

DEFINICIÓN, USO Y
CERTIFICACIÓN DE UNA
METODOLOGÍA DE DESARROLLO

por

Martín Balzamo (martin.balzamo@gmail.com)

Tesis propuesta como cumplimiento parcial de
los requisitos para la Maestría en

Ingeniería de Software

Universidad Nacional de La Plata

2009

Director

Alejandro Fernández

La fábula del Cazador de Dragones

... Se cuenta de un joven chino que dedicó toda su vida a aprender el arte de cazar dragones,

hasta que estuvo seguro de que ya dominaba todas las técnicas de como cazar dragones.

En ese momento se dio cuenta de que no habían en el mundo dragones que pudieran ser cazados

y el joven se dedicó a enseñar como cazar dragones...

As computer scientists we have developed programming languages, have had some success in formally describing them and their effects, have encoded enormously complex descriptions of information manipulation processes, and have had success in instantiating such processes to carry out such impressive jobs as guiding the landing of people on the moon, switching myriads of telephone calls in a second and navigating airplanes and rockets at high speeds. In doing all this, we as a community have developed considerable instincts, talents and tools for treating processes as tangibles. It seems most natural and most promising to harness these instincts, talents and tools to the job of describing software development and evolution processes just as we approach classical "programming".

Como científicos informáticos hemos desarrollado lenguajes de programación, los hemos descrito formalmente, a ellos y a su comportamiento, hemos codificado descripciones enormemente complejas de procesos de manipulación de información como por ejemplo para guiar la llegada del hombre a la luna, o conectar miles de llamadas de teléfono en un segundo o guiar aviones o cohetes a alta velocidad. Al hacer todo eso, nosotros como comunidad, hemos desarrollado instintos considerables, talentos y herramientas para tratar procesos como elementos tangibles. Parece muy natural y muy prometedor aprovechar estos instintos, talentos y herramientas al describir procesos de desarrollo y evolución de software como cuando hacemos “programación” clásica.

El párrafo anterior, incluido en su idioma original y traducido por el autor, fue tomado del paper SOFTWARE PROCESSES ARE SOFTWARE TOO [OSTERWEIL, 1987] que fue inspirador de esta tesis y de todo el trabajo realizado durante 2007 y 2008 hacia la certificación CMMI nivel 2 en un centro de desarrollo de software de 120 empleados en Buenos Aires, Argentina [CMMI, RESULTADOS].

OBJETIVO

DEFINICIÓN, USO Y CERTIFICACIÓN DE UNA METODOLOGÍA DE DESARROLLO

Esta tesis tiene como objetivo definir una metodología y tecnologías asociadas para desarrollar sistemas en un Centro de desarrollo de Software. La metodología fue implementada con éxito, durante el 2008, en un centro de desarrollo de reconocido prestigio, que consiguió a fin del año 2008 certificar CMMI nivel 2 [CMMI-SW, 2002]. Esto constituye una confirmación de la aplicabilidad y validez de la metodología.

El proceso de definición de la metodología será llevado a cabo utilizando los mismos principios y herramientas que se utilizan para desarrollar un sistema. Debido a que esta metodología será concebida desde su inicio para soportar una certificación, su adopción en otro centro disminuiría considerablemente los costos y tiempos que implican certificar una norma de calidad. Finalmente, al usar los mismos principios y herramientas que se utilizan para desarrollar un sistema, es de esperar que la capacitación a desarrolladores del Centro de Desarrollo sea más fácil.

TABLA DE CONTENIDO

Tabla de contenido	i
Lista de figuras.....	iv
Agradecimientos	v
Glosario	vi
Capítulo 1	1
Introducción.....	1
¿De qué trata esta tesis?.....	1
A quiénes está dirigido el texto	2
Organización del documento.....	3
Capítulo 2.....	5
Presentación del problema.....	5
Historia de CMM	5
Enfoque y estrategia.....	10
Relevancia y originalidad de este trabajo.....	11
Capítulo 3.....	13
Herramientas existentes	13
Herramientas para realizar evaluaciones CMMI:.....	14
Herramientas para gestionar proyectos:.....	14
Herramientas para gestionar pedidos o incidentes:	15
Conclusiones	16
Capítulo 4.....	17
MET.....	17
Capítulo 5.....	19
MET modelo de requerimientos	19
MET Modelo de Requerimientos	20
Objetivo	20
MET Modelo de Requerimientos	21
Modelo de Dominio	21
MET Modelo de Requerimientos	22
RFMET001 Administrar especificaciones de requerimientos (SRS).....	22
Capítulo 6.....	23
MET modelo de casos de uso.....	23
MET Modelo de Casos de uso	25
Breve descripción funcional.....	25
MET Modelo de Casos de uso	26
Orden común de ejecución de Casos de Uso.....	26
MET Modelo de Casos de uso	27
Descripción de Casos de Uso	27
CU01. Iniciando la Gestión de Una Petición.....	27
Capítulo 7	29
EL SRA (Sistema de Registro de auditorías)	29
Casos de uso de SRA.....	30

Pantallas de SRA.....	33
Capítulo 8.....	37
La experiencia de certificar cmmi nivel 2 y la implantación de met	37
PMAM 1 (7 de Septiembre de 2007)	38
Implantación de MET.....	39
PMAM 2 (8 de Agosto de 2008)	40
SCAMPI (19 de Diciembre de 2008)	42
Capítulo 9.....	44
Trabajo futuro y Conclusiones.....	44
Consolidar y continuar el desarrollo del SRA	44
Evolucionar MET para CMMI nivel 3.....	44
Programar MET para CMMI nivel 2	45
Conclusiones	46
Anexo I - MET Modelo de Requerimientos.....	48
Objetivo	48
Glosario.....	48
Relación con otros productos	49
Descripción general del producto	51
Perfiles de Actores	52
Modelo de Dominio	53
Supuestos y restricciones	53
Requerimientos del Producto	54
RFMET001 Administrar especificaciones de requerimientos (SRS).....	54
RFMET002 Realizar el Control de los requerimientos.....	54
RFMET003 Realizar Estimaciones de proyectos	54
RFMET004. Realizar el Control de Estimaciones.....	54
RFMET005 Administrar la planificación de proyectos.	54
RFMET006 Realizar el Control de la planificación de proyectos.....	54
RFMET007 Realizar seguimientos de proyectos.	55
RFMET008 Realizar el control sobre seguimientos.....	55
RFMET009 Realizar la planificación de una línea base.....	55
RFMET010 Realizar el control de la línea base	55
RFMET011 Realizar Métricas	56
RFMET012 Realizar Publicación de Métricas.....	56
RFMET013 Realizar el control sobre Publicación de Métricas	56
Anexo II - MET modelo de casos de uso	57
Breve descripción funcional	57
Diagrama de Casos de Uso.....	58
Orden común de ejecución de Casos de Uso.....	58
Referencias.....	59
Descripción de Casos de Uso	59
CU01. Iniciando la Gestión de Una Petición.....	59
CU02. Generando Repositorio para la Petición / Proyecto.	60
CU03. Auditando Creación del Repositorio.	61
CU04. Estimando una Petición.	62
CU05. Documentando Requerimientos del Proyecto.....	63

CU06. Auditando Estimación.....	64
CU07. Auditando Requerimientos del Proyecto.....	66
CU08. Planificando un Proyecto.....	68
CU09. Auditando Planificación.....	69
CU10. Gestionando No Conformidad.....	72
CU11. Realizando Seguimiento a un Proyecto.....	73
CU12. Auditando Seguimiento de un Proyecto.....	75
CU13. Auditando Auditorías.....	76
CU14. Liberando entregables.....	78
CU15. Auditando Liberaciones.....	79
CU16. Realizando Cambio de Alcance.....	80
CU17. Auditando Cambio de Alcance.....	81
CU18. Cerrando Proyecto.....	83
CU19. Auditando Cierre de Proyecto.....	83
CU20. Generando Métricas.....	85
CU21. Publicando Métricas.....	86
CU22. Auditando publicación de Métricas.....	87
Bibliografía	88
Índice.....	3

LISTA DE FIGURAS

<i>Descripción</i>	<i>Página</i>
Figura 1 CMM – Objetivos, costos y tiempos	7
Tabla 2 CMMI – PA's nivel 2.....	9
Figura 3 MET Diagrama de dominio.	21
Figura 4 MET Orden común de ejecución de Casos de Uso.	26
Figura 5 SRA Lista de proyectos.....	33
Figura 6 Lista de auditorías, no conformidades y niveles de escala	34
Figura 7 SRA Lista de no conformidades abiertas	36
Figura 8 Resultado PMAM 7 de Septiembre de 2007	38
Figura 9 Significado colores PMAM	38
Figura 10 Relación de MET con otros productos.....	49
Figura 11 MET Vista Sumaria.....	51
Figura 12 MET Diagrama de dominio.	53
Figura 13 MET Diagrama de casos de uso.....	58
Figura 14 MET Orden común de ejecución de Casos de Uso.	58

AGRADECIMIENTOS

El autor desea expresar su más sincero agradecimiento a familiares, profesores y compañeros de trabajo.

En especial a:

- A René por ponerse la certificación al hombro.
- A Cecilia por leer esta tesis con mucho esmero.
- A César y Germán por hacer el SRA..
- A Casco por aguantarme.

GLOSARIO

Modelo. Un modelo es una vista de un sistema de información [ARLOW, 2005]. Puede verse un sistema de información desde los casos de uso que resuelve o los requerimientos que satisface. El hecho de hablar de modelos y no de documentación lleva implícitamente a que cualquier cambio requiere actualizar todas las vistas o modelos.

Metodología. Conjunto de actividades y resultados asociados que producen un producto de software [SOMM, 2002]. Estas actividades son llevadas a cabo por los denominados ingenieros de software.

Proceso de desarrollo de software: Según la IEEE [IEEE], aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software; es decir, la aplicación de ingeniería del software.

Certificación. Práctica de mercado que garantiza que una organización trabaja con cierto nivel de calidad.

QC: *quality control* o Control de Calidad es el acto de dirigir, influenciar, verificar y corregir para asegurar la adherencia de cierto producto a un diseño o a una especificación [BAS, 2002]. La tarea más conocida de QC es *testing* o prueba del software.

QA: *quality assurance* o Aseguramiento de Calidad es el acto de dirigir, enseñar y auditar el proceso por el cual se desarrollan productos para proveer confianza respecto a la posible adherencia de un producto a una especificación de diseño [BAS, 2002]. La tarea más conocida de QA es la auditoría. La presunción en este caso es que un buen proceso controlado produce buenos productos.

CMMI: Capability Maturity Model Integration. Modelo para la mejora o evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software. Fue desarrollado por el Instituto de Ingeniería del Software [SEI, 2009] de la Universidad Carnegie Mellon [CMU], y publicado en su primera versión en enero de 2002 [CMMI-SW, 2002].

SCAMPI: El método SCAMPI (Standard CMMI Appraisal Method for Process Improvement) es un proceso diseñado y desarrollado por el Carnegie Mellon-SEI para ofrecer evaluaciones (benchmarks) de calidad con relación a los modelos Capability Maturity Model Integration (CMMI) [SCAMPI, 2005].

SCAMPI Lead Appraiser: Un "SCAMPI Lead Appraiser" (Líder-evaluador SCAMPI) es un profesional que ha completado con éxito el proceso de autorización-acreditación de la Carnegie Mellon University-SEI que incluye un amplio curriculum formativo y requisitos de experiencia profesional, así como superar con éxito una observación "in-situ" ejerciendo como líder de una evaluación (appraisal) SCAMPI Clase A en una organización-cliente [SCAMPI, 2005].

PMO Oficina de gestión de proyectos : Es un departamento o grupo que define y mantiene estándares de procesos, generalmente relacionados a la gestión de proyectos, dentro de una organización[PMI].

Software factory o fábrica de desarrollo: Una organización que desarrolla software exclusivamente. En Argentina, después de la crisis de 2001 [ARG, CRISIS], el desarrollo de software tuvo un crecimiento sostenido de aproximadamente un 20% anual [NOTICIAS, SOFTWARE]. La forma de empresa más expandida de la industria informática, en Argentina, es la fábrica de desarrollo o centro donde se desarrolla software. Típicamente, una organización con analistas, diseñadores, arquitectos, desarrolladores y testers. Algunas veces con una oficina de proyecto y un departamento de aseguramiento de la calidad.

SEI: Es un centro de investigación y desarrollo, subvencionado por el estado, que hace investigación en ingeniería de software [SEI, 2009].

