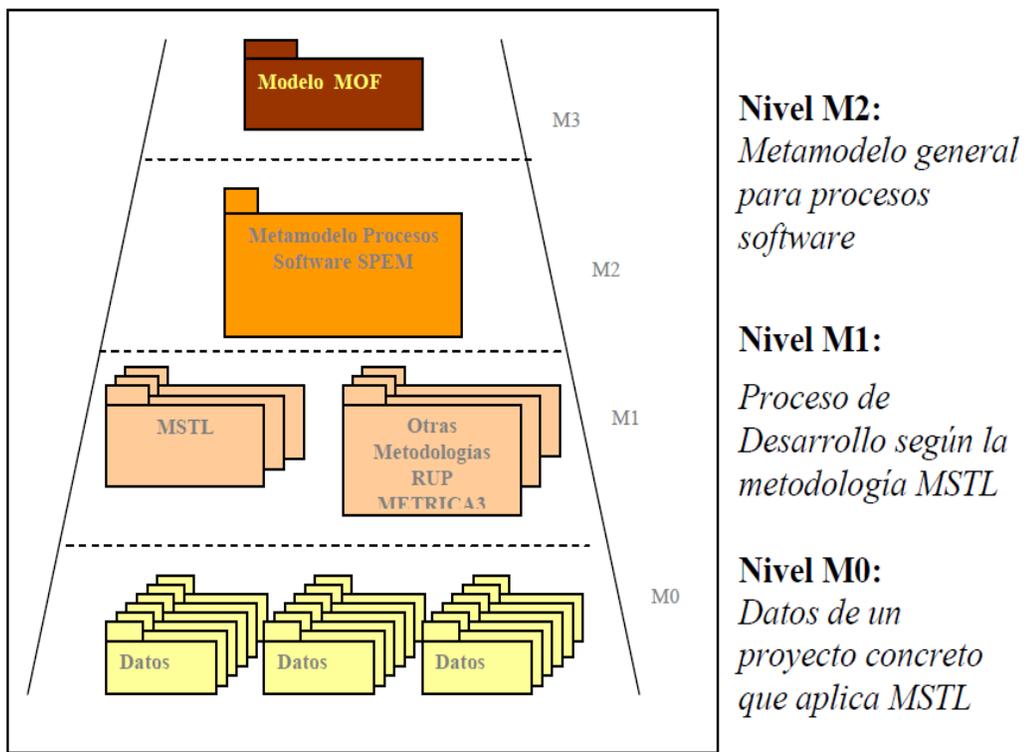


Figura 6.2 Relaciones de instanciación de los niveles MOF [10].

SPEM sirve para definir procesos de desarrollo de software y sistemas y sus componentes. Su alcance se limita a los elementos mínimos definidos en el proceso sin agregar características específicas del proceso, pero sirve para métodos y procesos de diferentes estilos, culturas, niveles de formalismo, o modelos de ciclos de vida.

SPEM no es un lenguaje de modelado de procesos en general, ya que está orientado a los procesos de software, pero incluye métodos para incluir diagramas de UML, BPMN [9], etc.

La figura 6.3 muestra la forma de representar procesos, basada en 3 elementos:

- **Roles:** representan quien realiza la tarea.
- **Producto de trabajo:** representan las entradas que se utilizan en las tareas y las salidas que producen.
- **Tarea:** representan el esfuerzo a realizar.

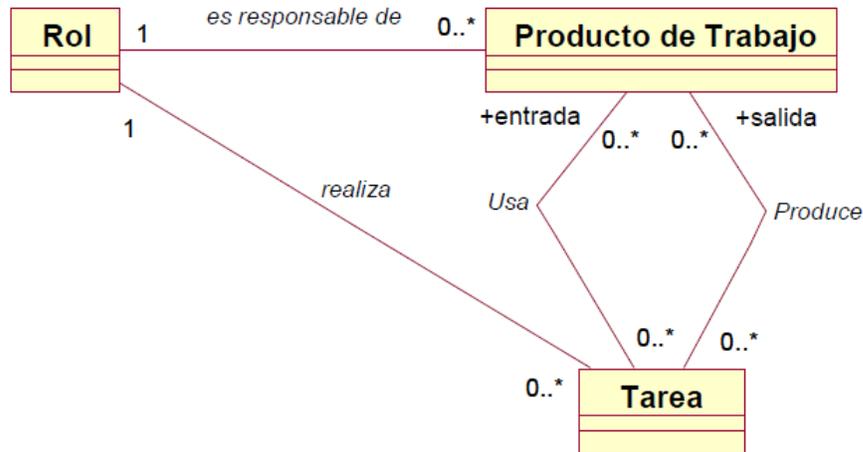


Figura 6.3 Forma de representar un proceso en SPEM. [11]

SPEM puede resolver problemas de proceso tales como:

- Miembros de los equipos que no poseen toda la información centralizada en cuanto a los procesos, SPEM da fácil acceso.
- Los desarrolladores poseen diferentes fuentes y versiones de la misma información, SPEM lo centraliza.
- Combinar e Integrar procesos que estén en formatos distintos, SPEM es un estándar.
- Existen variadas formas y estilos de representar los procesos, según libro, tecnología y herramienta utilizada.

Adicionalmente, SPEM provee un marco de trabajo conceptual para modelar, documentar, presentar, gestionar, intercambiar y realizar métodos y procesos de software. Por esto está destinado a ingenieros de procesos, jefes de proyectos, gestores de proyectos, los cuales son responsables de mantener e implementar procesos para sus organizaciones.

6.2. ARQUITECTURA

SPEM está estructurado en 7 paquetes de Metamodelado principales (ver Figura 6.4). La estructura divide al modelo en unidades lógicas, cada unidad extiende a la unidad que depende, agregando estructuras y capacidades adicionales al elemento de la unidad inferior. En general, el paquete de UML 2.0 mezcla mecanismos que extienden las capacidades de SPEM para modelar unidad por unidad. Como resultado, las unidades son definidas en capas inferiores a las que pueden ser realizadas por SPEM [12].

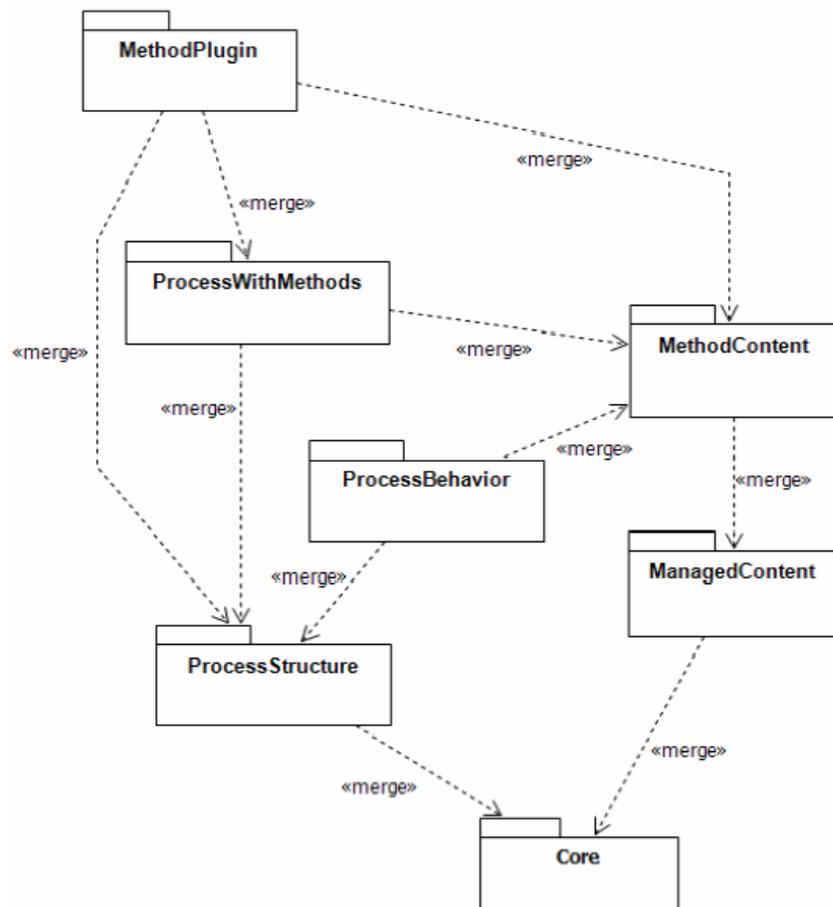


Figura 6.4 Arquitectura SPEM

Core: El paquete Core contiene las clases del meta-modelo y las abstracciones que construyen la base para todas las clases en los otros paquetes. En otras palabras, todas las clases.

Process Structure: Este paquete define las bases para todo el modelo de procesos. Soporta la creación de modelos de procesos simples y flexibles.

Process Behavior: El paquete “Process Structure” representa un proceso como una estructura estática, permitiendo la anidación de actividades y definiendo dependencias predecesoras. El paquete “Process Behavior” permite extender estas estructuras con modelos de comportamiento. Sin embargo, esto no define su propio modelo de comportamiento, sino que ofrece vínculos a los actuales modelos de comportamientos definidos externamente, habilitando el reutilizamiento de estos enfoques de otras especificaciones del OMG. Por ejemplo, un proceso definido con los conceptos de la estructura de procesos, pueden ser vinculados con un diagrama de actividad de UML que representan el comportamiento de ese proceso.

Managed Content: El desarrollo de procesos en varios casos no solo son representados con modelos, sino que son documentados como descripciones en lenguaje natural. El paquete Managed Content incluye conceptos para manejar el contenido textual como descripciones.

Method Content: Este paquete ayuda a usuarios y organizaciones a construir una base de conocimientos de desarrollo que es independiente de cualquier proceso específico y desarrollo de proyectos. Agrega conceptos para definir el ciclo de vida y procesos independientes reutilizables que proveen una base de conocimientos de métodos de desarrollo, técnicas y mejores prácticas. Este paquete comprende explicaciones textuales paso a paso, describiendo como específicamente las metas del desarrollo son logradas por los roles con que recursos y resultados fueron hechos, independiente de la capa de la arquitectura utilizada.

6.3. ESTRUCTURA DE PROCESOS

Este paquete de Metamodelado contiene las estructuras básicas para definir los procesos de desarrollo. Define como los proyectos de desarrollo deben ser ejecutados.

6.3.1. ACTIVIDAD

Una actividad es un elemento de desglose del trabajo y adicionalmente define una definición del trabajo, esta es una unidad básica de trabajo dentro de un proceso, así como un mismo proceso, simplificando, cada actividad representa un proceso.

Una actividad es una definición del trabajo que representa una unidad general de trabajo asignable a específicos actores que son representados como un Rol de uso.

6.3.2. ELEMENTO DE DESGLOSE

Un elemento de desglose es una generalización abstracta para cualquier tipo de proceso que es parte de una estructura de desglose. Define un conjunto de propiedades disponibles para todo tipo de especialización.

6.3.3. HITOS

Un hito es un evento significativo en un proyecto de desarrollo, tales como una decisión importante, o una reunión con un superior.

6.3.4. PROCESO ELEMENTO

Un proceso es cualquier elemento que es parte de un proceso. Es un elemento extensible que representa una generalización abstracta para todos los procesos que son parte de SPEM.

6.3.5. PROCESO PARÁMETRO

Un proceso parámetro es un elemento que es usado para definir procesos. Define las entradas y salidas de producto de trabajo.