Capítulo 1

INTRODUCCIÓN

¿De qué trata esta tesis?

Certificar CMMI nivel 2 de acuerdo al modelo CMMISW [CMMI-SW, 2002] es una actividad normada y precisa, llamada SCAMPI [SCAMPI, 2005], en la que un equipo evaluador entrevista a personas de la organización que desea certificar y, principalmente, revisa evidencia sobre la forma de trabajar de dicha organización. Esto es, controla la evidencia que muestra que la organización *realmente* aplica el proceso que dice aplicar y que cuando no lo hace, registra tales fallas como *no conformidades* y pone en funcionamiento los mecanismos que sean necesarios para corregirlas.

En general, en una organización con un área de Aseguramiento de la Calidad, existen analistas que controlan que se cumpla la metodología. Estas personas chequean la aplicación de los procesos y generan no conformidades. Comúnmente, la persona de Aseguramiento no deja registro de que auditó un proceso y si lo deja es en forma de minuta, si es que la auditoría se hizo en una reunión, o de un documento Word tipo checklist. Las no conformidades son generalmente informadas via e-mail que terminan almacenados sólo en las carpetas de e-mail de los destinatarios. En consecuencia, la evidencia suele estar distribuida en toda la empresa con diferentes formatos y en distintos soportes (mail servers, file servers, versionadores).

El proceso de certificación SCAMPI involucra a un “certificador” (Lead Appraiser) quien, luego de estudiar los procesos que la organización dice estar utilizando, comprueba que eso es así. Para ello recolecta información entrevistando a personas que trabajan en el lugar y pide evidencia (documentos, mails, registros) [CMMI, APPR]. Estos tres elementos, metodología escrita, evidencia y entrevistas, son los elementos que *cruza* el certificador. En primer término busca coherencia entre ellos y después los valida con respecto a lo que el modelo CMMI prescribe. Por ejemplo, si la metodología escrita dijera que el usuario envía un email al analista con sus requerimientos, mientras que en una entrevista un analista dijera que él habla por teléfono con el usuario y, finalmente, la

evidencia fuera un documento con los requerimientos, el certificador diría que no hay una política de requerimientos instaurada.

La propuesta de esta tesis es definir una metodología que pueda validarse contra lo prescripto por el Modelo CMMI. La forma en la que especifique la metodología debe minimizar el esfuerzo de implementación de la misma, de auditoría y certificación. A fin de minimizar el esfuerzo de implementación de la metodología se utilizarán en su desarrollo *las mismas técnicas que se utilizan para construir un sistema* [OSTERWEIL, 1987]. Finalmente, se propone aplicar la metodología durante seis meses, adecuarla, evolucionarla y certificar CMMI nivel 2.

El modelo CMMi se divide en veintidós áreas de proceso [CMMI, 1.2]. En particular, para el área de proceso de aseguramiento de la calidad se va a construir un sistema de información que deje registro detallado de las distintas actividades. Finalmente, teniendo el registro, chequear que un proyecto tenga todas las auditorías correspondientes es inmediato. Registrar una no conformidad, escalarla si no se resuelve en función del tiempo y registrar su cierre o cancelación son funcionalidades que proveerá dicho sistema.

A quiénes está dirigido el texto

El trabajo se encuentra dirigido principalmente a ingenieros de software, profesionales de la industria del software, y cátedras universitarias vinculadas al software.

La documentación aquí contenida puede tomarse como referencia para la adopción de prácticas de ingeniería del software acordes al nivel 2 del modelo CMMI-SW. Una organización que gestione proyectos de desarrollo de sistemas puede tomar los dos artefactos, Modelo de requerimientos y Modelos de casos de uso, y adecuar su organización al uso descrito en el segundo. Esto no es un trabajo menor, exige capacitar, entrenar, liderar, controlar. Si logra institucionalizar la metodología (que todos la usen) siguiendo esos casos de uso tiene resuelto dos de los tres pilares de la certificación: *metodología escrita*, ya que MET es una serie de documentos, y *las entrevistas*, asumiendo que la *implantación* de MET fue efectiva. El tercer pilar, el faltante, es la *evidencia de uso*.

El software generado para el área de aseguramiento de la calidad, cuya especificación y principales pantallas se incluyen en este trabajo, puede utilizarse fácilmente en cualquier organización que haga proyectos informáticos. El software aporta evidencia: en particular para el área de procesos más importante de la certificación: PPQA. Además este software contribuye directamente con la meta genérica de nivel 2 y con las prácticas genéricas de nivel 2:

- Meta genérica de nivel 2, GG 2: “Institucionalizar un proceso gestionado”.
- Práctica genérica de nivel 2, GP 2.8: “Monitorizar y controlar el proceso”.
- Práctica genérica de nivel 2, GP 2.9: “Evaluar objetivamente su cumplimiento”.
- Práctica genérica de nivel 2, GP 2.10: “Revisión del estado con los superiores”.

La experiencia de modelar la metodología como software es inspiradora. Quienes desarrollan la metodología sienten la contención que genera hacerlo en un contexto conocido (para hacer la metodología se utilizarán: diagramas UML [FOWLER, UML], casos de uso [COCKBURN, 2000], especificaciones de requerimientos [SOMMER, 1998]). El uso concreto de la metodología en proyectos y la certificación validan todo lo propuesto.

Organización del documento

El material se divide en nueve capítulos que abarcan la totalidad del trabajo de tesis.

- El capítulo 1. Introducción (el presente) sintetiza los objetivos de la tesis, a quiénes está dirigida, y de qué manera se encuentra organizado el material de la misma.
- El capítulo 2. Presenta el problema mostrando el contexto que da origen al trabajo de tesis. Muestra la relevancia y originalidad de la misma
- El capítulo 3. Estado del arte. Estudio de herramientas utilizadas para gestionar proyectos de software y para una certificación.

- El capítulo 4. Presentación de MET (METodología de gestión de proyectos), la estrategia utilizada para su construcción y las ventajas de hacerlos de esa forma.
- El capítulo 5. El modelo de requerimientos de MET. Incluye un modelo de dominio, su relación con otros sistemas o diagrama de contexto y la lista de requerimientos.
- El capítulo 6. El modelo de casos de uso de MET. Incluye los perfiles de usuarios de MET y la lista de los casos de uso de MET
- El capítulo 7. Presenta el sistema de registro de auditorías (SRA). Este sistema administra toda la evidencia al hacer aseguramiento de la calidad.
- El capítulo 8. La experiencia de certificar CMMI nivel 2. Documenta las adecuaciones a la metodología que tuvieron lugar durante la experiencia de uso.
- El capítulo 9. Trabajo futuro y posibles extensiones del presente trabajo. Incluye también las conclusiones y la comparación de MET y el SRA con el estado del arte.

Capítulo 2

PRESENTACIÓN DEL PROBLEMA

En este capítulo se presentan los conceptos principales del problema a resolver por el presente trabajo de tesis. El mismo comienza con un breve repaso histórico y una presentación de los conceptos principales. A continuación se analiza brevemente la situación actual. Finalmente identifica el problema a resolver y presenta la solución propuesta en el trabajo indicando su relevancia y originalidad.

Historia de CMM

A principios de la década del 90, una firma dedicada al estudio del mercado de Tecnologías de Información publica un reporte [THE STANDISH GROUP, 1995] sobre el éxito de los proyectos de desarrollo en la industria del software. El reporte, basado en encuestas hechas sobre proyectos de software, informaba los siguientes resultados estadísticos:

- El 30% de los proyectos se cancelaban
- El 54% de los proyectos excedían ampliamente los tiempos y costos estimados
- El 16% de los proyectos finalizaban exitosamente dentro del tiempo, el costo y la funcionalidad prevista

El Departamento de Defensa de los EEUU, el mayor contratista de software del mundo, para tratar de revertir esta situación, funda el SEI (Software Engineering Institute) [SEI, 2009] en la universidad Carnegie Mellon, con el propósito de estudiar el problema y encontrar alguna solución.

En 1991, el SEI publica el modelo CMM (Capability Maturity Model) [CMM, 1991]. El modelo está orientado a la mejora de los procesos relacionados con el desarrollo de software, para lo cual contempla las consideradas mejores prácticas de ingeniería de software y de management.

A partir de ese momento, el Departamento de Defensa exige que sus proveedores estén certificados en CMM, lo que impulsa a que el modelo tenga una amplia aceptación y se convierta en un estándar de facto dentro de la industria del software.

El modelo CMM original define cinco niveles de madurez dentro de los cuales se puede encontrar una organización [CMM, 1991]:

- Nivel 1 – Inicial: el proceso de software es impredecible, sin control y reactivo. El éxito de los proyectos depende del talento de los individuos.
- Nivel 2 – Repetible: existen procesos básicos de gestión de proyectos (costo, calendario, funcionalidad). Los procesos existentes hacen que se puedan repetir éxitos en proyectos de similares características.
- Nivel 3 – Definido: existe un proceso de software documentado y estandarizado dentro de la organización. Todos los proyectos utilizan una versión a medida del proceso.
- Nivel 4 – Manejado: la organización recolecta métricas del proceso software y de los productos desarrollados. Tanto el proceso como los productos se entienden y controlan cuantitativamente.
- Nivel 5 – Optimizante: existe una mejora continua del proceso software, basada en la realimentación cuantitativa del proceso y en la puesta en práctica de ideas y tecnologías innovadoras.

De acuerdo al modelo, la performance en general de una organización mejora notablemente a medida que la misma incrementa su nivel de madurez. La figura 1 muestra de manera conceptual las mejoras en la performance para cada nivel contemplando los factores Tiempo y Costo [PAULK, CMM1995]. Conclusiones similares se pueden extraer para otros factores como la Funcionalidad y la Calidad.

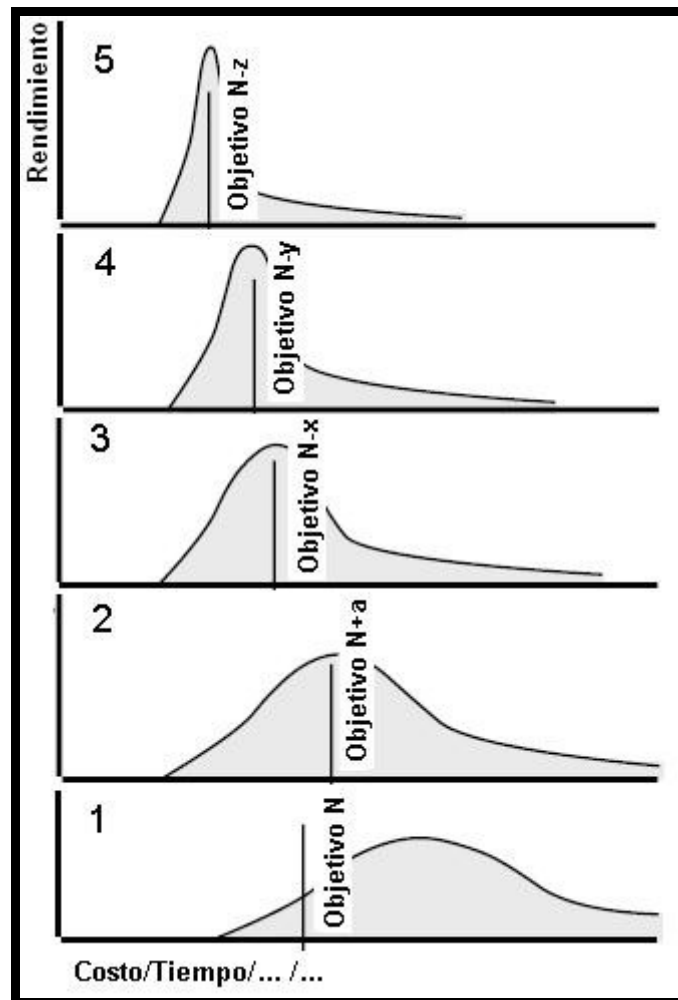


Figura 1 CMM – Objetivos, costos y tiempos

Variación del rendimiento con respecto a los objetivos fijados para los factores costo y tiempo, de acuerdo al nivel de CMM de la organización [Paulk et al, 1993].

En las organizaciones que se encuentran en el nivel 1, los objetivos generalmente son ampliamente excedidos por la realidad

En las organizaciones que se encuentran en el nivel 2, se establecen objetivos más acordes a la realidad.