6.3.6. EJECUTOR DE PROCESOS

Este representa una relación entre una actividad y una instancia de Rol de Uso. Una instancia de proceso ejecutor vincula una o más instancias de Rol de Uso a una actividad.

6.3.7. PROCESO DE ASIGNACIÓN DE RESPONSABILIDADES

Representa una relación entre una instancia de Rol de Uso y un Uso de Producto de Trabajo. Una instancia de Proceso de Asignación de Responsabilidades vincula uno o más Roles de Uso a solo un Uso de Producto de Trabajo.

6.3.8. ROL DE USO

Un Rol de Uso es un elemento que representa un ejecutante de una actividad o un participante de la actividad. Si es un ejecutante, el Rol de Uso y la actividad deben ser relacionados por un ejecutor de proceso. Si es un participante, entonces el Rol de uso es almacenado en un elemento de desglose menor de una actividad y puede ser usado por una de las sub-actividades como un ejecutor.

6.3.9. ELEMENTOS DE DESGLOSE DEL TRABAJO

Es un elemento especial de desglose que proporciona propiedades específicas para elementos de desglose.

6.3.10. PRODUCTO DE TRABAJO DE USO

Es un elemento de desglose especial que puede representar una entrada o salida para una actividad o un participante de una actividad. Si es una entrada o salida, el Producto de Trabajo de Uso necesita estar relacionado a una actividad mediante un Proceso Parámetro. Si es un participante, el Producto de Trabajo de Uso es almacenado en un elemento de desglose menor de una actividad y puede ser usado por una de las sub-actividades como una entrada/salida y/o puede ser relacionada a un Rol de Uso, por medio de Procesos de Asignación de Responsabilidades.

6.3.11. RELACIÓN PRODUCTO DE TRABAJO DE USO

Expresa una relación general entre productos de trabajo.

6.3.12. SECUENCIA DE TRABAJO

Es un elemento de desglose que representa una relación entre dos elementos de desglose en el cual un elemento de desglose de trabajo depende del comienzo o finalización de otro elemento de desglose de trabajo en orden de comenzar o terminar.

6.4. COMPORTAMIENTO DE PROCESOS

SPEM no apunta a ser un lenguaje genérico de modelado de procesos, pero provee su propio concepto de modelado de comportamientos. SPEM, deja a los implementadores elegir el modelo de comportamiento genérico que satisfaga sus necesidades. SPEM provee estructuras específicas adicionales para que los modelos de comportamiento genéricos aumenten su capacidad de describir características del desarrollo de procesos.

6.4.1. ACTIVITY_EXT

Representa una referencia al modelo de clases en un modelo de comportamiento externo representando una definición de un comportamiento.

6.4.2. CONTROL_FLOW_EXT

Representa una referencia a un modelo de clases en un modelo de comportamiento externo representando un control de flujo.

6.4.3. REFERENCIA EXTERNA

Es un clasificador que representa una generalización abstracta de todas las clases representando referencias a modelos de comportamiento externos.

6.4.4. STATE_EXT

Representa una referencia a un modelo de clases en un modelo de comportamiento externo representando un estado.

6.4.5. TRANSITION_EXT

Representa una referencia a un modelo de clases en un modelo de comportamiento externo representando una transición entre estados.

6.4.6. PARAMETRO DE DEFINICIÓN DE TRABAJO

Este paquete extiende la clase Definición de Parámetros de trabajo que está en el paquete Core con asociaciones a entradas o estados existentes.

6.5. MANEJADOR DE CONTENIDOS

El paquete de manejador de contenidos define los conceptos fundamentales para manejar descripciones textuales para procesos y elementos de método de contenidos. Introduce una clase abstracta llamada Elemento Describible, que a través de la mezcla de paquetes, sirve como clase padre para Elementos de Proceso, definido en el paquete Estructura de Proceso.

6.5.1. CATEGORÍA

Una categoría es un Elemento Describible usado para categorizar, por ejemplo, agrupar cualquier número de Elementos Describibles de cualquier subtipo basado en criterio definido por el usuario.

6.5.2. DESCRIPCIÓN DE CONTENIDO

Es una clase, que es usada para almacenar la descripción textual para un Elemento Descriptible. Define atributos estándar aplicables para todos los subtipos de Elementos Descriptibles.

6.5.3. ELEMENTOS DESCRIPTIBLES

Es un Elemento Extensible que representa una generalización abstracta para todos los elementos en SPEM que pueden ser documentados con una descripción textual. Ejemplos para Elementos Descriptibles son los Roles y Productos de Trabajo, los cuales tienen texto descriptivo que definen una guía de uso.

6.5.4. GUÍA

Guía, es un Elemento Descriptible que provee información adicional. Ejemplos de guías son los templates, checklists, herramientas mentores, estimadores, reportes, etc.

6.5.5. MÉTRICAS

Una Métrica es un Elemento Descriptible especial que contiene una o más restricciones que proporcionan medidas para cualquier Elemento Descriptible.

6.5.6. SECCIÓN

Una sección es una clase especial que representa una subsección especial de una Descripción de Contenido. Es usado para documentación a larga escala para organizar Elementos descriptibles en secciones.

6.6. MÉTODO DE CONTENIDOS

El paquete de Método de Contenidos define elementos principales de cada método como Roles, Tareas, Definición de Productos de Trabajo.

Fundamentalmente Método de Contenidos, es descrito por Tareas organizada en Pasos.

6.6.1. ASIGNACIÓN DE RESPONSABLE POR DEFECTO

Es un Elemento de Método de Contenidos que representa una relación entre instancias de Definición de Roles y Definición de Productos de Trabajo. Una instancia de Asignación de Responsable por defecto vincula uno o más instancias de Definición de Roles a una Definición de Productos de Trabajo.

6.6.2. CALIFICACIÓN

Es un Elemento de Método de Contenidos que documentan cero o más calificaciones, habilidades o competencias para un Rol y/o Tarea.

6.6.3. DEFINICIÓN DE ROL

Define un grupo de habilidades, competencias y responsabilidades de un Rol que es usado por una Definición de Tarea para definir quién debe realizar la tarea de mejor manera.

6.6.4. PASO

Un paso es una Sección y una Definición de trabajo el cual es usado para organizar una Definición de Tarea en partes pequeñas o subunidades de trabajo.

6.6.5. DEFINICIÓN DE TAREA

Define el trabajo realizado por instancias de Roles. Una Tarea es asociada a entradas y salidas de Productos de Trabajo.

6.6.6. DEFINICIÓN DE HERRAMIENTA

Es un elemento especial de un Elemento de Método de Contenido, que puede ser usado para especificar la participación de una herramienta en una Definición de Tarea.

7. LEVANTAMIENTO DE REQUERIMIENTOS UTILIZANDO SPEM

El Levantamiento de requerimientos es un proceso fundamental dentro del proceso de desarrollo de software, ya que el equipo de desarrollo recibe la descripción de las necesidades de un cliente, que en ocasiones no tiene los conocimientos que se requieren para levantar los requerimientos. Con esto el resultado de un sistema puede no ser completamente satisfactorio por parte del cliente. Parte de este problema es que los usuarios no saben lo que quieren, no aceptan como un compromiso los requerimientos escritos, insistirán en nuevos requerimientos después de fijar costos y agendas, no están disponibles y la comunicación con ellos es lenta, no participan en revisiones de avance del sistema, no entienden el proceso de desarrollo y no les interesa.

Para solucionar este problema existen herramientas y métodos para realizar el levantamiento de requerimientos, tales como Casos de Uso y UML, los cuales son medios para formalizar el proceso de levantamiento de requerimientos.

UML definirá en detalle el producto que el usuario necesita, pero para modelar el proceso de ingeniería de levantamiento de requerimientos es necesario SPEM como herramienta de modelado, para que todos los participantes del proyecto tengan la capacidad de comprender el proceso completo y posteriormente definir el alcance del proyecto cumpliendo los requerimientos que el usuario realmente necesita.

7.1. DEFINICIÓN DE ROLES

Es tarea primordial definir el equipo de trabajo que realizará las tareas de análisis y diseño, estos especialistas en SPEM definen un rol el cual se llamará “Analistas” (ver figura 8.1), estos analistas deberán cumplir con las siguientes características:

- Tener conocimientos del paradigma tradicional de la ingeniería de software y del ciclo de vida del software en Scrum
- Poseer conocimientos de modelado funcional (Diagramas de flujo de datos, diagramas de estado, etc.)
- Tener conocimientos de modelado de datos y sus técnicas (Diagramas entidad relación, modelado relacional, etc.)
- Poseer Conocimiento de la tecnología (arquitectura de software, bases de datos, etc.)

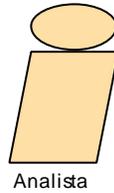


Figura 7.1 Definición del Rol Analista en SPEM

Generalmente es necesario que los analistas visiten la institución para poder observar claramente las necesidades del cliente, estos analistas definen un Rol de Uso del Rol analista, o sea una instancia del Rol “Analista”, este Rol de uso se llamará “Analista Terreno” (ver Figura 8.2)

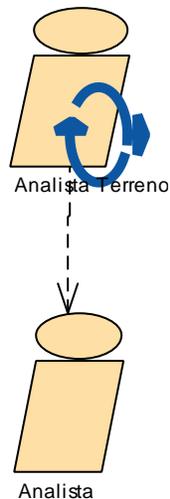


Figura 7.2 Definición del Rol de Uso “Analista Terreno” en SPEM

Como el levantamiento de requerimientos es un trabajo en equipo y todo trabajo en equipo necesita una persona que guía y supervisa las tareas del equipo, esta persona define un nuevo Rol de Uso llamado “Analista Líder” que es una instancia del Rol “Analista” (ver figura 7.3)

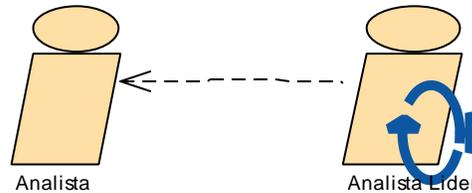


Figura 7.3 Definición del Rol de Uso llamado “Analista Líder” en SPEM

Para el proceso de levantamiento de requerimientos es importante tener bien identificados estos actores, para que las necesidades, alcances y límites del sistema queden bien definidas.

7.2. DEFINICIÓN DE TAREAS

7.2.1. IDENTIFICACIÓN EN TERRENO DE LAS NECESIDADES DEL CLIENTE

La tarea de identificación en terreno de las necesidades del cliente son realizadas por el Analista Terreno el cual tiene que observar cómo funciona el negocio, quienes son los participantes, que se hace, cual serían el modelo de datos, cuales serían los procesos involucrados, y las necesidades del cliente.

Luego de esto el analista en terreno debe generar un documento de necesidades, en el cual se detalle explícitamente las necesidades del cliente apoyándose por alguna herramienta de modelado para que todos los analistas del equipo tengan claro las reales necesidades del cliente. Este proceso se visualiza en la figura 7.4

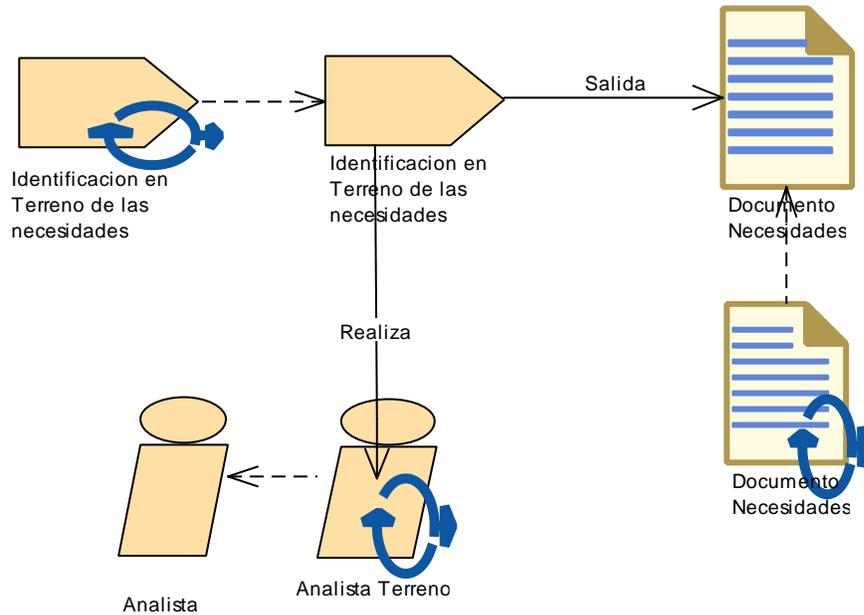


Figura 7.4 Tarea Identificación en terreno de las necesidades en SPeM

7.2.2. IDENTIFICAR STAKEHOLDERS

Otro personaje importante dentro del levantamiento de requerimientos son los Stakeholders. Según R. E. Freeman en su libro: “Strategic Management: A Stakeholder Approach” [20], define esta palabra como “quienes puedan afectar o son afectados por las actividades de una empresa”.

Estos grupos o individuos son las personas interesadas que deben ser considerados como un elemento esencial en la planificación estratégica de negocios. Dentro de la industria del software los Stakeholders son aquellos que están interesados en la realización de un proyecto o tarea, auspiciando el mismo mediante su poder de decisión o de financiamiento. En palabras simples son las personas comprometidas con el proyecto, por parte del cliente, las cuales entregarán la mayor cantidad de información relevante acerca del sistema que el cliente necesita.

Es necesario que el analista en terreno identifique quienes serán los Stakeholders, esto es crítico para el levantamiento de requisitos ya que de ellos depende que el proyecto tenga éxito o fracase, de la misma forma deben establecerse las relaciones entre el equipo de desarrollo y el cliente, de manera que exista una buena comunicación. Para esto debe establecerse una pauta para esta comunicación, ya que son estos quienes deben mediar entre el dominio de los usuarios del software y las palabras técnicas utilizadas por el analista líder.

En la actualidad no existen muchas metodologías o herramientas para identificar a los Stakeholders en un proyecto de software, ya que se asume erróneamente que este proceso es obvio en un desarrollo o bien que los usuarios directos, clientes y el equipo de analistas son los Stakeholders exclusivos de cualquier proyecto.

Para definir los Stakeholders se deben analizar las cualidades, habilidades, conocimiento y experiencia que este debe poseer para que pueda cubrir las características que exigen las actividades específicas del proyecto.

Esto se visualiza gráficamente en SPEM en la figura 7.5.

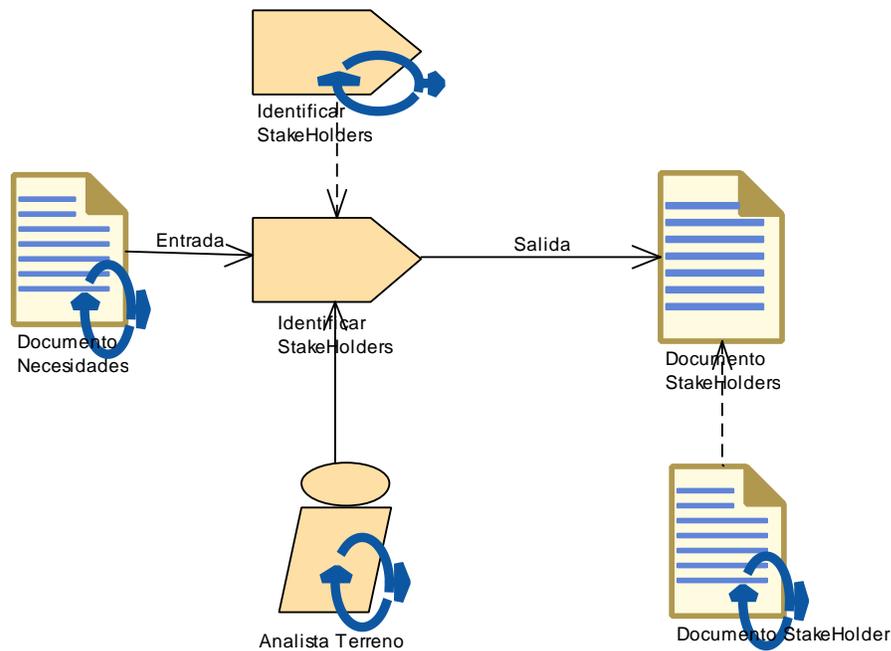


Figura 7.5 Tarea identificar Stakeholders en SPEM

7.2.3. CREAR REQUERIMIENTOS

Una vez que se obtiene el documento de necesidades y los Stakeholders identificados, se puede dar paso a la tarea de crear los requerimientos (ver figura 7.6). Siempre hay que tener en cuenta que las necesidades y los requerimientos no son lo mismo, ya que una necesidad es a una función o proceso que el usuario desea poseer en su sistema, pero por factores económicos, ambientales o tecnológicos no son posibles de entregar por parte de éste, en cambio los requerimientos son aquellas necesidades que pueden ser entregadas por el sistema.

El documento de requerimiento es útil, para comunicar de manera precisa los requerimientos, objetivos, y presunciones del dominio, además puede ser utilizado como un contrato legal entre las partes, servir como base para una estimación de tamaño, costo, precio y planificación de proyecto, ser una herramienta de evaluación para el producto final (verificación y validación), servir como un documento de control de cambios para una evolución del producto.

Además, este documento debe tener las siguientes cualidades:

- **Compleitud:** debe ser completo con respecto a: los objetivos que el usuario desea obtener del sistema, las entradas del sistema (todas las entradas existentes en el sistema), no tener secciones incompletas
- **Consistencia:** que no haya contradicciones en la formulación de objetivos, requerimientos y presunciones
- **Medibilidad:** que los requerimientos deben ser formulados de manera tal que su satisfacción pueda ser evaluada de manera no ambigua por parte del cliente.
- **Precisión:** que no exista vocabulario ambiguo, los objetivos, requerimientos y presunciones deben estar escritos de manera que no permitan interpretaciones distintas.
- **Factibilidad:** los objetivos y requerimientos deben ser realizables dentro del presupuesto y cronograma dispuestos .
- **Entendibilidad:** el documento debe ser entendible por todos los potenciales lectores.
- **Trazabilidad:** debe identificar las fuentes de los objetivos, requerimientos, y presunciones, además de relacionar los requerimientos y presunciones con los objetivos, facilitar las referencias de requerimientos en documentación futura. Buena Estructura, ítems definidos antes de ser usados, índices, formato correcto, etc.

- **Modificabilidad:** debe ser fácil de adaptar, extender, o disminuir, el impacto de modificar un ítem debe ser fácil de estimar.

Para finalizar, el documento de requerimientos es un proceso iterativo e incremental que debe ser supervisado por el Stakeholder y el analista líder y llegar en común acuerdo a un documento de requerimiento que asegure que el producto final sea lo que realmente el usuario necesite.

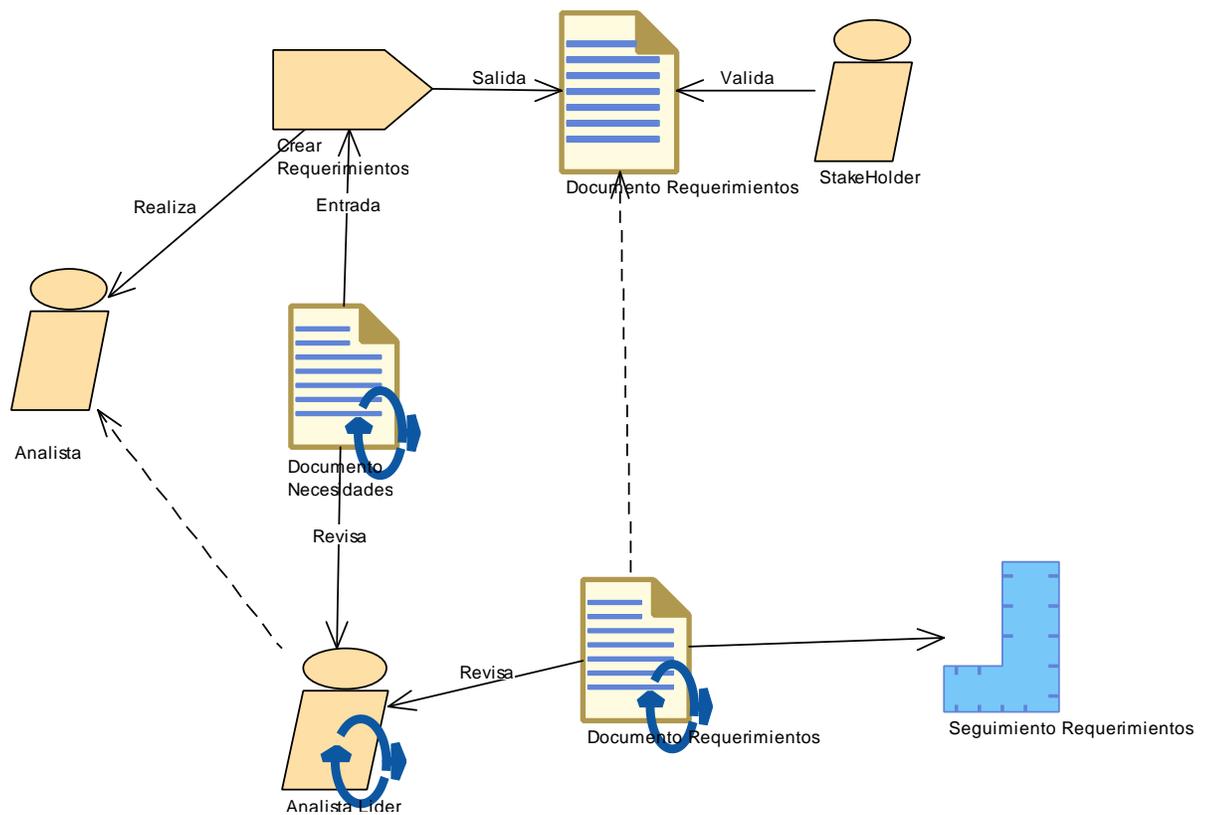


Figura 7.6 Tarea Crear requerimientos en SPEM

7.4. MÉTRICA PARA DEFINIR LA CALIDAD DEL LEVANTAMIENTO DE REQUERIMIENTOS

Para medir la calidad, se necesitan definir los parámetros a evaluar, una escala evaluadora y evaluadores.

En el modelo mostrado en la figura 7.7 se aprecia la existencia de 3 productos producidos por las distintas tareas realizadas, a estos productos se le definirán parámetros para medir el grado de calidad del producto.