En las organizaciones que se encuentran los niveles 3, 4, y 5, existe una menor dispersión de la realidad con respecto a los objetivos, y la performance mejora con cada nivel.

Algunas de las organizaciones que adoptaron el modelo fueron [SEIR, 2003]: Accenture, AT&T, Boeing, Ericsson, Fuji Xerox, Hewlett Packard, Hyundai, IBM, Motorola, Nasa, NCR, NEC, PriceWaterhouseCoopers, Samsung, Siemens y United Airlines.

Luego del éxito alcanzado por CMM, el SEI desarrolló modelos similares para otras disciplinas, entre las cuales figuraban la ingeniería de sistemas, la adquisición de software, las personas, y el desarrollo integrado de productos [CMMS, 2003]

A mediados de la década del 90, el SEI decide unificar todos los modelos, embarcándose en un esfuerzo que culmina en el año 2002 dando origen a una nueva generación llamada CMMI (Capability Maturity Model Integration) [CMMI, 2002].

El nuevo modelo CMMI brinda un marco con una estructura común para todas las disciplinas (ingeniería de software, ingeniería de sistemas, desarrollo integrado de productos, adquisición de productos, personas) y agrega una nueva forma de representación además de la conocida representación por niveles. La nueva forma de representación se llama Continua y está orientada a medir la mejora en los procesos de manera individual en vez de hacerlo de manera conjunta como la representación por niveles [CMMI, 2002].

Dentro de esta nueva generación de modelos, el sucesor directo del CMM original es el denominado CMMI-SW [CMMI-SW, 2002]. Este modelo presenta una mayor cobertura con respecto a las áreas de proceso, y agrega el concepto de representación continua.

En paralelo con el desarrollo de CMMI, el SEI elaboró un método para la evaluación formal del modelo denominado SCAMPI (Standard CMMI Appraisal Method for Process Improvement) [SCAMPI, 2001]. El método define una serie de reglas para la evaluación del modelo, las cuales deben utilizarse para valorar las distintas partes del mismo durante una evaluación formal.

La definición del SEI del SCAMPI es detallada y difícil de leer, sin embargo se puede resumir el proceso en las siguientes etapas [PHILLIPS, 2004]:

- Capacitación previa: Lo prescripto por CMMI es interpretado por el lead appraiser y un equipo de evaluación formado por cuatro, cinco o seis integrantes. Para ser parte del equipo de evaluación se debe asistir a un curso específico, habilitante y requerido por el SEI, para ser team member en una evaluación.
- La planificación: El Lead Appraiser pide la lista de los proyectos y selecciona tres en los que va a focalizarse. A partir de esta selección arma una agenda con los distintos niveles de la organización.
- Las entrevistas y revisión de la evidencia: El Lead Appraiser modera reuniones en las que pregunta, para cada área de proceso de nivel dos, las prácticas específicas esperadas en ese nivel. Más adelante se muestran las áreas de proceso de nivel dos.
- La devolución final de hallazgos: El Lead Appraiser presenta los hallazgos a la organización. Si la organización cumple con todas las prácticas habrá certificado CMMI.

En [SCAMPI] puede verse la definición de SCAMPI del SEI.

Las áreas de procesos de nivel dos son:

Administración de Requerimientos (REQM)
Planificación de Proyectos (PP)
Control y Monitoreo de Proyectos (PMC)
Mediciones y Análisis (MA)
Aseguramiento de Calidad de Procesos y Productos (PPQA)
Administración de la Configuración (CM)
Administración de Acuerdos con Proveedores (SAM)

Tabla 2 CMMI – PA's nivel 2

En síntesis, el modelo CMMI y, en particular, la certificación implica costos altos y plazos en el orden de semestres o años, necesarios para la preparación previa a la adopción, para

la adopción propiamente dicha y, de llevarse a cabo, para la certificación. El problema se ve más acentuado en las organizaciones pequeñas, donde los recursos económicos, humanos y temporales suelen ser menores que en las grandes organizaciones.

Enfoque y estrategia

Los entregables relevantes de esta tesis son:

- El modelo de requerimientos de MET, la *METodología* del Centro de Desarrollo. Este documento describe que es lo que MET resuelve.
- El modelo de casos de uso de MET. Este documento describe en que momentos de un proyecto hay que hacer algo, quien lo hace y donde.
- La especificación del sistema SRA, sistema para registro de auditorías, y sus principales pantallas. El SRA es el sistema que utiliza el área de Aseguramiento de la calidad para registrar y organizar su trabajo. Este sistema es utilizable en cualquier fábrica de software que intente certificar nivel dos, ya que el proceso que controla y las auditorías que permite hacer son configurables.
- La experiencia adquirida al certificar, ya que en ese capítulo se expresa que le faltó a MET.

El enfoque, para desarrollar MET, es el mismo que el utilizado para construir cualquier producto de software en el centro de desarrollo: conseguir los requerimientos, conseguir los casos de uso. Normalmente a esto le sigue diseñar y construir software. *En este caso mucha parte del software ya está construido:*

- Administrador de pedidos por parte del usuario. (IT Governance)
- Administrador de proyectos. (IT Governance)
- Registro de horas. (IT Governance)

- Repositorio de fuentes y documentos. (SVN)
- Email server. (Microsoft Exchange)
- Administrador de requerimientos (TRAC)

Relevancia y originalidad de este trabajo

La certificación es un trabajo que se desarrolla en tres ejes:

- Evaluación de los procedimientos o procesos o metodología.
- Entrevista con empleados.
- Revisión de evidencia.

El objetivo de este trabajo es construir una metodología de desarrollo de software para una fábrica de sistemas (en inglés *software factory*). La metodología será plasmada en dos artefactos centrales: el modelo de requerimientos de la metodología, y el modelo de casos de uso de la metodología. Estos dos artefactos deberán utilizarse como descripción de la metodología de la organización que intenta certificar. La ventaja de usar este enfoque (eso es, especificar la metodología de la misma manera que se especifica un sistema) a la hora de escribir una metodología es la aceptación entre los integrantes de la software factory, acostumbrados a tratar con requerimientos y casos de uso en su tarea diaria. Es necesario que durante el proceso de evaluación todos los entrevistados tengan muy en claro la metodología de trabajo (institucionalización de la metodología). Capacitar a todos los empleados de una *software factory* es un trabajo complejo. Este enfoque reduce esa complejidad.

Una certificación es un proyecto complejo y caro. Esta tesis es relevante porque permitiría acortar los plazos sensiblemente. Cualquier organización podría utilizar la metodología aquí propuesta y el software aquí presentado y resolver las áreas de procesos de CMMI de:

- Planificación
- Estimación
- Seguimiento
- Aseguramiento de la calidad
- Administración de las configuraciones

(Nota: En algún párrafo anterior teníamos siete áreas para nivel dos. La de métricas es propia de lo que quiere medir cada organización y la de administración de proveedores puede descartarse en aquellas organizaciones que no le contratan software a proveedores)

Las contribuciones de esta tesis son las siguientes:

- Una metodología de gestión de proyectos muy orientada a una certificación CMMI nivel 2.
- Una herramienta de registro de proyectos, seguimientos, auditorías y no conformidades.
- Una evaluación de la aplicabilidad de la metodología y la herramienta en la certificación del Centro de Desarrollo.

Esta tesis es original por el enfoque en la construcción de la metodología como un sistema de información.

Entender que la evidencia para una certificación puede obtenerse a partir de la automatización de todos los controles es también una idea original. La idea de construir un motor que chequee distintos registros y escale los problemas según un organigrama es útil para aplicar a cualquier problema de administración de tareas.

El proceso de certificación suele hacerse en tres etapas: pre evaluación, adecuación según las observaciones de la pre evaluación y certificación. El producto MET fue construido para solucionar los problemas detectados en una pre evaluación del Centro de Desarrollo de una empresa de telecomunicaciones de Latinoamérica.

Capítulo 3

HERRAMIENTAS EXISTENTES

Al momento de realizarse esta tesis y después de buscar en distintos sitios, publicaciones y habiendo consultado a una decena de expertos, **no encuentro evidencia de una metodología escrita con casos de uso.**

Existen metodologías más abstractas que MET (UP [SCOTT, 2002], Scrum [SCHWABER, SCRUM] , etc.) y existen, al menos en Argentina, a partir de una economía favorable para empresas extranjeras a partir de 2002, muchos centros de desarrollo donde cada uno tiene su metodología escrita. Esta aseveración no es producto de un trabajo de campo, sin embargo en el sitio del SEI se puede encontrar la cantidad de certificaciones por nivel y por país y, en 2008, Argentina fue el país con mayor crecimiento en número de certificaciones [CMMI, SAS]. Esto se explica por dos razones, los costos bajos en Argentina y la subvención que el gobierno otorga a aquellas organizaciones que desarrollan software, requiriendo para esto una certificación de calidad reconocida mundialmente. Muchas de las organizaciones que quieren acceder a este beneficio certifican CMMI nivel 2.

Durante la certificación en el Centro de Desarrollo, MET y sus ideas fueron validadas con dos evaluadores (*lead appraiser*) y con, al menos, cuatro team members que trabajan en diferentes consultoras en mejora de procesos. Todos coincidieron en no haber visto una metodología escrita como MET.

En esta sección se investigan productos no comparables con MET, pero que pueden tener algún punto de contacto con CMMI, con MET o con el SRA. Existen en el mercado herramientas para hacer una evaluación CMMI, existen herramientas para gestionar proyectos, existen herramientas para administrar incidentes. Esta sección muestra un breve resumen de diversas herramientas. Algunas de ellas podrían utilizarse para registrar proyectos, auditorías y no conformidades, pero son herramientas y no una metodología.

MET es una *especificación de requerimientos* y una *lista de casos de uso*. Esta tesis trata sobre la experiencia de hacer MET, muestra MET en su totalidad como anexos ya que es utilizable en cualquier centro de desarrollo, especifica el SRA que es el sistema que a la hora de hacer MET surgió como necesidad de informatización y finalmente se analiza que más hubo que construir para completar la certificación.

Herramientas para realizar evaluaciones CMMI:

A continuación se detallan herramientas que sirven para llevar adelante una evaluación. El hecho de incluir herramientas que soportan una evaluación es sólo para ver ideas de construir software para problemas que tenemos los ingenieros de software y en particular cuando queremos certificar CMMI.

Es el tipo de herramienta que usaría un evaluador.

- CMM-Quest: permite efectuar evaluaciones de acuerdo al modelo CMMI-SE/SW en su representación continua. La evaluación se limita a asignar valores a los objetivos.
- IME Toolkit: permite efectuar evaluaciones de acuerdo al modelo CMMI-SE/SW. Las evaluaciones consisten en asignar valores numéricos a las prácticas, en base a los cuales la herramienta genera puntajes para las áreas de proceso. No brinda soporte para el método SCAMPI. No posee guías de asistencia para la evaluación [IME Toolkit, 2003]
- Appraisal Wizard: soporta evaluaciones para gran parte de los modelos CMM y métodos de evaluación propuestos por el SEI a lo largo de la historia (entre ellos, todos los CMMI y SCAMPI). Está pensada para cubrir todas las necesidades del método SCAMPI, requiriendo amplios conocimientos del mismo por parte del usuario. Requiere que el usuario ingrese todos los valores que se asignan en las distintas instancias de evaluación (prácticas, objetivos, áreas de proceso) y no cuenta con la capacidad de sugerir valores facilitando las tareas de ingreso de datos. Al brindar un soporte tan amplio y detallado, la herramienta es compleja de utilizar [Appraisal Wizard, 2003].