7.4.1. PARAMETROS A EVALUAR

A. Documento Necesidades

- a. Explica el funcionamiento del negocio
- b. Explica los actores involucrados
- c. Explica cual es el objetivo de la aplicación
- d. Calidad del modelo de datos
- e. Definición de procesos involucrados
- f. Explica las necesidades del cliente
- g. Calidad de la priorización de necesidades

B. Documento Stakeholders

- a. Nivel de interés del Stakeholder
- b. Poder de decisión
- c. Nivel de compromiso
- d. Nivel de conocimiento del negocio
- e. Nivel de experiencia en el negocio

C. Documento Requerimientos

- a. Completitud
- b. Pertinencia
- c. Consistencia
- d. Medibilidad
- e. Precisión
- f. Factibilidad
- g. Entendibilidad
- h. Trazabilidad
- i. Estructura

j. Modificabilidad

Los elementos del punto C son explicados en el punto 7.2.3

7.4.2. ESCALA EVALUADORA

Para evaluar cada parámetro y ponderar se utilizará una escala ordinal del 1 al 5, la cual se describe como la siguiente:

1: Muy Bajo	2: Bajo	3: Medio	4: Alto	5: Muy Alto
-------------	---------	----------	---------	-------------

Donde estos niveles medirán la calidad del parámetro evaluado por el evaluador correspondiente.

7.4.3. EVALUADORES

Todos los parámetros deben ser evaluados por el analista líder y los Stakeholders. Se necesitan los dos puntos de vista, los Stakeholders por parte del cliente y el analista líder por parte de la institución desarrolladora del producto, para mantener objetividad y balancear los resultados en caso de subjetividad. En caso de que existan varios Stakeholders es necesario que realicen la evaluación en conjunto

7.4.4. MÉTRICA FINAL

Una vez que se han evaluado todos los parámetros es necesario calcular el promedio ponderado según el peso que cada evaluador colocó a cada uno de los parámetros, para obtener el nivel de calidad de cada producto desarrollado.

Por ejemplo, para el caso del producto documento de necesidades, se presenta la tabla 7.1

Parámetro	Peso	Analista Líder	Peso	Stakeholders
Explica el funcionamiento del negocio	2	4	5	3
Explica los actores involucrados	3	5	2	2
Explica cual es el objetivo de la aplicación	3	2	3	5
Calidad del modelo de datos	5	2	2	3
Definición de procesos involucrados	5	3	4	4
Explica las necesidades del cliente	2	4	5	3
Calidad de la priorización de necesidades	3	1	5	2
PROMEDIO		2,82		3,11

Tabla 7.1 Ejemplo: evaluación de calidad del producto documento necesidades

Si promediamos ambos promedios (Promedio del Analista Líder y Promedio de los Stakeholders), se obtiene un promedio de 2,96. Esto da una calidad del producto de un nivel medio.

Al momento de obtener el resultado el analista líder debe evaluar si el producto es apto para seguir con el siguiente proceso, la implementación del sistema, o debe corregir el producto para luego realizar nuevamente la evaluación.

8. DESARROLLO DE SOFTWARE CON SCRUM EN SPEM

Una vez que el documento de requerimientos está terminado y los Stakeholders están definidos, se puede comenzar con el proceso de implementación del software.

8.1. DEFINICIÓN DE ROLES

8.1.1. SCRUMMASTER

El ScrumMaster es el responsable de asegurar que el equipo se adhiera a los valores, prácticas, y normas de Scrum. El ScrumMaster es el encargado de entrenar y enseñar al equipo para realizar un producto de calidad de manera eficiente y eficaz. También ayuda al equipo a entender y utilizar la autogestión, sin embargo no maneja el equipo de desarrollo, el equipo utiliza auto-organización, es decir el equipo se organiza de tal manera que mejor permita desempeñar la obra. [6]

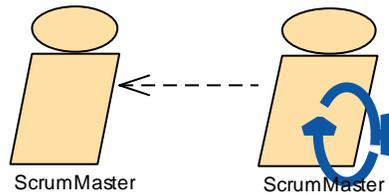


Figura 8.1 Definición del Rol de Uso ScrumMaster en SPEM

Además el ScrumMaster necesita ser un referente dentro del equipo, un compañero especial, para ello debe ser y parecer un líder. Un líder que sepa dirigir al equipo, transmita confianza, seguridad y valores de sacrificio por el equipo. Debe poseer el don de la empatía, saber estar a la altura de cada uno de sus miembros, poder armonizar y canalizar la energía individual para que exista sinergia, y se produzca un beneficio general.

No existe una opción verdadera, sino que se debe elegir la opción que se considere mejor para el equipo.

8.1.2. PRODUCT OWNER

El product owner puede ser una persona por parte del cliente o dentro de la empresa de desarrollo. Se recomienda que el product owner no sea parte del equipo de desarrollo, ni tampoco ser ScrumMaster, ya que sus intereses pueden ser comprometidos [7].

Como product owner, este representa al cliente, además de ser el encargado de negociar con el equipo y con el Scrum Master, la prioridad del trabajo a realizar.

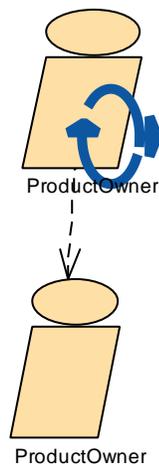


Figura 8.2 Definición del Rol de Uso ProductOwner en SPEM

Los roles del product owner son:

- Definir las características del producto
- Decidir sobre las fechas de lanzamiento
- Ser responsable de la rentabilidad del producto
- Priorizar las características según valor del mercado
- Ajustar las características y prioridades por iteración
- Aceptar o rechazar resultados del trabajo

8.1.3. EQUIPO

El tamaño óptimo para un equipo es siete personas, más o menos dos. Cuando hay menos de cinco miembros del equipo, hay menos interacción y el aumento de la productividad como un resultado menos. Es más, el equipo puede encontrar limitaciones de habilidades en las partes para el desarrollo del Sprint e incapaz de poder entregar parte del producto. Si hay más de nueve

miembros, simplemente se necesita demasiada coordinación. Los equipos grandes generan demasiada complejidad para que un proceso empírico se pueda manejar.

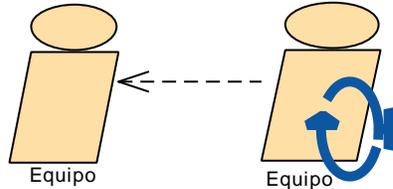


Figura 8.3 Definición del Rol de Uso Equipo en SPEM

La Composición del equipo puede cambiar al final de un Sprint. Cada vez que la membrecía del equipo sea cambiada, la productividad ganada por la auto-organización es disminuida. Por esto se debe tener cuidado cuando se cambia la composición del equipo.

8.2. DEFINICIÓN DE TAREAS

8.2.1. DEFINIR PRODUCT OWNER

El proceso de selección del product owner no es una tarea trivial ya que al momento de conversar con el cliente tiene que actuar como interlocutor del equipo de desarrollo y cuando conversa con el equipo de desarrollo debe actuar como interlocutor del cliente, es por esto que en la definición del product owner éste debe ser una persona con las capacidades para realizar esta tarea.

El product owner debe ser capaz de canalizar las necesidades del negocio, sabiendo escuchar a las partes interesadas en el producto y transmitir las “en objetivos de valor para el producto”, al equipo. Además de maximizar el valor para el negocio con respecto al retorno de la inversión, abogando por los intereses del negocio, revisar el producto e ir adaptándole sus funcionalidades, analizando las mejoras que estas puedan entregar un mayor valor para el negocio.[10]

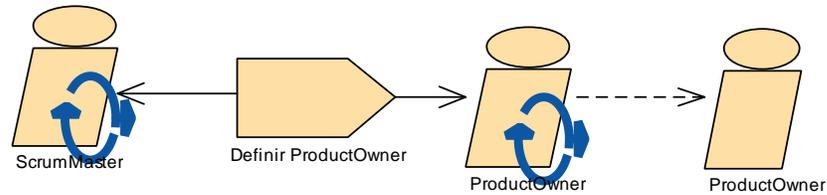


Figura 8.4 Tarea Definir Product Owner en SPEM

Por lo tanto el Product Owner debe tener aptitudes tales como: Excelente facilidad de comunicación en las relaciones interpersonales, excelente conocimiento del negocio, facilidad de análisis de relaciones costo beneficio, visión de negocios.

8.2.2. REUNIÓN PLAN LANZAMIENTO

El propósito de planificación y lanzamiento es el de establecer un plan y objetivos que los equipos de Scrum y el resto de las organizaciones puedan entender y comunicarse. La planificación de lanzamiento responde las preguntas: ¿Cómo se puede convertir la visión en un producto ganador en las mejores maneras posibles? ¿Cómo se puede cumplir o exceder la satisfacción del cliente para que aumente el retorno de su inversión?

El plan de lanzamiento establece el objetivo del lanzamiento, los riesgos principales, y las características generales y la funcionalidad que el lanzamiento contendrá. Asimismo, establece una fecha de entrega y el costo probable si no cambia nada. El cliente puede inspeccionar el avance y realizar cambios a este plan de lanzamiento sprint por sprint.

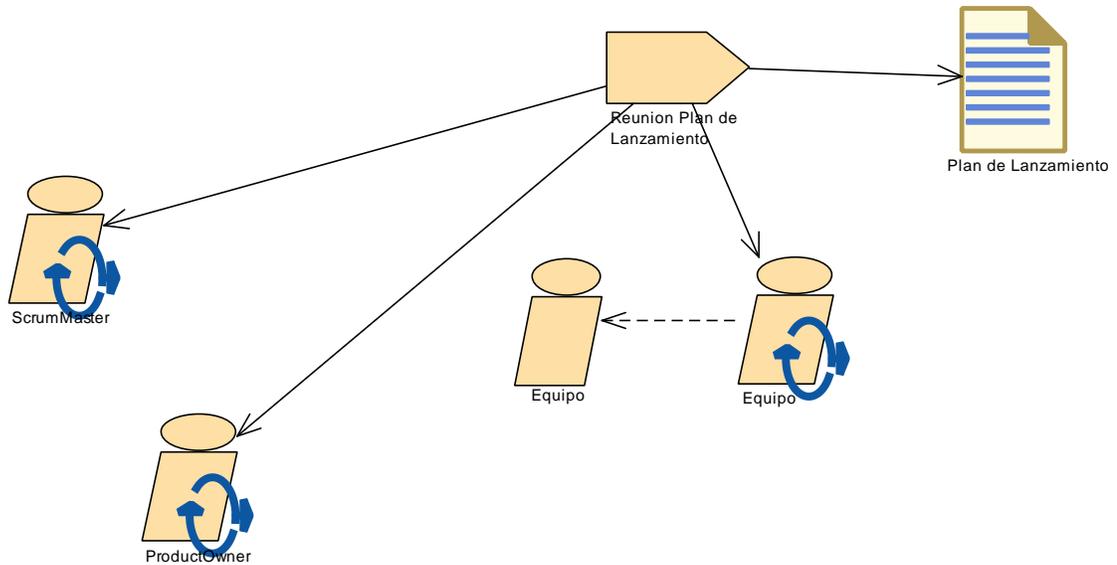


Figura 8.5 Tarea Reunión Plan de Lanzamiento en SPEM

8.2.3. DEFINICIÓN Y PLANIFICACIÓN DE SPRINTS

La reunión de planificación de Sprint es un encuentro que consiste en definir cómo se va a construir la funcionalidad en un incremento del producto durante el sprint. Existen dos partes en la planificación del sprint, el ¿Qué hacer? y el ¿Cómo se hará?

En la primera parte el product owner presenta los requerimientos prioritarios y en conjunto con el equipo de desarrollo planifican que funcionalidad se desarrollará en el siguiente sprint.

La segunda parte de la planificación, el equipo define la arquitectura y el diseño de la funcionalidad que fue seleccionada para posteriormente definir el trabajo o tareas que conformarán la pila de requerimientos del sprint, para construir dicha funcionalidad durante el próximo sprint

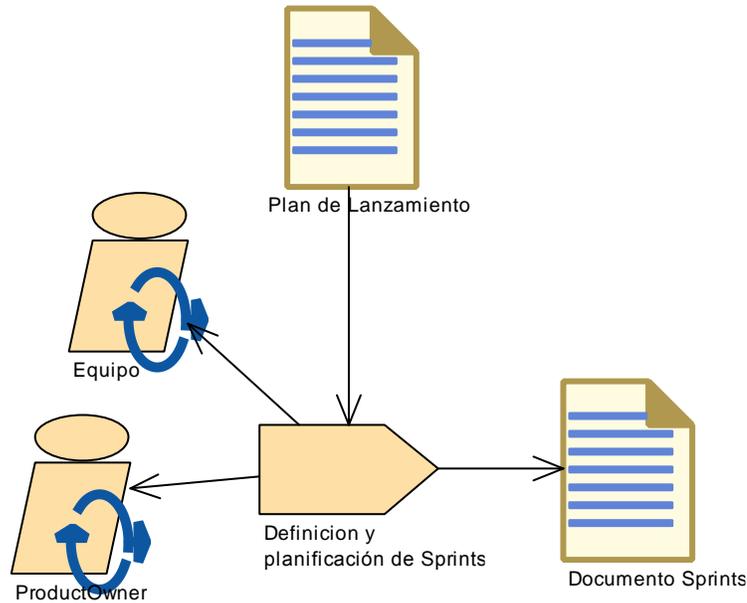


Figura 8.6 Tarea Definición y planificación de Sprints en SPEM

La reunión de planificación de sprint debe tener como salida la pila de requerimientos para el sprint, la duración del sprint, fecha y lugar de la reunión de revisión, los objetivos del sprint, fija las reuniones diarias (hora y lugar).

Los beneficios de esta planificación es que todos los miembros del equipo tienen la misma visión del objetivo y se ha utilizado los conocimientos y experiencias de todos para elaborar la mejor solución entregable en el mínimo tiempo y con el mínimo esfuerzo, eliminando tareas innecesarias, detectando conflictos y dependencias entre tareas, estas razones generan un aumento en la productividad.

Además, potencia el compromiso del equipo sobre el objetivo común de la iteración, ya que es todo el equipo quien asume la responsabilidad de completar en la iteración los requisitos que selecciona. Facilita la ayuda de cualquier miembro si se detecta algún impedimento que bloquea el progreso de la iteración, especialmente si cuando se está llegando al final de la iteración es necesaria la participación de todos para poder completar los objetivos comprometidos. Es cada una de las personas la que se responsabiliza de realizar sus tareas (a las que se auto asignó) en los tiempos que proporcionó. Si existe falta de compromiso con respecto al resto de los miembros del equipo se hará muy evidente en las reuniones diarias.

Para finalizar, genera una estimación más confiable, dado que tiene en cuenta los diferentes conocimientos, experiencias y habilidades de los integrantes del equipo

8.2.4. REUNIÓN DIARIA

Esta reunión tiene una duración estimada de 15 minutos, durante esta reunión cada integrante del equipo explica lo que ha logrado desde la última reunión, lo que va a realizar antes de la próxima reunión y cuáles han sido los obstáculos que ha tenido.

Estas reuniones ayudan a mejorar la comunicación del equipo, identificar y eliminar los obstáculos, destacar y promover la rápida toma de decisiones, y mejorar el nivel de conocimiento de todos los proyectos.

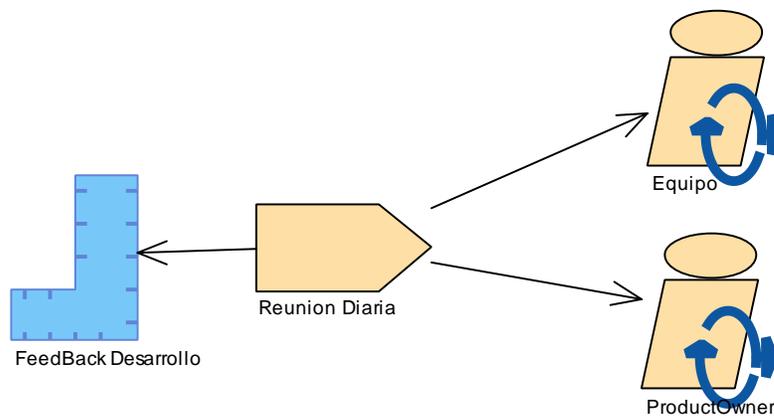


Figura 8.7 Tarea Reunión Diaria en SPEM

Es responsable del ScrumMaster velar por que estas reuniones se realicen y mantengan en el tiempo, además de enseñar cómo realizar la reunión en un período corto de tiempo, como también asegurarse que los integrantes del equipo hablen brevemente.

8.2.5. DESARROLLO

El objetivo del desarrollo es finalizar un sprint, basándose en el documento de planificación del sprint. Una vez finalizado el desarrollo se obtiene el producto sprint, el cual es un incremento del producto final.

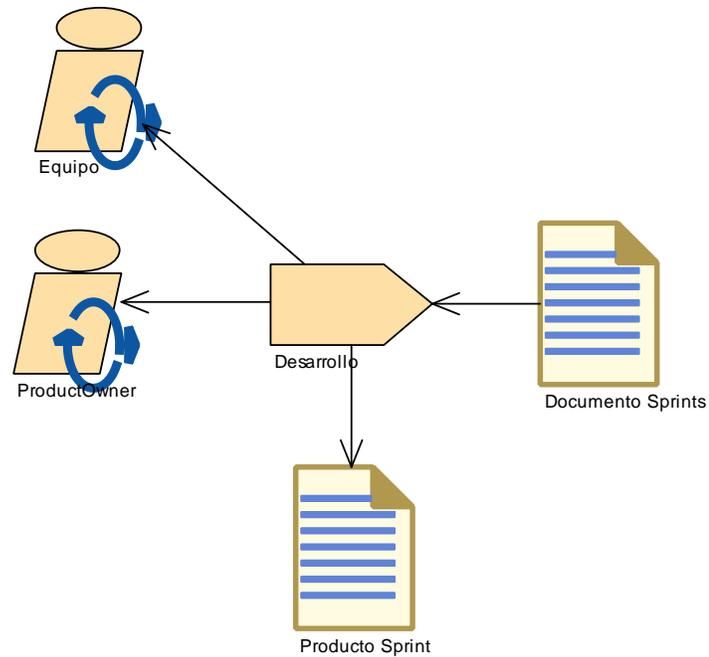


Figura 8.8 Tarea Desarrollo en SPeM

8.2.6. REVISIÓN DE SPRINT

La Revisión del Sprint brinda una inspección del progreso del proyecto al finalizar cada Sprint. Basándose en esta inspección se realizan adaptaciones al proyecto. El equipo estimó hasta dónde llegará al finalizar el Sprint, y fija el curso acorde. Al fin del Sprint el equipo presenta el Incremento Del Producto que pudo construir. El gerente, clientes, usuarios y el Dueño Del Producto verifican el Incremento Del Producto. Escuchan las historias del equipo que surgieron durante el transcurso del Sprint. Escuchan qué salió mal y qué salió bien. Verifican en dónde realmente se encuentran en la construcción de todo el sistema. Luego de todo esto, finalmente pueden tomar una decisión con información acerca de qué hacer a continuación. En otras palabras, determinan el mejor curso de acción para poder alcanzar el destino.

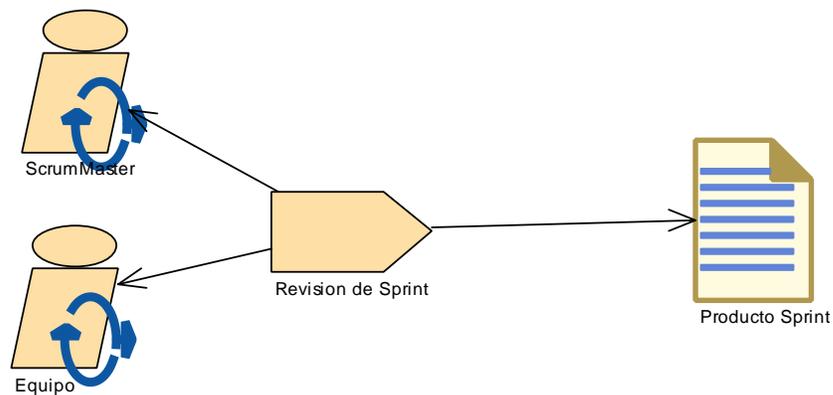


Figura 8.9 Tarea Revisión de Sprint en SPEM

Durante la reunión, todos ven funcionar la demostración del producto en el entorno del cliente o usuario. A medida que se visualiza, consideran qué funcionalidad podría ser agregada durante el próximo Sprint. El incremento del producto es el punto inicial para una lluvia de ideas. Por ejemplo, alguien podría sugerir luego de ver la demostración: "Si hiciéramos el control de costos del paciente a mano, podrías comenzar a usar el sistema ahora", o "Esto podría resolver los problemas de seguimiento del inventario en los distritos. ¿Qué faltaría hacer para que el sistema tome la información de la base de datos?".

A medida que el equipo muestra el incremento del producto, ayuda a los espectadores a entender las fortalezas y debilidades del producto, y las dificultades y éxitos de la experiencia de trabajar juntos.

El proceso completo se puede apreciar en la figura 8.10

8.4. MÉTRICA PARA DEFINIR LA CALIDAD DEL DESARROLLO DE SOFTWARE EN SCRUM

Para medir la calidad, se necesitan definir los parámetros a evaluar, una escala evaluadora y evaluadores.

En el modelo mostrado en la figura 8.10 se aprecia la existencia de 3 productos producidos por las distintas tareas realizadas, a estos productos se le definirán parámetros para medir el grado de calidad del producto.

8.4.1. PARÁMETROS A EVALUAR

A. Plan de lanzamiento

- a. Definición de los objetivos
- b. Definición del plan
- c. El plan está bien relacionado con los objetivos
- d. Definición de riesgos
- e. Definición de funcionalidades
- f. Definición de fechas de entrega
- g. Definición costo probable
- h. Posee escalabilidad

B. Documento Sprints

- a. Definición de lo que se va a realizar en el sprint
- b. Definición de cómo se hará el sprint
- c. Definición de funcionalidades prioritarias
- d. Arquitectura de la aplicación
- e. Fecha de reuniones
- f. Definición de objetivos

C. Producto Sprints

- a. Funcionalidad
- b. Usabilidad
- c. Fiabilidad
- d. Rendimiento
- e. Soporte

8.4.2. ESCALA EVALUADORA

Para evaluar cada parámetro y ponderar se utilizará una escala ordinal del 1 al 5, la cual se describe como la siguiente:

1: Muy Bajo	2: Bajo	3: Medio	4: Alto	5: Muy Alto
-------------	---------	----------	---------	-------------

Donde estos niveles medirán la calidad del parámetro evaluado por el evaluador correspondiente.