Herramientas para gestionar proyectos:

A continuación se detallan algunas herramientas que sirven para gestionar proyectos. Es el tipo de herramienta que usa un líder de proyecto. En cualquiera de estas se puede modelar una auditoría como una tarea o como un hito de proyecto. Se puede tener diferentes nombres de hitos para las diferentes auditorías. Modelar las no conformidades es más difícil. Las no conformidades son más parecidas a incidentes y es el tercer tipo de

herramientas que vamos a ver. (Recordar que el SRA es la herramienta usada en esta experiencia para modelar auditorías y no conformidades)

- Microsoft Project: Permite gestionar proyectos, esto es administrar tareas en el tiempo, su esfuerzo, fechas, recursos, dependencia entre tareas. Se puede modelar cualquier tipo de tarea. Desde el punto de vista de generar y registrar evidencia al haber hecho una auditoría, una *tarea de auditoría en un proyecto* hecha al cien por ciento significaría que la tarea se hizo. Las no conformidades y la vista de los proyectos desde el punto de vista de QA, esto es ver que proyectos tuvieron auditorías, cuales tienen no conformidades abiertas, cuales tienen no conformidades resueltas es incómodo o imposible.
- ITG: Permite gestionar proyectos, registrar horas, registrar incidentes, llevar una carta Gantt, hacer líneas bases o fotos de un proyecto a cierto momento. Es una herramienta que utiliza el líder de proyecto, la oficina de proyectos y/o el área de soporte porque modela incidentes. En ITG se pueden definir templates de proyectos y estos podrían tener las auditorías planeadas. La evidencia de la auditoría realizada podría ser la existencia de horas asignadas al proyecto desde QA. Esto requiere que el líder de proyecto habilite al auditor de QA a cargar horas y es incómodo. Las no conformidades pueden modelarse con incidencias. El hecho de usar incidencias, que normalmente se utilizan para modelar una falla en producción y un pedido a soporte requiere configuración. Además los incidentes no se asocian a proyecto mientras que las no conformidades son de un proyecto. Con esfuerzo se puede configurar esta herramienta para que funcione como el SRA.

Herramientas para gestionar pedidos o incidentes:

A continuación se detallan algunas herramientas que sirven para gestionar incidentes o pedidos a un área de soporte o mesa de ayuda. Si uno modela un proyecto como un incidente, una tarea como un incidente, una auditoría como un incidente y una no conformidad como un incidente se puede utilizar una de estas herramientas. Su funcionalidad básica es el registro tipo bitácora.

- GFORGE: Es una herramienta administrar incidentes, tareas, foros y documentación. Es una herramienta open source. Permite planificar liberaciones de versiones y asociarle incidentes resueltos. No tiene la posibilidad de planificar auditorías.
- Remedy Action Request: Es una herramienta para administrar incidentes. Tiene un lenguaje propio de programación que permite adecuarla a cualquier necesidad. El lenguaje propietario que tiene es muy restrictivo.
- Bugzilla: Es una herramienta para administrar incidentes. Se integra con la herramienta de SCM y permite asociar incidentes a modificaciones al código. Por ser abierta se puede adecuar reprogramándola o programando algún plugin. También puede configurarse para, por ejemplo, registrar una auditoría como un incidente.

- Trac: Es una herramienta para administrar incidentes muy similar a Bugzilla. Es open source, integrable con una herramienta de versionado y también puede configurarse para, por ejemplo, registrar una auditoría como un incidente. Sin embargo, en su sitio web, Trac se define como una herramienta *minimalista* para la administración de incidentes y recomiendan, sus creadores, usarla como está sin hacerle adecuaciones.

Conclusiones

Comparamos una amplia gama de productos que utilizan los ingenieros de software para comparar MET y el SRA con sistemas informáticos existentes en la industria. MET no es comparable con otras metodologías que estén escritas como requerimientos y casos de uso, porque al momento de escribir esta tesis no encontré una metodología escrita de esa forma. El SRA es comparable con algunos productos de mercado.

Con respecto al SRA, ninguna de las herramientas revisadas es específica de QA, ni está pensada para administrar auditorías. Algunas de ellas, especialmente las de manejo de incidentes, pueden adecuarse. Sin embargo, esas herramientas no están pensadas para hacer auditorías y no conformidades. Es decir, el problema principal de las herramientas de manejo de incidentes es, como su nombre lo indica, el manejo de incidentes. Si uno modela un incidente como una no conformidad, puede ajustarse a estas herramientas.

En una organización que utilice alguna de estas herramientas, si la organización tuviera programadores con el conocimiento para extenderla o configurarla, se debería comparar el costo de tal extensión contra el costo de usar el SRA propuesto en esta tesis.

El software SRA es un software que se utiliza en el área de Aseguramiento de la Calidad (QA). En una fábrica de software (software factory) con un sistema para el manejo de proyectos y otro para el manejo de incidentes, agregar un software para QA no le agrega un nuevo sistema a los integrantes de la software factory. Sólo a los de QA, que probablemente utilicen el de gestión de proyectos para saber que proyectos están vigentes, para auditarlos, y le registren horas a los mismos cuando lo controlan.

Como decíamos en el primer párrafo, la idea de escribir los procesos como requerimientos y casos de uso, y versionar la metodología en un repositorio, trazarle líneas base, versionarla, no tiene comparación con herramientas.

Capítulo 4

MET

MET (METodología de gestión de proyectos) es un producto desarrollado según el método descrito por Craig Larman en [UML&PATTERNS, 2004]. El mismo propone escribir los casos de uso, escribir el modelo de dominio y determinar, a partir de los casos de uso, el comportamiento de los objetos del modelo de dominio.

MET está hecho utilizando los lineamientos del proceso unificado [UP, 2003]. El proceso unificado propone la utilización de *modelos* y llama *modelos* a diferentes vistas del software: el software visto desde los requerimientos que resuelve, el software visto desde los test que soporta. Esta idea tiene algunas ventajas muy importantes a la hora de desarrollar software:

- La documentación tiene la misma importancia que el software. Es tan importante generar una línea de código como un caso de uso.
- La relación entre los modelos o trazabilidad facilita la mantenibilidad del producto. Un cambio, por ejemplo sobre un requerimiento, afecta a los casos de uso relacionados al mismo y esto a los elementos del diseño relacionado.

Los requerimientos de MET se consiguieron a partir de entender el modelo CMMI. Dado que el nivel 2 de CMMI tiende a ordenar los proyectos desde el punto de vista de la planificación, la estimación, el seguimiento y las auditorías, los requerimientos de MET se refieren a esas mismas cuestiones.

Los casos de uso de MET representan la interacción entre MET y los usuarios de MET. El hecho de plantearse los casos de uso ordena y ayuda a comprender quién empieza una acción, qué registra y qué hace el sistema (el sistema como el modelo informático que uno está tratando de construir). Esto es general a cualquier desarrollo en cualquier dominio, pero en este caso es especialmente útil para entender que resuelve esta metodología.

Debido a que MET se utilizará para certificar CMMI nivel 2, como ya se explicó en el capítulo uno, es necesario capacitar a todos los integrantes de la *software factory* en el uso de MET. Una metodología escrita en requerimientos y casos de uso es más fácil de transmitir a ingenieros de software que procedimientos narrados con alguna estructura.

MET cumple entonces con los requisitos definidos en [SEDORF, 1997] para ser una metodología:

- Está documentada: Los requerimientos y casos de uso de MET documentan MET.
- Es repetible: La definición de MET es independiente del líder de proyecto o el usuario que lo pide. Es igual para toda aplicación de la misma.
- Es enseñable: Está hecha en el *idioma* de los desarrolladores.
- Está basada en técnicas probadas: Está definida según CMMI.
- Ha sido validada: Se utiliza en una organización y se utilizó en un SCAMPI.
- Es apropiada: Está definida para satisfacer los requerimientos de CMMI nivel 2.

MET MODELO DE REQUERIMIENTOS

Este capítulo presenta un resumen del Modelo de Requerimientos de MET. El mismo puede verse como el Anexo I. El formato del modelo de requerimientos es el mismo que el de cualquier producto desarrollado en el centro de desarrollo. Tiene las siguientes secciones:

- **Objetivo:** El objetivo es como máximo una carilla que explica que hace el producto.
- **Glosario:** El glosario es una lista de términos necesaria para poder leer el documento introduciendo vocabulario propio del dominio del problema.
- **Relación con otros productos:** Es un diagrama que muestra la relación del producto con otros productos. Se explica cada relación y cada producto incluido en el diagrama. Este gráfico es útil para entender los límites funcionales del producto.
- **Descripción general del producto:** Este es un diagrama donde se ven los actores que interactúan con el producto.
- **Modelo de dominio:** Este diagrama respeta el standard UML y es útil para comprender el producto por dentro.
- **Lista de requerimientos:** Esta es la lista de requerimientos. Cada requerimiento es un párrafo que describe qué resuelve el producto.

Dado que el modelo de requerimientos se incluye como anexo, en este capítulo sólo se incluye el objetivo, se explica el modelo de dominio y se da un requerimiento como ejemplo. Las secciones entre líneas son extracciones del anexo.

MET Modelo de Requerimientos

Objetivo

El objetivo de la metodología es definir los procesos para los distintos proyectos llevados a cabo en el centro de desarrollo. La metodología debe servir para:

- Definir un proyecto, es decir definir los roles que participan, los entregables que va a generar y la forma de controlarlo.
- Estimar proyectos, en esfuerzo y fechas de compromiso.
- Hacerle seguimiento a un proyecto, en lo que respecta a riesgos, avances y re estimaciones.
- Asegurar la calidad de los proyectos
- Medir los proyectos
- Definir los entregables de un proyecto, como se versionan y como se agrupan en una línea base.
- Definir la forma de controlar la aplicación de la misma.

Como se ve, en el objetivo se puede entender sobre qué aspectos, de los que podría incluir una metodología, va a tratar MET. Si uno hiciera a partir de este objetivo el conocido trabajo de subrayar sustantivos y conseguir a partir de ellos objetos para modelar un problema, conseguiría una buena parte del modelo de dominio que vamos a ver a continuación.

MET Modelo de Requerimientos

Modelo de Dominio

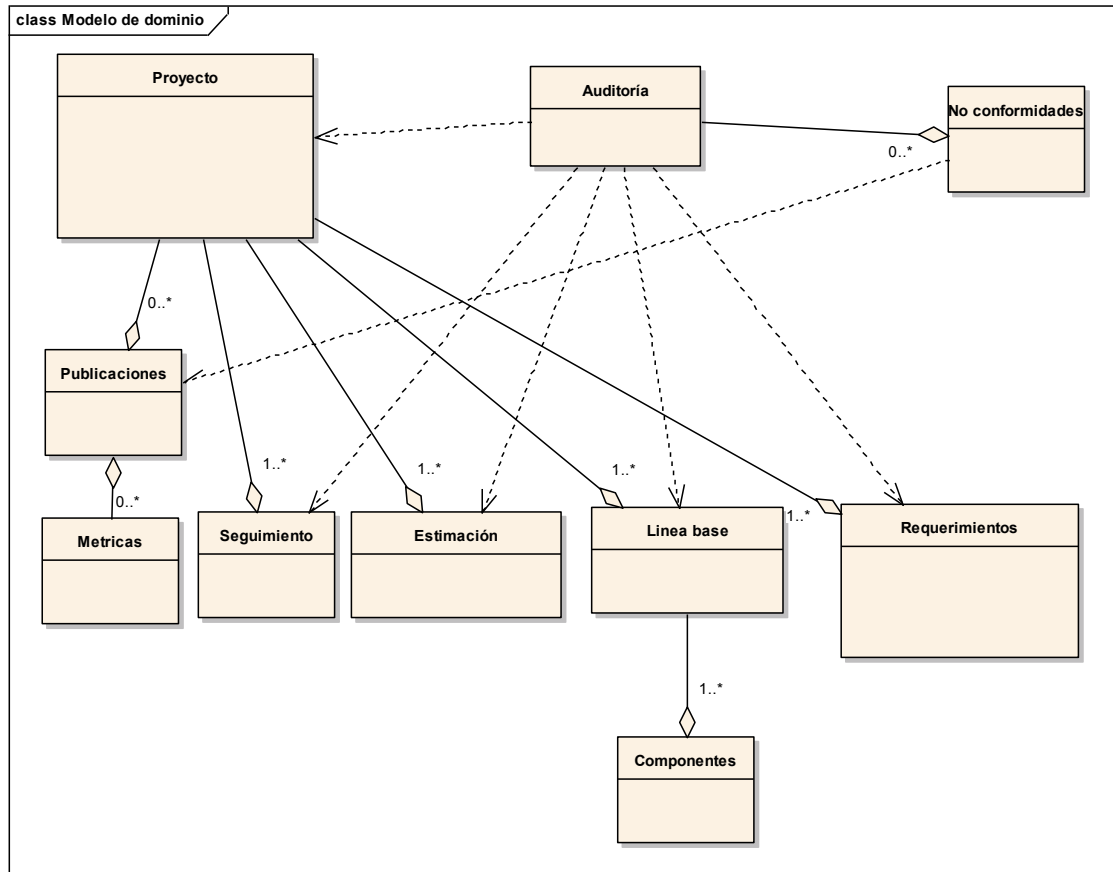


Figura 3 MET Diagrama de dominio.