8.4.3. EVALUADORES

Todos los parámetros deben ser evaluados por el ScrumMaster y el ProductOwner. Se necesitan los dos puntos de vista, los Product Owner por parte del cliente y el ScrumMaster por parte de la institución desarrolladora del producto, para mantener objetividad y balancear los resultados en caso de subjetividad.

8.4.4. MÉTRICA FINAL

Una vez que se han evaluado todos los parámetros es necesario calcular el promedio ponderado según el peso que cada evaluador colocó a cada uno de los parámetros, para obtener el nivel de calidad de cada producto desarrollado.

Un ejemplo para el caso del producto sprint, se presenta la tabla 8.1

Si calculamos ambos promedios (Promedio del ScrumMaster y Promedio del ProductOwner), obtenemos un promedio de 3,4. Esto nos da una calidad del producto de un nivel medio.

Al momento de obtener el resultado el ProductOwner debe evaluar si el producto es apto para seguir con un nuevo incremento o generar nuevos requerimientos en la pila de Sprints, y así realizar posteriormente nuevamente la evaluación.

Parámetro	Peso	ScrumMaster	Peso	ProductOwner
Definición de lo que se va a realizar en el sprint	4	4	4	3
Definición de como se hará el sprint	5	5	1	2
Definición de funcionalidades prioritarias	4	4	2	1
Arquitectura de la aplicación	5	5	3	1
Fecha de reuniones	3	5	5	2
Explica las necesidades del cliente	3	5	5	3
Definición de objetivos	2	4	2	2
PROMEDIO		4,61		2,18

Tabla 8.1 Ejemplo de evaluación de la calidad del producto “Sprint”

9. PUESTA EN MARCHA

9.1. PROYECTO

9.1.1. EMPRESA

Trigensoft Ltda. Es una empresa joven que nace formalmente en mayo de 2007. Sin embargo, su historia y afán de emprendimiento comienza mucho antes, a fines de 2005, cuando sus integrantes fundadores deciden agruparse y se proponen un claro objetivo:

"Crear una empresa y a través de ella generar nuevas e innovadoras soluciones informáticas, orientadas a dar respuestas concretas a las necesidades del país".

En esa línea surgen múltiples alternativas de emprendimiento, sin embargo, la más atractiva e interesante de ellas nace en marzo de 2006, inserta en la participación de sus socios en la Feria del Software de la Universidad Técnica Federico Santa María. Instancia que además de fortalecer el espíritu emprendedor de Trigensoft, deriva en la incorporación de un nuevo integrante al equipo: Eduardo Silva.

Bajo este escenario, Trigensoft inicia un estricto proceso de investigación e identificación de oportunidades, donde los atributos más relevantes que debían estar presentes en la aplicación a desarrollar eran:

- Poseer potencial de comercialización.
- La necesidad no debía estar satisfecha por una tecnología previa.
- Ser un aporte para el ámbito de las tecnologías de la información nacionales.

En este contexto, y derivado de múltiples conversaciones mantenidas con médicos de la especialidad de ginecología y obstetricia, nace la idea de desarrollar Ecoinformat, un software para Windows diseñado especialmente para satisfacer las necesidades del segmento gineco-obstetra nacional y orientado a facilitar la realización de exploraciones obstétricas, ginecológicas y exámenes Doppler.

De este modo, Trigensoft inicia en marzo de 2006 el proceso de desarrollo de Ecoinformat, contando con la supervisión de los Doctores Fernando Amor Lillo y Humberto Vaccaro Cerva. Dicho proceso se extiende en su primera etapa hasta Octubre del mismo año y concluye con la participación destacada de los emprendedores en la Feria del Software de la Universidad Técnica Federico Santa María.

Adicionalmente, en Junio de 2006, Trigensoft se hace merecedor del Premio otorgado por Microsoft Chile y el Instituto Internacional para la Innovación Empresarial: "Concurso para

Emprendedores.NET Microsoft 2006”. Instancia dirigida a fomentar la innovación y el emprendimiento en la generación de nuevos negocios tecnológicos y que provee a Trigensoft de financiamiento para el desarrollo del primer prototipo de Ecoinformat.

Finalmente, durante el primer semestre de 2007, Trigensoft centró sus esfuerzos en la captación de recursos financieros adicionales para culminar el desarrollo y validación de Ecoinformat, así como potenciar su introducción futura al mercado. Ello se logró mediante la adjudicación del Fondo Concursable “Capital Semilla, Línea 1” de CORFO. A partir de dicho momento, hasta la fecha, los socios de Trigensoft han centrado su atención en la formalización de la empresa, ejecutar estudios de mercado, fortalecer el plan de negocios, proteger la propiedad intelectual del software, así como validarlo comercialmente. Todo ello ha concluido con éxito y coloca a los emprendedores en perspectiva para iniciar la etapa más importante de este proceso: el lanzamiento formal de Ecoinformat al mercado.

Actualmente Trigensoft es una empresa de desarrollo de software que accedió a participar en este proyecto con la única condición que no se revelaran los nombres personales de sus clientes, solo el nombre de la empresa.

9.1.2. CLIENTE

En la década de los '90, BancoEstado se interesó en buscar una solución a las dificultades que enfrentaban los microempresarios marginados del sistema financiero y que no tenían acceso a las herramientas necesarias para el desarrollo de sus negocios. La respuesta fue la creación de BancoEstado Microempresas S.A., Asesorías Financieras, filial que nace con el objetivo de promover la igualdad de oportunidades de financiamiento para todos los chilenos y chilenas, especialmente de los sectores de menores ingresos.

BancoEstado Microempresas atiende a trabajadores independientes o dueños de negocios que, con al menos un año de antigüedad, desarrollen actividades productivas, de comercio o de servicios, con un volumen de ventas inferior a las UF 2.400 al año (aprox. US\$ 100.000). Se especializa en atender, de manera integral, las necesidades de financiamiento de emprendedores que desarrollan sus negocios en los rubros de Comercio, Transporte de Pasajeros o Carga, Pequeña Agricultura, Pesca Artesanal, Servicios Profesionales o Técnicos Independientes, Manufactura, Artesanado y Turismo.

Hoy, BancoEstado Microempresas se encuentra con la satisfacción de ser líder del sector que atiende y contar con los más altos estándares de calidad en sus procesos y atención. La experiencia inédita de desarrollar, desde un banco público, un programa masivo, eficiente y rentable, con la misión de brindar acceso financiero a miles de microempresarios y trabajadores independientes del país, ha transformado a BancoEstado Microempresas en un referente en la industria de las microfinanzas tanto a nivel nacional como internacional. Por ello ha recibido el

reconocimiento de importantes y prestigiosas organizaciones nacionales y extranjeras que han visto en su gestión un ejemplo a seguir [14].

9.1.3. DESCRIPCIÓN PROYECTO

BancoEstado Microempresas entrega la responsabilidad a Trigensoft de realizar mejoras al sistema actual de soporte para clientes internos del banco, el cual según los mismos usuarios presenta una serie de fallas que se necesitan reparar.

El objetivo principal del sistema es entregar soporte online a través de las diferentes sucursales bancarias a lo largo del país. Este soporte se entrega mediante el sistema de tickets, donde el usuario de alguna sucursal bancaria envía una pregunta con la posibilidad de adjuntar archivos, para luego ser procesada por algún operador de BancoEstado Microempresas y dar una respuesta al usuario.

BancoEstado Microempresas solicita a Trigensoft realizar unas mejoras al sistema y accedió para trabajar en este proyecto con la condición de que no se revelara ningún nombre de las personas participantes en este proyecto.

9.1.4. PARTICIPANTES Y ROLES

A continuación se describen las personas participantes en el proyecto y sus respectivos roles:

Nombre	Rol	Empresa
Eduardo Silva	Analista Terreno	Trigensoft
Mauricio Leiva	Analista Líder	Trigensoft
Sebastián Korner	Equipo Programador	Trigensoft
Carlos Orrego	ScrumMaster	Trigensoft
Persona1	StakeHolder	BancoEstado
Persona2	ProductOwner	BancoEstado

Tabla 9.1 Nombre y Rol de las personas que participan en el proyecto.

9.2. CAPACITACIÓN

Para que las personas participantes en este proyecto comprendieran que es Scrum, como usarlo y lograr ejecutar su rol de una manera excelente, se realizó una capacitación de 16 horas en la cual se tocaron los siguientes puntos.

- Introducción a Scrum
 - Breve Historia
 - Conceptos
 - Definición: Roles
 - Definición: Artefactos
 - Definición: Herramientas
 - Definición: Ciclo de Vida
- Planeación Ágil
 - Planeando el Sprint
 - Definiendo el criterio de fin de tareas
 - Comunicación
- Preparando un Sprint específico
 - Reuniones de planeación del Sprint
 - Facilitando el trabajo en equipo
 - Reunión diaria

10. RESULTADOS OBTENIDOS

BancoEstado, para este proyecto, accedió a entregar la información de la evaluación realizada por sus funcionarios pero el documento final realizado era confidencial, a lo cual solo se mostrarán los resultados obtenidos y no el documento.

10.1. DOCUMENTO NECESIDADES

Luego de una primera visita por el analista en terreno (Eduardo Silva) para evaluar las necesidades de la empresa, se creó una primera versión del documento necesidades, el cual fue enviado a la gente de BancoEstado para su evaluación con la siguiente tabla.

En una primera iteración del documento necesidades estos, los resultados obtenidos se representan en la tabla 10.1

Parametro	Peso	Analista Líder (Mauricio Leiva)	Peso	Stakeholder (Persona 1)	Desv. Estandar Peso	Desv. Estandar Eval.
Explica el funcionamiento del negocio	5	2	3	1	1,41	0,51
Explica los actores involucrados	4	1	1	1	2,12	0
Explica cual es el objetivo de la aplicación	3	1	5	1	1,41	0
Calidad del modelo de datos	5	2	2	1	2,12	0,54
Definición de procesos involucrados	3	1	4	1	0,7	0
Explica las necesidades del cliente	3	1	5	1	1,41	0
Calidad de la priorización de necesidades	4	1	4	1	0	0
PROMEDIO		1,37		1	1,31	0,15

Tabla 10.1 Primera iteración análisis de calidad documento necesidades

Observando la tabla 10.1 claramente se observa que el documento necesidades no satisface las necesidades de BancoEstado y tampoco las de Trigensoft. Se puede observar claramente como una gran cantidad de desviaciones resulta con valor cero, lo cual se observa que ambos evaluadores estuvieron de acuerdo en varios puntos en el nivel de calidad del documento. Como consecuencia, Eduardo Silva procede nuevamente a realizar el documento necesidades.

Al observar las desviaciones estándar del peso y de la evaluación, se aprecia que no hay concordancia entre los evaluadores con respecto al peso, lo cual demuestra diferencias de opiniones entre los evaluadores en los niveles de importancia de cada uno de los atributos.

La tabla 10.2 muestra una mejoría en la calidad del documento, pero hubo una falta en la calidad de la priorización de necesidades, a lo cual el analista en terreno (Eduardo Silva) tuvo que mejorar solo la priorización de las necesidades para entregar un documento de un nivel de calidad bueno.

Parámetro	Peso	Analista Líder (Mauricio Leiva)	Peso	Stakeholder (Persona 1)	Desv. Estandar Peso	Desv. Estandar Eval.
Explica el funcionamiento del negocio	5	3	3	3	1,41	0
Explica los actores involucrados	4	4	1	3	2,12	0,58
Explica cual es el objetivo de la aplicación	3	2	5	3	1,41	0,51
Calidad del modelo de datos	5	3	2	3	2,12	0
Definición de procesos involucrados	3	4	4	3	0,7	0,5
Explica las necesidades del cliente	3	3	5	3	1,41	0
Calidad de la priorización de necesidades	4	1	4	1	0	0
PROMEDIO		2,85		2,66	1,31	0,22

Tabla 10.2 Segunda iteración análisis de calidad documento necesidades

Luego de realizar una tercera iteración, se obtiene un promedio de calidad sobre 3 (ver tabla 10.3) el analista líder decide que es un documento de necesidades que cumple con las necesidades del cliente y se procede a realizar el documento de Stakeholder. La gran mayoría de las desviaciones estándar de la evaluación son menores a 1, esto demuestra que los evaluadores estuvieron en un alto grado de concordancia en la evaluación.

Parámetro	Peso	Analista Líder (Mauricio Leiva)	Peso	Stakeholder (Persona 1)	Desv. Estandar Peso	Desv. Estandar Eval.
Explica el funcionamiento del negocio	5	3	3	3	1,41	0
Explica los actores involucrados	4	4	1	3	2,12	0,58
Explica cual es el objetivo de la aplicación	3	3	5	3	1,41	0
Calidad del modelo de datos	5	3	2	3	2,12	0
Definición de procesos involucrados	3	4	4	3	0,7	0,5
Explica las necesidades del cliente	3	3	5	3	1,41	0
Calidad de la priorización de necesidades	4	3	4	3	0	0
PROMEDIO		3,25		3	1,31	0,15

Tabla 10.3 Tercera iteración análisis de calidad documento necesidades

10.2. DOCUMENTO STAKEHOLDER

El analista en terreno al enterarse de que se está evaluando la calidad del producto, consultó cuales serán los parámetros a evaluar, para con ello no realizar varias iteraciones del producto.

Parámetro	Peso	Analista Líder (Mauricio Leiva)	Peso	Stakeholder (Persona 1)	Desv. Estandar Peso	Desv. Estandar Eval.
Nivel de interés de los Stakeholders	2	2	3	4	0,7	1
Poder de Decisión	4	3	1	3	2,12	0
Nivel de Compromiso	4	2	1	4	2,12	1,16
Nivel de Conocimiento del Negocio	5	3	3	2	1,41	0,5
Nivel de experiencia en el Negocio	2	4	2	4	0	0
PROMEDIO		2,7		3,3	1,27	0,53

Tabla 10.4 Primera iteración análisis de calidad documento Stakeholder

A pesar que según los resultados de la tabla se observó un nivel de calidad bueno, el Analista Líder (Mauricio Leiva) solicitó al Analista en Terreno realizar el documento con más detalle, solicitó que se observen más argumentos para ver el nivel de compromiso y el nivel de interés de los Stakeholders, a pesar de que BancoEstado quedó conforme con estos puntos del documento

Al observar la desviación estándar del peso se puede apreciar que solo hubo concordancia en la importancia del atributo Nivel de experiencia del negocio, ambos opinaron que no era tan relevante como otros puntos. Pero también pensaron muy diferente en cuanto al nivel de importancia del Poder de decisión y nivel de compromiso.

Parámetro	Peso	Analista Líder (Mauricio Leiva)	Peso	Stakeholder (Persona 1)	Desv. Estandar Peso	Desv. Estandar Eval.
Nivel de interés de los Stakeholders	2	4	3	4	0,7	0
Poder de Decisión	4	3	1	3	2,12	0
Nivel de Compromiso	4	3	1	4	2,12	0,58
Nivel de Conocimiento del Negocio	5	3	3	2	1,41	0,51
Nivel de experiencia en el Negocio	2	4	2	4	0	0
PROMEDIO		3,2		3,3	1,27	0,21

Tabla 10.5 Segunda iteración análisis de calidad documento Stakeholder

Con estos resultados el Analista Líder quedó conforme con el documento de Stakeholders y tomó la decisión de proceder a la siguiente tarea que es realizar el documento de requerimientos.

Al observar las desviaciones estándar de la evaluación, se aprecia que hubo una cercanía de opiniones con respecto a la calidad del atributo, lo cual demuestra una concordancia de opiniones entre el analista líder y el stakeholder.

10.3. DOCUMENTO REQUERIMIENTOS

El Analista en Terreno (Eduardo Silva) confeccionó el documento de requerimientos basándose en los puntos que se evaluarán en la calidad.

Parámetro	Peso	Analista Líder (Mauricio Leiva)	Peso	Stakeholder (Persona 1)	Desv. Estandar Peso	Desv. Estandar Eval.
Compleitud	5	3	2	4	2,12	0,54
Pertinencia	4	3	2	4	1,41	0,52
Consistencia	4	2	3	3	0,7	0,50
Medibilidad	5	2	1	4	2,82	1,2
Precisión	3	2	5	4	1,41	1,03
Factibilidad	5	1	4	3	0,7	1
Entendibilidad	1	4	5	3	2,82	0,6
Trazabilidad	3	3	2	2	0,7	0,5
Estructura	2	4	3	3	0,7	0,5
Modificabilidad	3	3	1	2	1,41	0,55
PROMEDIO		2,45		3,2	1,47	0,71

Tabla 10.6 Primera iteración análisis de calidad documento requerimientos

El analista consideró que el documento estaba parcialmente bien pero que existían requerimientos que no eran factibles por el presupuesto con el cual constaba BancoEstado para esta etapa del proyecto, por lo cual es necesario eliminar algunos requerimientos que no sean prioridad para BancoEstado. Este punto el Stakeholder lo evaluó de buena manera porque no conoce el presupuesto total ya que el presupuesto es administrado por el personal de finanzas de BancoEstado.

Parámetro	Peso	Analista Líder (Mauricio Leiva)	Peso	Stakeholder (Persona 1)	Desv. Estandar Peso	Desv. Estandar Eval
Compleitud	5	3	2	4	2,12	0,54
Pertinencia	4	3	2	4	1,41	0,52
Consistencia	4	3	3	3	0,7	0
Medibilidad	5	3	1	4	2,82	0,6
Precisión	3	3	5	4	1,41	0,51
Factibilidad	5	4	4	4	0,7	0
Entendibilidad	1	4	5	3	2,82	0,6
Trazabilidad	3	3	2	3	0,7	0
Estructura	2	4	3	3	0,7	0,5
Modificabilidad	3	4	1	3	1,41	0,55
PROMEDIO		3,3		3,5	1,47	0,38

Tabla 10.7 Segunda iteración análisis de calidad documento requerimientos

El analista líder revisó nuevamente el documento y cambió varios aspectos para obtener una mejor evaluación, ya que para Trigensoft este documento, es el más importante antes de proceder a la etapa de desarrollo. La evaluación de la calidad del documento subió de una manera que el Analista Líder accedió a comenzar con el desarrollo del sistema.

10.4. PLAN DE LANZAMIENTO

Una vez que ya se aprobaron los requerimientos comienza la metodología de desarrollo en Scrum la cual tendrá a Carlos Orrego como ScrumMaster y Persona 2 como Product Owner. En una primera iteración del plan de lanzamiento realizado por el equipo de desarrollo a cargo del ScrumMaster.

Parámetro	Peso	ScrumMaster (Carlos Orrego)	Peso	ProductOwner (Persona 2)	Desv. Estándar Peso	Desv. Estándar Eval.
Definición de los objetivos	2	2	4	1	1,41	0,52
Definición del plan	4	2	2	1	1,41	0,52
El plan está bien relacionado con los objetivos	3	1	2	2	0,70	0,5
Definición de riesgos	5	1	4	1	0,70	0
Definición de funcionalidades	5	4	5	3	0	0,5
Definición de fechas de entrega	5	4	5	2	0	1
Definición costo probable	5	2	5	3	0	0,5
Posee Escalabilidad	3	2	1	2	1,41	0
PROMEDIO		2,37		2	0,7	0,44

Tabla 10.8 Primera iteración análisis de calidad plan de lanzamiento

Los resultados en una primera iteración son insatisfactorios para el ScrumMaster tanto como para el ProductOwner, en lo que es definición de riesgos. Ambos coinciden en que el plan de contingencia no satisface todos los riesgos posibles que tiene el proyecto. Es por esto que se realizó una segunda iteración del plan de lanzamiento en que el ScrumMaster tuvo un nivel de participación alto, ya que existía un par de días de atraso en el proyecto y no quería realizar más iteraciones del mismo documento.

Parámetro	Peso	ScrumMaster (Carlos Orrego)	Peso	ProductOwner (Persona 2)	Desv. Estandar Peso	Desv. Estandar Eval.
Definición de los objetivos	2	4	4	4	1,41	0
Definición del plan	4	4	2	3	1,41	0,52
El plan está bien relacionado con los objetivos	3	4	2	4	0,70	0
Definición de riesgos	5	4	4	4	0,70	0
Definición de funcionalidades	5	4	5	3	0	0,5
Definición de fechas de entrega	5	4	5	4	0	0
Definición costo probable	5	4	5	4	0	0
Posee Escalabilidad	3	4	1	4	1,41	0
PROMEDIO		4		3,75	0,7	0,12

Tabla 10.9 Segunda iteración análisis de calidad plan de lanzamiento

En esta segunda iteración se observa que el ScrumMaster considera un documento de alto nivel de calidad lo cual se ratifica por el ProductOwner.

10.5. DOCUMENTO SPRINTS

El equipo en conjunto con el ProductOwner realizan el documento Sprints en el cual va el conjunto de requerimientos ordenados por prioridad por el ProductOwner. El resultado de una primera iteración del documento se muestra en la Tabla 10.10

Parámetro	Peso	ScrumMaster (Carlos Orrego)	Peso	ProductOwner (Persona 2)	Desv. Estandar Peso	Desv Estandar Eval.
Definición de lo que se va a realizar en el sprint	5	2	3	2	1,41	0
Definición de como se hará el sprint	4	1	1	1	2,12	0
Definición de funcionalidades prioritarias	4	2	3	3	0,7	0,5
Arquitectura de la aplicación	5	1	2	3	2,12	1,08
Fecha de reuniones	3	1	5	4	1,41	1,54
Explica las necesidades del cliente	3	4	5	2	1,41	1,03
Definición de objetivos	3	1	5	2	1,41	0,51
PROMEDIO		1,66		1,70	1,51	0,66

Tabla 10.10 Primera iteración análisis de calidad documento Sprints

El ScrumMaster consideró que las fechas de reuniones para revisiones de un Sprint eran irreales y no se podían cumplir. Por su parte el ProductOwner no encontró que quedarán explícitamente claras las necesidades del cliente. Además, ambos consideraron que no estaba bien definido por el equipo de desarrollo la arquitectura del sprint además de la planificación. Por estas razones se decide por el ScrumMaster realizar una nueva iteración al documento.