El modelo de dominio es un diagrama estándar de UML [UML, OMG] y se define como un modelo de clases simplificado. En este diagrama se muestran elementos del dominio como Proyecto y Auditoría. Se muestra también una relación de dependencia entre ellos sin nombre y mostrando la cardinalidad en un sentido.

Este diagrama se explica de la siguiente forma: MET modela proyectos que se estiman, que tienen requerimientos, que se les hace seguimiento, a los que se les traza líneas base sobre componentes, a los que se le publican métricas. Todo esto se audita y si hay no conformidades se las publican.

MET Modelo de Requerimientos

RFMET001 Administrar especificaciones de requerimientos (SRS)

Descripción
Registrar especificaciones de requerimientos (SRS) prescribiendo la forma mediante un template y definiendo el lugar donde persistirlo.

Este es uno de los trece requerimientos de MET. Todos tienen el mismo estilo. MET tiene que prescribir como se escriben los requerimientos de un proyecto y el lugar donde se guardan los mismos.

La lista de todos los requerimientos que resuelve MET puede verse en el Anexo I, MET - Modelo de requerimientos.

Capítulo 6

MET MODELO DE CASOS DE USO

Este capítulo presenta el Modelo de Casos de Uso de MET, el mismo tiene las mismas secciones que cualquier producto desarrollado en el centro de desarrollo. Es el artefacto más importante de MET. La lista de casos de es la columna vertebral de la capacitación dictada a todos los empleados del centro de desarrollo. Los casos de uso de auditorías muestran en que puntos se controla objetivamente la adhesión a MET.

Un template para casos de uso describe el formato y las secciones del documento. En nuestro caso, en el template se incluyen guías para que el analista escriba casos de uso apropiados. Para más información sobre como escribir casos de uso, ver [COCKBURN, 2000] y [BRAMBLE, 2001].

Al igual que en el capítulo anterior, no se incluyen todo el documento. A continuación veremos que secciones tiene un modelo de casos de uso, para poder leer el anexo si fuera de interés del lector entender MET al detalle.

Un modelo de casos de uso tiene las siguientes secciones:

- **Breve descripción funcional:** Esta sección es una descripción que explica al lector, en una carilla, qué uso va a tener el sistema.
- **Diagrama de Casos de Uso:** Este es el único diagrama prescripto por UML 2.0 para casos de uso. El mismo establece que cada caso de uso se dibuja como una elipse, los actores como el dibujo de una persona que hace un niño de jardín de infantes y entre un actor y un caso de uso se dibuja una línea que significa que ese actor inicia ese caso de uso. Además, entre los casos de uso puede haber una relación de uso o extensión, para poder agrupar porciones de casos de uso que se repiten.

- **Orden común de ejecución de Casos de Uso:** Este diagrama no es estándar de UML y es un tema controversial en el centro de desarrollo. Los casos de uso no deberían tener orden (según la teoría), sin embargo hace algunos años, cuando decidimos qué escribir en un modelo de casos de uso, nos pareció interesante incluir este diagrama porque ayuda mucho a entender el problema que se está modelando.
- **Referencias:** Esta sección incluye referencia a objetos físicos como direcciones http, file servers, etc.
- **Descripción de Casos de Uso:** Esta es la lista de casos de uso y cada uno se describe en una tabla de estímulo del actor, respuesta. El hecho de usar tablas no es ningún estándar de mercado ni hay dos organizaciones que escriban sus casos de uso igual. UML no prescribe nada al respecto ya que es un estándar para hacer diagramas. En el Centro de desarrollo nos pareció la mejor forma de hacerlo. Entre otras alternativas, analizamos la posibilidad de usar una narración, un diagrama de estados UML o un pseudo algoritmo.

A continuación se incluye la breve descripción funcional, el orden común de ejecución de los casos de uso y un caso de uso a modo de ejemplo. El documento completo se incluye en esta tesis y es el Anexo II.

MET Modelo de Casos de uso

Breve descripción funcional

El modelo de casos de uso de MET tiene por finalidad identificar el conjunto de actividades desarrolladas en el Centro de desarrollo al gestionar proyectos. Estas actividades se dictan desde el área de Calidad con el objetivo final de estandarizar los procesos de gestión aplicando el concepto de Calidad Total, generando acciones a realizar, registros de las actividades realizadas, control sobre el cumplimiento de los procedimientos y registros sobre el control realizado.

La *breve descripción funcional* es breve pero se desprenden, a partir de ella, algunos comentarios. El primero es que MET fue elaborado en el área de calidad o quality assurance. Esto no es así en todas las empresas. A mi juicio, la mejor manera de hacerlo es teniendo un grupo encargado de la metodología o SEPG (Software Engineering Process Group) que sea el dueño del proceso mientras que calidad controla su aplicación o adherencia al mismo.

También es llamativa la referencia a Calidad Total y esto es explicable analizando a quienes hicimos MET. Uno soy yo, el redactor, que hice mis primeras herramientas en calidad con Total Quality Management (TQM), y el otro, el revisor, es un analista que hizo cursos de TQM y es Green Belt Six Sigma. Estas son técnicas o modelos estrechamente relacionados con el aseguramiento de la calidad en una empresa, pero están fuera del alcance de este trabajo. Sin embargo, mi intención es mostrar el documento tal cual está por dos razones: como evidencia del trabajo hecho y para reutilizarlo.

MET Modelo de Casos de uso

Orden común de ejecución de Casos de Uso

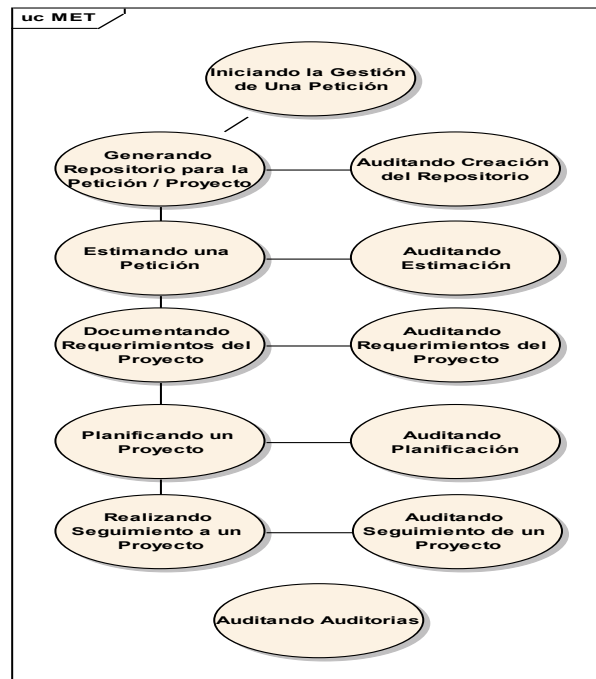


Figura 4 MET Orden común de ejecución de Casos de Uso.

Este diagrama muestra el ciclo de vida de un proyecto. El mismo se inicia como una petición de un cliente, luego el área de infraestructura crea un repositorio para el mismo y QA controla tal tarea.

Una vez aceptada la petición, el jefe de proyecto la estima y QA controla la estimación. Después el jefe de proyecto escribe los requerimientos y planifica y, nuevamente, QA controla. Durante la ejecución del proyecto hay actividades de seguimiento y auditoría desde QA de las mismas. También hay una actividad de QA de QA o auditoría de auditorías. Esta es una práctica evaluada por el modelo CMMI, es decir, hay que auditar que se hagan las auditorías.

MET Modelo de Casos de uso

Descripción de Casos de Uso

CU01. Iniciando la Gestión de Una Petición

Breve descripción.

La petición es el disparador que da origen a un proyecto, el siguiente caso de uso permite conocer de manera detallada las acciones correspondientes al inicio de gestión de una petición.

Actores intervinientes

Código	Nombre
JP	Jefe de Proyecto
PMO	Oficina de Proyectos
SGP	Sistema de Gestión de Proyectos

Precondiciones

Código	Descripción
PRE1	Debe existir una petición por parte de un cliente.
PRE2	El JP determinado por la PMO recibió un e-mail con el ID de la petición.
PRE3	La petición esta en estado Pendiente de aceptación.

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
JP	Accede a la petición	PMO	Confirma aceptación en SGP.
	Identifica el Origen		
	Analiza la petición	SGP	La petición queda pendiente de Estimación.
	Acepta la Petición		
	Comunica a la PMO la aceptación		

Postcondiciones

Código	Descripción
POS1	La petición queda pendiente de estimación.

Flujo Alternativo

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
JP	Accede a la petición	PMO	Confirma el rechazo en SGP.
	Identifica el Origen		
	Analiza la petición	SGP	La petición queda Rechazada
	Rechaza la petición		
	Comunica a la PMO el rechazo		

Postcondiciones

Código	Descripción
POS1	La petición queda rechazada.

Este caso de uso modela el inicio de un proyecto. El caso de uso tiene una breve descripción para ayudar al lector a entender el caso de uso. Tiene actores y, en este caso, tiene un actor o sistema donde se modela la respuesta, a veces en la oficina de proyectos (PMO), a veces en el sistema de gestión de proyectos (SGP).

El caso de uso tiene precondiciones, poscondiciones y flujos alternativos. Este caso de uso, si bien parece sencillo prescribe una cantidad importante de pasos en el ciclo de vida de un proyecto. Todo empieza con una petición, el jefe de proyecto es definido por la Oficina de Proyecto, el Jefe de Proyecto puede aceptar o rechazar la petición. Parecen cosas sencillas, pero son fundamentales para el funcionamiento de un Centro de Desarrollo. El lector en este momento, de trabajar en un lugar donde se desarrolla software, debería preguntarse: cómo empieza un proyecto, como se asigna el responsable, tengo registrados todos los prospectos o candidatos a proyectos, todos los proyectos, los rechazados. Independientemente de la certificación, poder contestar alguna de estas preguntas o la mayoría, parece ser importante en una casa que desarrolla software.

EL SRA (SISTEMA DE REGISTRO DE AUDITORÍAS)

EL SRA es el sistema de registro de auditorías y modela las actividades de QA. Para entender la funcionalidad del mismo, y en coherencia con el trabajo hecho en esta tesis, vamos a ver los casos de uso del SRA. A diferencia de lo hecho para MET, en esta sección sólo se enumerará la lista de casos de uso. El desarrollo del SRA *como producto abierto* no es parte de esta tesis, sin embargo en la UNLP, hay una tesis de grado en desarrollo para hacerlo.

Al mostrar la lista de casos de uso que resuelve el SRA se puede entender su funcionalidad. Además se incluirán algunas pantallas del mismo, dado que el SRA existe y está en funcionamiento en el Centro de Desarrollo y es propiedad del Centro de Desarrollo. Al igual que con MET, el hecho de automatizar las actividades de calidad en un sistema y escribir el análisis de dicho sistema ayuda a comprender el problema, a capacitar a nuevos analistas de QA y a dejar evidencia para mostrar en una certificación o para distribuir tareas y que un analista pueda continuar el trabajo hecho por otros.

El hecho de que el SRA exista y esté en funcionamiento es una demostración empírica de su utilidad. El hecho de usar la evidencia generada por el SRA en una evaluación y que esta evaluación reconozca la efectividad de dicho sistema le da sentido a la idea de desarrollarlo y usarlo. Por ser un producto del Centro de Desarrollo, el software no puede incluirse como uno de los entregables de esta tesis.

Los casos de uso como el consultando próxima auditoría (CU04) o agendar una auditoría (CU022) automatizan tareas propias del trabajo de QA. Un proyecto que está en ejecución, por ejemplo, cada quince días tiene una tarea de seguimiento. Esta tarea hay que auditarla y para eso el analista puede agendarse la auditoría o puede sugerírselo el sistema.

El SRA genera evidencia del trabajo realizado en SQA. Si revisamos la definición de la IEEE del aseguramiento de la calidad:

“Una guía planificada y sistemática de todas las acciones necesarias para proveer la evidencia adecuada de que un producto cumple los requerimientos técnicos establecidos.

Un conjunto de actividades diseñadas para evaluar el proceso por el cual un producto es desarrollado o construido.” [IEEE, 1990]

También podemos ver otras definiciones como la del SEI:

“El aseguramiento de la calidad del software provee claro control del proceso que está siendo usado por el proyecto y del producto que se está construyendo.” [PAULK, CMM1995]

Y finalmente una definición que se ajusta muy bien a lo que resuelve el SRA. Schulmeyer y McManus definen SQA como

“Las actividades sistemáticas que proveen evidencia de la capacidad o disponibilidad de uso del producto de software total.” [SCHUL, 1992]

Veamos entonces la lista de casos de uso.