Parámetro	Peso	ScrumMaster (Carlos Orrego)	Peso	ProductOwner (Persona 2)	Desv. Estandar Peso	Desv. Estandar Eval.
Definición de lo que se va a realizar en el sprint	5	3	3	3	1,41	0
Definición de como se hará el sprint	4	2	1	3	2,12	0,5
Definición de funcionalidades prioritarias	4	3	3	4	0,7	0,5
Arquitectura de la aplicación	5	2	2	4	2,12	1,08
Fecha de reuniones	3	3	5	4	1,41	0,51
Explica las necesidades del cliente	3	3	5	4	1,41	0,51
Definición de objetivos	3	3	5	3	1,41	0
PROMEDIO		2,6		2	1,51	0,44

Tabla 10.11 Segunda iteración análisis de calidad documento Sprints

En esta segunda iteración el ProductOwner considera que el documento se encuentra en óptimas condiciones como para comenzar el desarrollo, pero en este caso el ScrumMaster decide que no es conveniente, porque aún la arquitectura está deficiente, y él considera el argumento, que una buena arquitectura antes del desarrollo es la clave para obtener un producto de calidad. Por esta razón el ScrumMaster decide dar una tercera iteración al documento, sacrificando tiempo de desarrollo en esto.

Parámetro	Peso	ScrumMaster (Carlos Orrego)	Peso	ProductOwner (Persona 2)	Desv. Estandar Peso	Desv. Estandar Eval.
Definición de lo que se va a realizar en el sprint	5	3	3	3	1,41	0
Definición de como se hará el sprint	4	4	1	4	2,12	0
Definición de funcionalidades prioritarias	4	3	3	4	0,7	0,5
Arquitectura de la aplicación	5	4	2	4	2,12	0
Fecha de reuniones	3	4	5	4	1,41	0
Explica las necesidades del cliente	3	3	5	4	1,41	0,51
Definición de objetivos	3	4	5	4	1,41	0
PROMEDIO		3,55		2,12	1,51	0,14

Tabla 10.12 Tercera iteración análisis de calidad documento Sprints

En esta tercera iteración el ScrumMaster considera que está todo planificado de manera óptima para realizar el desarrollo y autoriza al equipo de desarrollo de comenzar la programación de los Sprints

10.6. PRODUCTO SPRINT

Para este proyecto hubo una pila con 10 Sprints para este caso en particular, solo se analizarán 3 productos, los 3 primeros y de mayor importancia para BancoEstado.

Parámetro	Peso	ScrumMaster (Carlos Orrego)	Peso	ProductOwner (Persona 2)	Desv. Estandar Peso	Desv. Estandar Eval.
Funcionalidad	5	4	5	3	0	0,5
Usabilidad	3	3	5	3	1,41	0
Fiabilidad	5	3	5	4	0	0,5
Rendimiento	3	3	3	3	0	0
Soporte	2	4	5	3	2,12	0,54
PROMEDIO		3,38		3,21	0,7	0,3

Tabla 10.13 Análisis de calidad producto sprint N°1 primera iteración

Se observa en la tabla 10.13, que el producto de mayor importancia, en su primera iteración se obtuvo un óptimo resultado de calidad dejando conforme al ProductOwner y al ScrumMaster, con esto el equipo de desarrollo procede a realizar el sprint N°2.

Parámetro	Peso	ScrumMaster (Carlos Orrego)	Peso	ProductOwner (Persona 2)	Desv. Estandar Peso	Desv. Estandar Eval.
Funcionalidad	5	2	5	1	0	0,5
Usabilidad	3	2	5	1	1,41	0,51
Fiabilidad	5	1	5	2	0	0,5
Rendimiento	3	4	3	4	0	0
Soporte	2	4	5	3	2,12	0,54
PROMEDIO		2,27		2,04	0,7	0,41

Tabla 10.14 Análisis de calidad producto sprint N°2 primera iteración

Los resultados obtenidos (ver tabla 10.14) al momento de analizar funcionalidad, usabilidad y fiabilidad no fueron los mejores ya que el formulario de ingreso de datos contenía errores y además no tenía un formato ordenado con lo cual el nivel de usabilidad era deficiente, por estas razones se procede a realizar una nueva iteración de este sprint.

Una vez que se solucionaron los problemas de funcionalidad, usabilidad y fiabilidad que poseía este sprint el resultado se observa en la tabla 10.15.

Parámetro	Peso	ScrumMaster (Carlos Orrego)	Peso	ProductOwner (Persona 2)	Desv. Estandar Peso	Desv. Estandar Eval.
Funcionalidad	5	3	5	4	0	0,5
Usabilidad	3	4	5	3	1,41	0,51
Fiabilidad	5	3	5	3	0	0
Rendimiento	3	4	3	4	0	0
Soporte	2	4	5	3	2,12	0,54
PROMEDIO		3,44		3,34	0,7	0,31

Tabla 10.15 Análisis de calidad producto sprint N°2 segunda iteración

Con estos resultados obtenidos en calidad el ProductOwner y el ScrumMaster quedan conformes con lo cual se procede a realizar el sprint siguiente.

Al realizar el sprint N°3 los resultados se observan en la tabla 10.16:

Parámetro	Peso	ScrumMaster (Carlos Orrego)	Peso	ProductOwner (Persona 2)	Desv. Estandar Peso	Desv Estandar Eval.
Funcionalidad	5	4	5	4	0	0
Usabilidad	3	4	5	1	1,41	1,54
Fiabilidad	5	4	5	4	0	0
Rendimiento	3	4	3	1	0	1,5
Soporte	2	4	5	4	2,12	0
PROMEDIO		4		2,95	0,7	0,6

Tabla 10.16 Análisis de calidad producto sprint N°3 primera iteración

En este caso en particular el ScrumMaster consideró que el producto sprint N°3 era de un nivel de calidad excelente, pero el ProductOwner no consideró que tenía buena usabilidad y en el momento de presentar el sprint, este no tenía un rendimiento acorde a las necesidades del usuario. Por estas razones se decidió realizar una nueva iteración.

Al realizar una segunda iteración en el sprint N°3 los resultados fueron los siguientes:

Parámetro	Peso	ScrumMaster (Carlos Orrego)	Peso	ProductOwner (Persona 2)	Desv. Estandar Peso	Desv. Estandar Eval.
Funcionalidad	5	4	5	4	0	0
Usabilidad	3	4	5	3	1,41	0,51
Fiabilidad	5	4	5	4	0	0
Rendimiento	3	4	3	3	0	0,5
Soporte	2	4	5	4	2,12	0
PROMEDIO		4		3,65	0,7	0,2

Tabla 10.17 Análisis de calidad producto sprint N°3 segunda iteración

Con estos resultados en el producto Sprint N°3 las peticiones de usabilidad y rendimiento por parte del ProductOwner fueron cumplidas, con lo que se logra una satisfacción del cliente.

11. CONCLUSIÓN

Cuando se habla de calidad de software, siempre se hace mención al desarrollo del producto y las fases que éste tiene, pero nunca se toma en cuenta el proceso de los agentes externos al proceso de ingeniería del desarrollo del producto.

Por otra parte Scrum ayuda directamente al proceso de desarrollo de software dando una gestión regular de las expectativas del cliente, entregando resultados anticipados, flexibilidad y adaptación respecto a las necesidades del cliente, mitigación sistemática de los riesgos.

Gracias al uso de SPEM podemos mejorar en los procesos de software la comprensión y comunicación humana, al utilizar modelos que son comprendidos por cualquier persona. Además de facilitar la reutilización de procesos, al estar documentados de una manera simple, estos se pueden reusar en algún otro proceso de software parecido. También dar soporte a la gestión y mejora de procesos, guiar la automatización humana y dar soporte para la ejecución informática del proceso.

SPEM tiene un enfoque al proceso, más que al desarrollo del producto en específico, pero por ser una metodología de modelado relativamente nueva, en la actualidad no existen parámetros que definan si el proceso modelado con SPEM es de calidad o no, lo cual se propone como solución.

Gracias al método de medición desarrollado, se puede medir calidad “just in time”, o sea a medida que van terminando las tareas, se puede evaluar con que calidad se ha realizado esta, sin entorpecer el flujo del proceso. Dando al equipo de trabajo la posibilidad de mejorar la forma de realizar la tarea, ya que existirá un aprendizaje al evaluar el producto.

Observando los datos obtenidos se puede dar un buen grado de satisfacción al cliente, ya que este tiene participación en la evaluación de calidad, además de participar constantemente en el proceso del desarrollo del producto. Este método ayuda a acercar al cliente con el equipo de desarrollo aumentando la calidad del producto final.

El producto final obtenido es de calidad, ya que satisface las necesidades del cliente, cumple con los requerimientos funcionales, y lo más importante es que el cliente considera que es un producto de calidad, debido a que participó en las evaluaciones de calidad aceptando el producto solo si poseía un nivel de calidad buena o mejor.

Además se observa que las desviaciones estándar en cada iteración tienden a disminuir y los promedios de calidad aumentan, lo que significa que ambos evaluadores concuerdan en que la calidad del producto va en aumento.

Para finalizar, se definieron parámetros para medir calidad del software y sus respectivos indicadores, además de un método para mejorarlos. Este método se implementó en una empresa, y post análisis realizado, se encontró con que la calidad del producto fue en aumento a medida que se realizaban iteraciones. Por estas razones es posible señalar que los objetivos planteados al inicio del proyecto fueron cumplidos.

12. REFERENCIAS

- [1] Juan Pablo Palacios, “El Modelo Scrum”, 2006,
http://www.navegapolis.net/files/s/NST-010_01.pdf
- [2] Kent Beck, Mike Beedle y otros, “Manifiesto for Agile Software Development”, 2001, <http://agilemanifesto.org/>
- [3] Capability Maturity Model Integration. SEI – Software Engineering Institute ,2010
- [4] Carnegie Mellon. SEI- Software Engineering Institut, 2010 .
- [5] Ken Schwaber, “Guía sobre Scrum”, Mayo 2009.
- [6] Juan Palacio, Claudia Ruata, “Scrum Manager”, Enero 2011,
<http://www.safecreative.org/work/1012268137397>
- [7] Juan Manuel Cueva Lovelle, “Calidad del Software”, Octubre 1999.
- [8] Mary Beth Chrissis, Mike Konrad, Sandy Shrum, “Guía para la integración de procesos y la mejora de productos”, 2009.
- [9] Yves Wautelet, Manuel Kolp, Youssef Achbany, “S-Tropos: An Iterative SPEM-Centric Software Project Management Process”,
- [10] Francisco Ruiz, “Software Process Engineering”, Noviembre 2007.
- [11] Francisco Ruiz, “Guía de Uso de SPEM 2 con EPF Composer v3.0” , Abril 2008.
- [12] Object Management Group, “Software & Systems Process Engineering Meta-Model Specification v2.0”. Abril 2008
- [13] Craig Larman, “UML y Patrones” 2Ed, Prentice hall, 2003
- [14] Banco Estado, Sitio web Corporativo.
<http://www.corporativo.bancoestado.cl/Acerca-BancoEstado/Gobierno-Corporativo/BancoEstado-y-filiales/Microempresas-Asesorias-Financieras.aspx>
- [15] Pressman, R., Software Engineering: A Practitioner's Approach, McGraw-Hill, 4th edition, 1997.
- [16] Humphrey, W., A Discipline for Software Engineering, Addison-Wesley, 1995.
- [17] [5] Eickelman, N. and Hayes, J. eds., “New Year’s Resolutions for Software Quality”, IEEE Software, Jan-Feb 2004, pp. 12-13.

[18] IEEE Standard Glossary of Software Engineering Terminology, IEEE Std 610.12-1990, 1990.

[19] Tom DeMarco, Controlling Software Projects: Management, Measurement, and Estimates. Prentice Hall, 1986

[20] R. Edward Freeman, Strategic Management Paperback: A Stakeholder Approach, Cambridge University Pr, 2010