Casos de uso de SRA

CU01. Dando de alta un proyecto
--

Permite crear un nuevo proyecto con los datos de la petición que ha sido estimada o aceptada.

CU02. Monitoreando un proyecto

Permite ver en una lista los proyectos en gestión y los finalizados.

CU03. Modificando un proyecto

Permite modificar datos de un proyecto.

CU04. Consultando la próxima auditoría de un proyecto-Pendiente
--

Permite consultar la próxima auditoría de un proyecto. A partir de que se crea un proyecto en SRA comienzan las auditorías, dependiendo del avance que tenga el proyecto cada auditoría tiene que realizarse a tiempo. Esta consulta pretende anticiparnos e informarnos la próxima auditoría a realizar en cada proyecto.

CU05. Dando de alta un empleado
--

Permite crear un nuevo empleado del Centro de desarrollo.

CU06. Consultando empleados
Permite ver en una lista todos los empleados del Centro de desarrollo.
CU07. Modificando un empleado
Permite modificar datos de un empleado.
CU08. Dando de alta un aspecto
Permite crear los aspectos que sirven para describir a las no conformidades que pueden tener los proyectos.
CU09. Consultando un aspecto
Permite ver en una lista todos los aspectos que sirven para auditar proyectos y pueden generar no conformidades.
CU010. Modificando un aspecto
Permite modificar datos de un aspecto.
CU011. Dando de alta un tipo de auditoría
Permite crear los tipos de auditoría que sirven para auditar las peticiones. En su creación se tiene en cuenta el estado de la petición.
CU012. Consultando un tipo de auditoría
Permite ver en una lista todos los tipos de auditorías.
CU013. Modificando un tipo de auditoría
Permite modificar datos de un tipo de auditoría.
CU014. Auditando un proyecto
Permite auditar uno o más proyectos en gestión.
CU015. Registrando una no-conformidad
Permite asociar no conformidades que surgen al auditar uno o más proyectos en gestión.
CU016. Escalando una no-conformidad
Permite escalar o notificar a los responsables del proyecto cuando una o varias no-conformidades están sin solución o abiertas durante el lapso de tiempo que tienen de acuerdo al tipo de auditoría.
CU017. Cerrando una no-conformidad
Permite cerrar una o varias no-conformidades de proyecto/s.

CU018. Realizando la auditoría de auditorías

Permite realizar un meta control de todas las auditorías realizadas a los proyectos.

CU019. Configurando orígenes

Este caso de uso prevé que se configuren los distintos orígenes válidos de un proyecto.

CU020. Tipos de proyecto

Este caso de uso prevé que se configuren los distintos tipos de proyectos válidos.

CU021. Configurando nivel de escalas

Este caso de uso prevé que se configuren los distintos niveles de escalas que se usan al escalar.

CU022. Agendando auditorías

Este caso de uso permite agendar auditorías y disparar un aviso o alarma recordando la auditoría a realizar.

Pantallas de SRA

El SRA es el sistema que utiliza el auditor de QA. El auditor llega a la mañana y abre la lista de proyectos. En esta lista, aparecen en rojo los proyectos que tienen no conformidades pendientes.

La siguiente es la pantalla que muestra la lista de proyectos. Estos son todos los proyectos auditable desde el SRA. En esta pantalla se pueden ver las distintas acciones que se pueden hacer en el sistema. El Panel de control muestra la lista de proyectos, mostrando para cada uno el estado en el que está, la descripción del mismo, un código interno, la fecha que se dio de alta y quien es el jefe de proyecto.



Id Petición	Evento	Descripción	Cod Petición	Fecha Alta	Jefe Proyecto
73036	Petición Aceptada	Mediación campo vacío de evento MOC y MTC	R-ARG-08289	2008-12-12 09:01:37	J_SUPPICH
73818	Petición Aceptada	Apl de std. de seg. al manejo de claves de usuarios en COINS	R-ARG-08282	2008-12-04 10:23:10	RDSUCATTE
73412	Planificación Realizada	Testing CC3 y puntos - SGU Impresión y Distribución	R-ARG-08280	2008-11-28 09:45:05	G_COCINA
73267	Planificación Realizada	Modificaciones en el proceso generación de archivos NRTDE	R-ARG-08278	2008-11-21 17:15:33	A_ROTMAN
73042	Planificación Realizada	Testing Altamira	R-ARG-08279	2008-11-21 16:24:38	G_COCINA
72838	Petición Finalizada	Extractor de Datos	R-ARG-08277	2008-11-13 15:15:59	G_COCINA
72949	Petición Aceptada	Redondeo tráfico Datos a k-bytes	R-ARG-08275	2008-11-13 13:48:43	J_SUPPICH
71756	Estimación Aprobada	Capacitación funcional y de la aplicación INX	R-ARG-08276	2008-11-12 16:06:38	RDSUCATTE
72923	Petición Aceptada	Definición estándares Desarrollo C++	R-ARG-08273	2008-11-12 10:43:10	J_SUPPICH
72850	Petición Finalizada	Certificar el uso del rol de base de datos OPER en CC3	R-ARG-08272	2008-11-12 10:35:25	G_COCINA
72789	Petición Finalizada	SGU-Facturación-Consultoría-Estimación de Cambios	R-ARG-08271	2008-11-07 13:43:58	C_MORENO
72788	Planificación Realizada	SGU-Facturación-Paralelo de Ciclo 11	R-ARG-08270	2008-11-07 13:41:51	G_COCINA
72781	Planificación Realizada	SGU Facturación - Migración FWK 5.50	R-ARG-08269	2008-11-07 12:19:32	C_MORENO
72543	Planificación Realizada	SGU-Incorporar impresión del operador de retención	R-ARG-08268	2008-11-05 15:31:30	C_MORENO
72541	Planificación Realizada	Desarrollo Web de Compras 2	R-ARG-08266	2008-11-05 13:04:18	A_DZEMBROVSKY
72539	Estimación Aprobada	Consultoría - CATER en Canales Indirectos	R-ARG-08263	2008-11-04 13:59:27	A_DUBARRY
72371	Planificación Realizada	Eliminación de Puntos a Socios dados de baja	R-ARG-08264	2008-11-04 13:44:09	A_QUINTILI
72489	Estimación Aprobada	Certificación de la funcionalidad de BAHQT Argentina	R-ARG-08265	2008-11-04 13:37:49	G_COCINA
71981	Proyecto en Ejecución	Consultoría PRY 33237-Cons Detalle TLRD	R-ARG-08260	2008-10-28 11:53:21	J_SUPPICH
72069	Petición Aceptada	Soporte para el proyecto Migración a Itanium	R-ARG-08261	2008-10-28 09:55:40	J_SUPPICH
71810	Estimación Aprobada	Nuevo Movistop o parecido	R-ARG-08259	2008-10-20 00:00:00	J_SUPPICH
71856	Petición Finalizada	Implementación de Web de Compras	R-ARG-08258	2008-10-18 00:00:00	A_DZEMBROVSKY

Figura 5 SRA Lista de proyectos

Al elegir un proyecto en particular se puede obtener las auditorías que tiene y las no conformidades que tiene. La siguiente es la pantalla que muestra las auditorías hechas en cada etapa y las no conformidades levantadas.

Auditar Estimación		2008-11-17 09:28:14		X
No evidencia fecha y preparado por		2008-11-17 09:28:24		
Nivel Escala	Fecha Alta	Observación	Estado	
Escala Jefe	2008-11-17 09:28:24	Sin observación adicional	Cerrada	
Escala Gerente	2008-11-21 09:30:38	Sin observación adicional	Cerrada	
Escala Staff	2008-11-20 09:04:57	Sin observación adicional	Cerrada	

Auditar Rq Proyectos		2008-11-17 09:28:29		X
No evidencia fecha y preparado por		2008-11-17 09:28:37		
Nivel Escala	Fecha Alta	Observación	Estado	
Escala Jefe	2008-11-17 09:28:37	Sin observación adicional	Cerrada	
Escala Gerente	2008-11-21 09:30:40	Sin observación adicional	Cerrada	
Escala Staff	2008-11-20 09:14:00	Sin observación adicional	Cerrada	

Auditar Plan de Proyecto		2008-12-12 09:26:04		X

Auditar Creación de Repositorio		2008-11-07 09:28:42		X

Figura 6 Lista de auditorías, no conformidades y niveles de escala

El SRA automatiza las tareas de un auditor de calidad. Cada vez que empieza un nuevo proyecto, se da de alta en SRA. Esto genera un directorio en SVN, el sistema versionador para generar la información del proyecto. A cada *paso* del proyecto, el analista de QA se entera y le hace la auditoría. De haber algo no conforme a lo esperado, según prescribe MET, el auditor genera una no conformidad. La conformidad le llega por email al jefe de proyecto. Al tiempo, el auditor controla y si no encuentra solución de la misma, le llega al gerente y finalmente se trata en el comité de dirección.

El SRA es un buen lugar, como software, para *configurar* tareas automáticas. Por ejemplo, cuando se acepta un proyecto, se elabora un plan, en el plan se completa una tabla con la lista de fechas y artefactos a liberar en esa fecha. El auditor de QA, se agenda estas fechas para ir un día después a ver si al liberar, generaron en la carpeta salida los paquetes a entregar. Un proceso que busque estas tareas de control de liberaciones agendadas, que le avise a soporte que se libera un cambio, puede automatizar el aviso a soporte de una liberación. En ese sentido, el SRA puede servir para automatizar tareas a partir de eventos de proyectos.

La siguiente pantalla muestra la lista de no conformidades abiertas. Las no conformidades se escalan en tres niveles para su resolución. Se notifican al jefe, luego al gerente y luego al comité de dirección.

Auditoria - Windows Internet Explorer

SRA
Sistema de registro de auditorías

Panel de Control | Agenda de Gestión | Alta Proyecto | Alta Genérica | Reporte NC | Ayuda | Salir

FZOCARATO

LISTA DE NO CONFORMIDADES ABIERTAS

Exportar a Excel

Id Petición	Descripción	Tipo	Sts. Proyecto	Jefe Proyecto	Auditoria	No conformidad	Escala	Fecha Alta	Usuario Alta
70494	SOU - Distribución - Soporte a la Implantación y Certificación	IMPLANTACIÓN	Proyecto en Ejecución	A_QUINTILI	Auditar liberaciones	No evidencia entregables	Escala Gerente	2008-12-18 12:40:49	FZOCARATO
71855	Implantación de Web de Compras	IMPLANTACIÓN	Petición Finalizada	A_DZEMBROVSKY	Auditar Cliente de proyecto	No evidencia doc. obligatorios de Proyecto	Escala Gerente	2008-12-15 09:08:42	FZOCARATO
71855	Implantación de Web de Compras	IMPLANTACIÓN	Petición Finalizada	A_DZEMBROVSKY	Auditar Plan de Proyecto	No evidencia Doc Plan de Proyecto	Escala Staff	2008-12-15 09:09:20	FZOCARATO
68996	Prototipo Proceso de Inyección Genérico Sistema Coins	DESARROLLO	Proyecto en Ejecución	J_FERAUD	Auditar Cambio de Alcance	No evidencia cambios en documentación	Escala Staff	2008-11-04 09:35:47	FZOCARATO
71855	Implantación de Web de Compras	IMPLANTACIÓN	Petición Finalizada	A_DZEMBROVSKY	Auditar Estimación	No evidencia Doc. Estimación	Escala Staff	2008-10-28 12:18:40	M_LATTANZIO
68996	Prototipo Proceso de Inyección Genérico Sistema Coins	DESARROLLO	Proyecto en Ejecución	J_FERAUD	Auditar Plan de Proyecto	No evidencia Doc Plan de Proyecto	Escala Staff	2008-10-22 00:00:00	M_LATTANZIO
68992	Capacitación utilitarios MDR - Uruguay	CAPACITACIÓN	Proyecto en Ejecución	J_BLANCO	Auditar Plan de Proyecto	Documento Incompleto	Escala Staff	2008-10-21 18:11:51	LOPEZ_R
68996	Prototipo Proceso de Inyección Genérico Sistema Coins	DESARROLLO	Proyecto en Ejecución	J_FERAUD	Auditar Rq Proyectos	No evidencia RQ Proyecto	Escala Staff	2008-09-18 00:00:00	M_LATTANZIO
68996	Prototipo Proceso de Inyección Genérico Sistema Coins	DESARROLLO	Proyecto en Ejecución	J_FERAUD	Auditar Estimación	No evidencia Doc. Estimación	Escala Staff	2008-09-18 00:00:00	M_LATTANZIO

Figura 7 SRA Lista de no conformidades abiertas

Capítulo 8

LA EXPERIENCIA DE CERTIFICAR CMMI NIVEL 2 Y LA IMPLANTACIÓN DE MET

Este capítulo resume la experiencia de certificar e intenta revisar la historia de cómo se hizo MET, ver qué le pasó a MET en cada evaluación (si hubo que adecuarla, si los empleados hicieron referencia a MET, si aparecieron nuevos artefactos). Al igual que cualquier producto informático, hubo que *desplegar*, *instalar* o *implantar* MET. El detalle de la implantación se cuenta en este capítulo.

La certificación en el centro de desarrollo tuvo tres etapas. Una primera evaluación, de nombre PMAM, donde la P es la inicial de la empresa que hizo la evaluación y MAM quiere decir Multipurpose-Assessment Method.

Durante la evaluación el evaluador califica cada área de proceso. Para hacer esto el evaluador entrevista empleados del centro de desarrollo, revisa evidencia y revisa la metodología. Durante un PMAM, que es un método propio de la consultora, se obtienen calificaciones en colores que indican el estado de esa área.

PMAM 1 (7 de Septiembre de 2007)

El cuadro que refleja el resultado de esta evaluación es el siguiente:



Figura 8 Resultado PMAM 7 de Septiembre de 2007

Y el significado de los colores es el siguiente:

4	Institucionalizado		90%-100%
3	Implementado		61%-89%
2	Definido		41-60%
1	En curso		21-40%
0	Evidencia insuf.		0-20%

Figura 9Significado colores PMAM

Como se ve, los mayores problemas estaban en PPQA. Y las observaciones o findings de PPQA fueron las siguientes:

- O1: No se evidencia la actividad de evaluación de procesos contra estándares.
- O2: A excepción de revisiones de código fuente (CheckStyle) no se evidenciaron revisiones de adhesión a estándares de otros productos de trabajo (ej Diseño)
- O3: No se evidenció la comunicación de no conformidades de evaluación de proceso, ni su seguimiento hasta el cierre.

Estas observaciones son muy pocas pero muy contundentes. No había ni estándares ni chequeo contra los estándares.

Esta evaluación dio origen a MET y al SRA. MET como metodología, inexistente hasta ese momento, y el SRA como sistema que ayudaría a mejorar PPQA.

Implantación de MET

El primer trabajo fue escribir MET. Este fue un trabajo de un equipo chico, dos personas: un redactor y un revisor. La primera versión de MET fue presentada a un grupo pequeño de cinco integrantes del Centro y a la dirección. Hubo pequeñas mejoras a partir de estas dos presentaciones y la versión de MET de diciembre de 2007 estaba lista para ser desplegada en el centro de desarrollo.

Al mismo tiempo, estaba desarrollada la primera versión del SRA¹ y para hacer auditorías de existencia de los tres entregables más sencillos de un proyecto: el plan, los requerimientos y la estimación.

El primer hito importante en la implantación de MET fue capacitar en una sesión de dos horas a cada integrante del centro de desarrollo. Lograr que entendieran la metodología, sus entregables y sus puntos de control. Al mismo tiempo, se publicó en una wiki interna una sección correspondiente a metodología de proyectos.

Implementamos un proceso de chequeo nocturno de proyectos, copiando la idea de las metodologías ágiles conocida como *nighty build*. Este chequeo valida todas las noches que cada proyecto tenga la documentación correspondiente. Además elabora un informe que se publica en la wiki detallando para cada proyecto cuantas auditorías tiene, cuantas no conformidades, quien es el responsable, un link a la documentación.

A medida que íbamos usando la metodología, íbamos adecuándola. Tuvimos que hacer *tailoring*, es decir adecuarla a diferentes tipos de proyectos. No es lo mismo un desarrollo, que una consultoría o que un pedido de estimación. Estos tres tipos de peticiones existen en el Centro de Desarrollo. Una consultoría se estima, se le hace seguimiento, genera un informe, pero no genera software. Un pedido de estimación es un presupuesto. Si bien la

¹ En esta tesis se incluye la especificación del SRA y las principales pantallas del mismo, pero no el SRA ya que es propiedad del Centro de Desarrollo. Una versión *open source* del mismo se desarrolla como tesis de grado en la UNLP.

certificación aplica a proyectos de desarrollo, nos pareció útil modelar estos casos de negocio, ya que parte de nuestro día a día es presupuestar soluciones informáticas.

Tuvimos que ajustar los tiempos para hacer auditorías. Al principio auditábamos a los tres días de aceptada la petición, por ejemplo, pero se nos superponían las etapas. Cambiamos los puntos de control y agregamos auditorías y tareas automatizables en algunos puntos importantes como la generación de líneas base, esto es guardar una *foto* de estimación, requerimientos y plan al momento de la aceptación por parte del cliente.

Y todo esto tuvimos que comunicarlo periódicamente a todo el centro. A veces por mail, a veces con capacitaciones en aula, especialmente para jefes de proyectos, que son claves en un proceso de mejora hacia nivel dos de cmmi. A veces lo hicimos con capacitaciones asistidas por computadora (e-learning).

PMAM 2 (8 de Agosto de 2008)

Después de casi un año, de desarrollar MET, de capacitar a todo el Centro de desarrollo hicimos una segunda evaluación. Los resultados fueron los siguientes:



Como se ve, la evaluación mejoró. Sin embargo hay observaciones que hacen que el verde sea claro y no oscuro. En particular las dos siguientes:

- OG1: Si bien existe la política de seguir MET y existen distintos lineamientos sobre cómo ejecutar y planear los procesos, no se han documentado políticas específicas para los distintos procesos bajo evaluación.

- OG4: No se evidencia la definición del alcance de roles y responsabilidades para los roles de algunos procesos específicos (Encargado de Gestión de la Configuración, Auditor Aseguramiento de Calidad, Responsable de Métricas).

Esto hizo que tuviéramos que definir dos documentos nuevos, que demuestran que MET no es suficiente documentación del proceso. Los tres documentos fueron los siguientes:

- Políticas: Un documento con políticas. Para entenderlo, esta es un ejemplo de una de las políticas incluidas en tal documento: El centro de desarrollo entiende con el término Proyecto a toda actividad que comienza con una solicitud de solución informática o de servicio, que se estime en tiempo y esfuerzo, que se planifique y que se controle, esta solicitud puede o no alcanzar componentes de software.
- Planes: Un documento intermedio entre las políticas y MET. Más abstracto que MET y más concreto que el de políticas. Aquí un ejemplo de lo que se enuncia en el plan: “Se planifica la asignación de tareas por medio de ticket en la herramienta TRAC.”
- Roles: Un documento donde se describen los roles en la organización. Por ejemplo: Jefe de Proyecto - El Jefe de proyecto es la persona responsable de la planificación y gestión de un proyecto. Forma un equipo de trabajo y designa responsabilidades a los integrantes del mismo. Su Rol Principal es que cuenta con las principales funciones de toma de decisiones, supervisión, seguimiento y control de las actividades realizadas durante el ciclo de vida de un proyecto, designando responsables a nivel operativo para la gestión. Define y aprueba planes de gestión, estima tiempo y esfuerzo, apoya y alienta la consolidación y las políticas definidas.

Estos tres documentos, políticas, planes y roles son sencillos de hacer y es un ejercicio interesante dejar plasmar definiciones generales que están en la cabeza de algún director o fueron escritas una minuta de una reunión en un documento de políticas, por ejemplo.

Hubo también otras observaciones, pero la más difícil de resolver tiene que ver con la trazabilidad entre requerimientos y productos de software. Las observaciones son las siguientes:

O1: No siempre se evidencia la revisión de requerimientos por parte del equipo interno de proyecto.

O2: No todos los proyectos tienen release notes, por lo que se perdería parte de la trazabilidad. Definir en qué proyectos es obligatorio.

O3: No se evidencia que se revise si hay inconsistencias entre requerimientos y planes.

Esto quiere decir que no gestionábamos nuestros proyectos por requerimientos. Para eso tuvimos que resolver algunos puntos que no tienen que ver con MET sino con qué hacer para saber a mitad de un proyecto que requerimientos están listos y cuáles no y además saber el impacto de un requerimiento en el software y dado un componente de software poder llegar a qué requerimientos afecta. Este punto, según un panel hecho en el SEPGLA 2008, en Mar del Plata, es el punto de más consultas al SEI. Qué significa trazabilidad.

Lo más relevante para esta tesis es que MET resolvió la metodología escrita y el SRA aportó toda la evidencia requerida sobre auditorías y no conformidades.

SCAMPI (19 de Diciembre de 2008)

Finalmente el 19 de diciembre certificamos CMMI nivel 2,[CMMI, RESULTADOS]. Aparte de observaciones, el proceso revela fortalezas y, para alguien que ha participado en evaluaciones, una fortaleza es un motivo de orgullo en la organización y las fortalezas suelen ser pocas. Las fortalezas fueron tres:

- El uso de TRAC para gestión de requerimientos y trazabilidad
- El SRA para automatizar la tarea de QA
- La PMO para planificación y seguimiento de proyectos.

El segundo punto es relevante para esta tesis. El hecho de automatizar con herramientas informáticas el proceso de desarrollo o alguna parte de él mereció un reconocimiento en una evaluación formal.

Un reconocimiento no formal y que es muy relevante para este trabajo es el comentario de todos los team members sobre la coherencia entre todos los entrevistados con respecto a MET. Esto es una demostración empírica de que la metodología fue aceptada y está *institucionalizada*. En CMMI esto quiere decir que todos los integrantes de la organización conocen y siguen la metodología.

TRABAJO FUTURO Y CONCLUSIONES

En este capítulo se detallan las distintas posibilidades de trabajo futuro y se incluye una sección con conclusiones.

Consolidar y continuar el desarrollo del SRA

El SRA existe y se utiliza en el Centro de Desarrollo. Está hecho en PHP y en el capítulo dedicado al SRA se pueden ver algunas pantallas del mismo. Hacerlo genérico y como un producto cerrado e instalable es un trabajo posible y que actualmente es parte de una tesis de grado en la UNLP.

Para comprender esto veamos un ejemplo de lo que quiere decir hacerlo fácilmente extensible. Hoy el SRA del centro de desarrollo interactúa con SVN y desde el SRA, por ejemplo, se da de alta al iniciar el proyecto el repositorio en SVN. Hacer que el SRA pueda funcionar con otro versionador o que pueda configurarse el comando a generar para crear el repositorio harían del SRA un producto genérico.

Además, hacer de un software *ad-hoc* o *a medida* un producto (que tenga manuales de usuario, de instalación, de configuración, extensible, instalable, etc.) es un trabajo cuyo esfuerzo es de un orden de magnitud mayor (diez veces más) y suele ser, erróneamente, minimizado.

Además, el SRA debería ser utilizable en cualquier software factory, independiente de MET. Este acomplamiento debería ser revisado y evitado al extender el SRA. El SRA debería servir para auditar cualquier metodología, para modelar la relación entre el flujo de tareas del auditor (*workflow*).

Evolucionar MET para CMMI nivel 3

El camino natural después de certificar nivel 2, es certificar nivel 3. Para eso habría que adecuar MET, extendiendo los requerimientos para nivel 3. El nivel 3 tiene muchas áreas de procesos:

- RD: Requirement development
- TS: Technical solution
- PI: Product integration
- VER: Verification
- VAL: Validation
- OPF: Organizational process focus
- OPD + IPPD: Organizational process definition
- OT: Organizational training
- IPM+IPPD: Integrated project management
- RSKM: Risk management
- DAR: Decision analysis and resolution

En una evaluación preliminar efectuada por el lead appraiser se observó que tales áreas de proceso no son actualmente cubiertas por MET a excepción de OPD + IPPD.

Programar MET para CMMI nivel 2

Programar MET para CMMI nivel 2 sería continuar el ciclo normal de desarrollo según los estándares del Centro de Desarrollo para hacer productos y hacer un diseño (modelo de diseño, modelo de clases y modelo de datos), desarrollar un software, hacer manuales de instalación, operación, usuario y configuración.

Incluir en este sistema todo el software para una software factory. Un sistema como el SGP (el Sistema que se utiliza para la Gestión de Proyectos), para hacer proyectos, registrar horas o buscar algún sistema open source para esto (Hoy en el centro de

desarrollo se usa ITG y no es extensible ni integrables). Como en cualquier empresa en el Centro de Desarrollo hay múltiples sistemas y no están integrados. Por ejemplo, cuando en el Sistema de Gestión de Procesos se acepta una petición, le llega un mail a la QAO (oficina de QA) y uno de los integrantes da de alta el proyecto en SRA.. Para una integración adecuada, el SGP debería permitir ejecutar procesos ante eventos y el SRA debería tener un servicio para dar de alta un proyecto. Esta es una integración posible. Construir un sistema es hacer todas las integraciones o hacer un sistema único. O evaluar cuando hacer o usar algo existente. Por ejemplo, hacer algo como SVN no parece tener ningún sentido ya que es estándar de mercado, estable, open source. En el Centro de desarrollo usábamos un sistema propietario de una gran empresa de software a nivel mundial y decidimos reemplazarlo por SVN.

Conclusiones

MET se construyó utilizando técnicas que sirven para construir software. Esto era una idea interesante y desafiante, pero en septiembre de 2007 parecía un juego de palabras con problemas de autoreferencia. Construir la fue sencillo, dada la experiencia de quienes la hicimos a la hora de especificar problemas de negocio. En este caso el negocio era el negocio del Centro de Desarrollo: hacer software.

MET se comunicó en diez sesiones a grupos de aproximadamente quince personas. La aceptación fue relativamente buena (sabemos de la renuencia natural de los *informáticos*, nombre genérico para analista, programadores, arquitectos, testers, líderes de proyecto, analistas de soporte, etc. a cualquier tipo de *metología* o *proceso*) y eso quedó explícitamente reflejado durante las entrevistas llevadas a cabo por el equipo evaluador durante el SCAMPI.

El SRA fue una de las fortalezas marcadas en la evaluación. Este es un detalle no menor, ya que un evaluador, que tiene unas decenas de evaluaciones por año, ve muchos centros de desarrollo y muchas herramientas como para que decida, en una evaluación de madurez, recalcar la importancia de una herramienta.

El SRA fue de mucha ayuda cuando a dos meses del SCAMPI tuvimos que reemplazar al principal analista de QA. El nuevo analista pudo hacer rápidamente su trabajo (una semana) gracias a la automatización del mismo dentro del SRA.

Finalmente certificamos, y vista la evaluación de septiembre de 2007, estábamos muy lejos. Se puede concluir que:

- La experiencia es reutilizable. El hecho de tratar el proceso de desarrollo como software y modelarlo, guardarlo en un repositorio, trazarle líneas bases es cómodo, funciona, es familiar al día a día de un ingeniero de software.
- MET es reutilizable. Los casos de uso de MET se pueden adecuar con poco esfuerzo. Y tal vez no es reutilizable cada fila de cada tabla de cada caso de uso. Se pueden reutilizar casos de uso aislados, se pueden reutilizar los títulos de los casos de uso en su conjunto.
- El SRA es reutilizable. Los tipos de auditoría en el SRA son configurables, esto permite ir ajustando e incrementando el nivel de auditorías en una suerte de progresión partiendo de controles sencillos hacia controles cada vez más profundos. Al principio auditábamos que el documento de estimación existiera, hoy vemos si hay horas en la estimación para la PMO (oficina de proyectos) y si las horas estimadas coinciden con la carta Gantt del proyecto, que se hace en el SGP (Sistema de Gestión de Proyectos).

Estamos empezando a formar un SEPG (Software Engineering Process Group) y cuando en la dirección tuvimos que entender qué tareas tendría este grupo, alguien dijo: “Mantiene MET”. Y alguien agregó: “Soporta MET también”. En un Centro de Desarrollo donde los productos se desarrollan, se mantienen o evolucionan y se soportan, tratar a los procesos del centro como un producto de software más facilita la comprensión y el tratamiento de algo tan difícil en nuestra comunidad de informáticos como lo es la *mejora de procesos*. La certificación fue una GRAN satisfacción y cada uno de estos pequeños hechos son la satisfacción de cada día.

ANEXO I - MET MODELO DE REQUERIMIENTOS

Objetivo

El objetivo de la metodología es definir los procesos para los distintos proyectos llevados a cabo en el centro de desarrollo. La metodología debe servir para:

- Definir un proyecto, es decir definir los roles que participan, los entregables que va a generar y la forma de controlarlo.
- Estimar proyectos, en esfuerzo y fechas de compromiso.
- Hacerle seguimiento a un proyecto, en lo que respecta a riesgos, avances y re estimaciones.
- Asegurar la calidad de los proyectos
- Medir los proyectos
- Definir los entregables de un proyecto, como se versionan y como se agrupan en una línea base.
- Definir la forma de controlar la aplicación de la misma.

Glosario

- **Entregable**
Cualquier artefacto que se genera, se versiona y se usa en una línea base.
- **Rol**
Responsabilidad que cubre una persona en un proyecto.
- **Línea Base**
Publicación de un conjunto de versiones de entregables.
- **Norma**
Pautas definidas en un área con el objetivo de regular las actividades, define el que hacer, dentro de este marco la norma se encuentra representada por el Caso de Uso.
- **Procedimiento**
Conjunto de actividades guiadas con la finalidad de cumplir una Norma, define el como hacer.

- **No-cumplimiento**
Falta de cumplimiento de una Norma.
- **PMO**
Oficina de Proyectos.
- **JP**
Jefe de Proyecto
- **ITG**
Herramienta de gestión de proyectos de tecnología informática.
- **SRS**
Especificaciones de Requerimientos de Sistema

Relación con otros productos

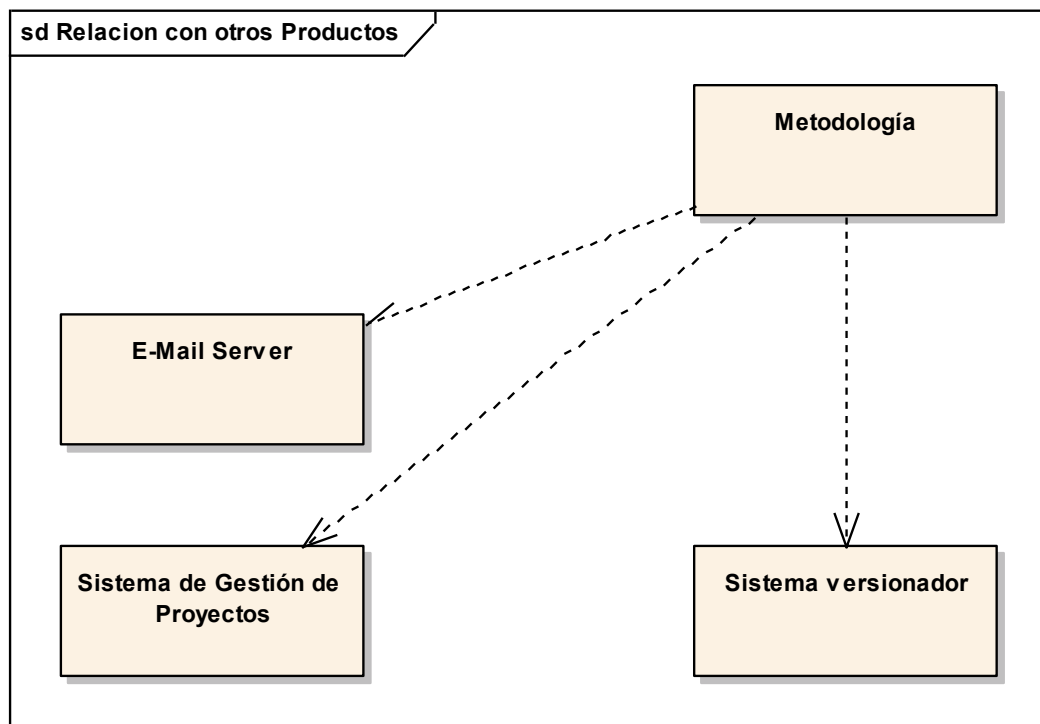


Figura 10 Relación de MET con otros productos.

Relación con el E-Mail Server

Descripción	Servidor de Correo electrónico
Interacción	Relación directa con el uso de la metodología. Es el recurso primario de comunicación escrita.
Es requerido	[SI]

Relación con el Sistema de Gestión de Proyectos

Descripción	Sistema de Gestión de Proyectos
Interacción	Por medio del sistema se reciben las peticiones, se imputan las horas en esfuerzo, se monitorea, se cargan las incidencias, etc. El sistema permite mantener el canal formal, sirve de fuente de información.
Es requerido	[SI]

Relación con el Sistema Versionador de Artefactos

Descripción	Sistema Versionador
Interacción	Permite mantener las versiones de los artefactos entregables de un proyecto.
Es requerido	[SI]

Descripción general del producto

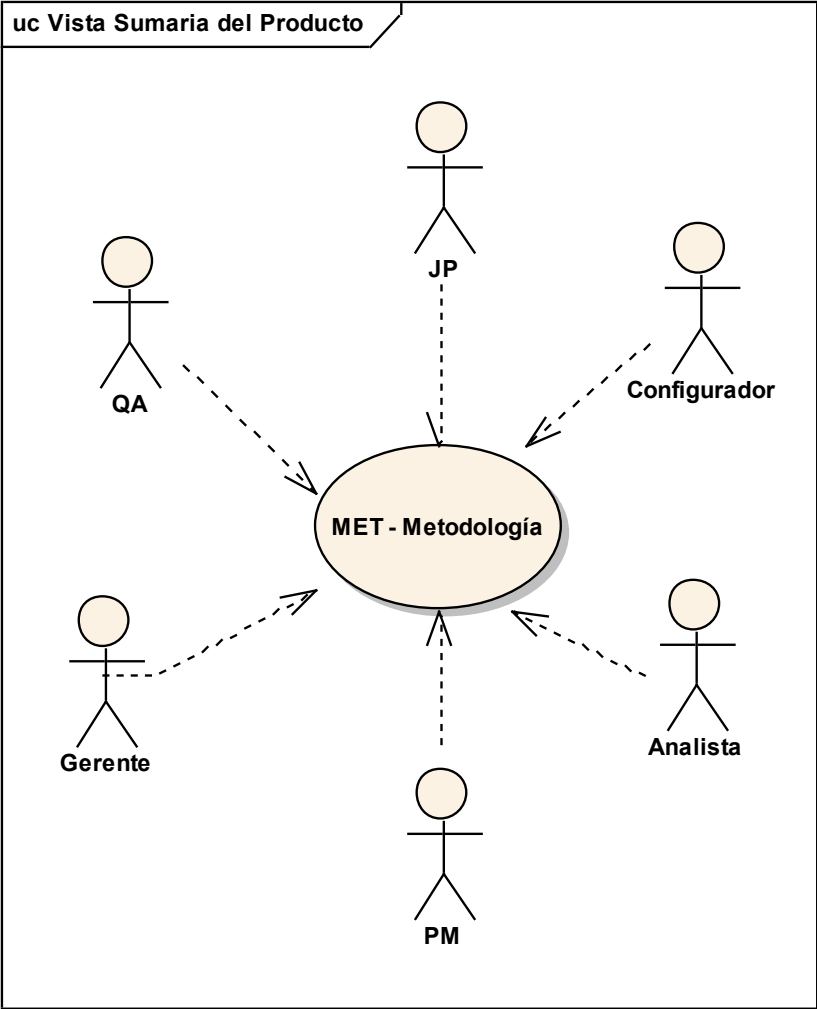


Figura 11 MET Vista Sumaria.

Perfiles de Actores

QA	Encargado de controlar la aplicación de la Metodología.
Gerente	Segundo nivel de escalamiento de no cumplimientos, el jefe reporta al gerente del proyecto.
Analista Programador	Responsable de generar los artefactos técnicos del proyecto, Reporta al jefe de proyecto
Jefe	Reporta al Gerente, primer nivel de Control.
Configurador	Responsables de Generar líneas base.
PM	Responsable de controlar el cronograma. Avance, esfuerzo y fechas.
Analista RQ	Responsable de generar los artefactos más abstractos del proyecto
Analista Soporte	Primera línea responsable de atender las incidencias generadas en producción, responsable de generar la transacción operativa para liberar las soluciones y paquetes de instalación.
Tester	Responsable de certificar el comportamiento funcional del producto.

Modelo de Dominio

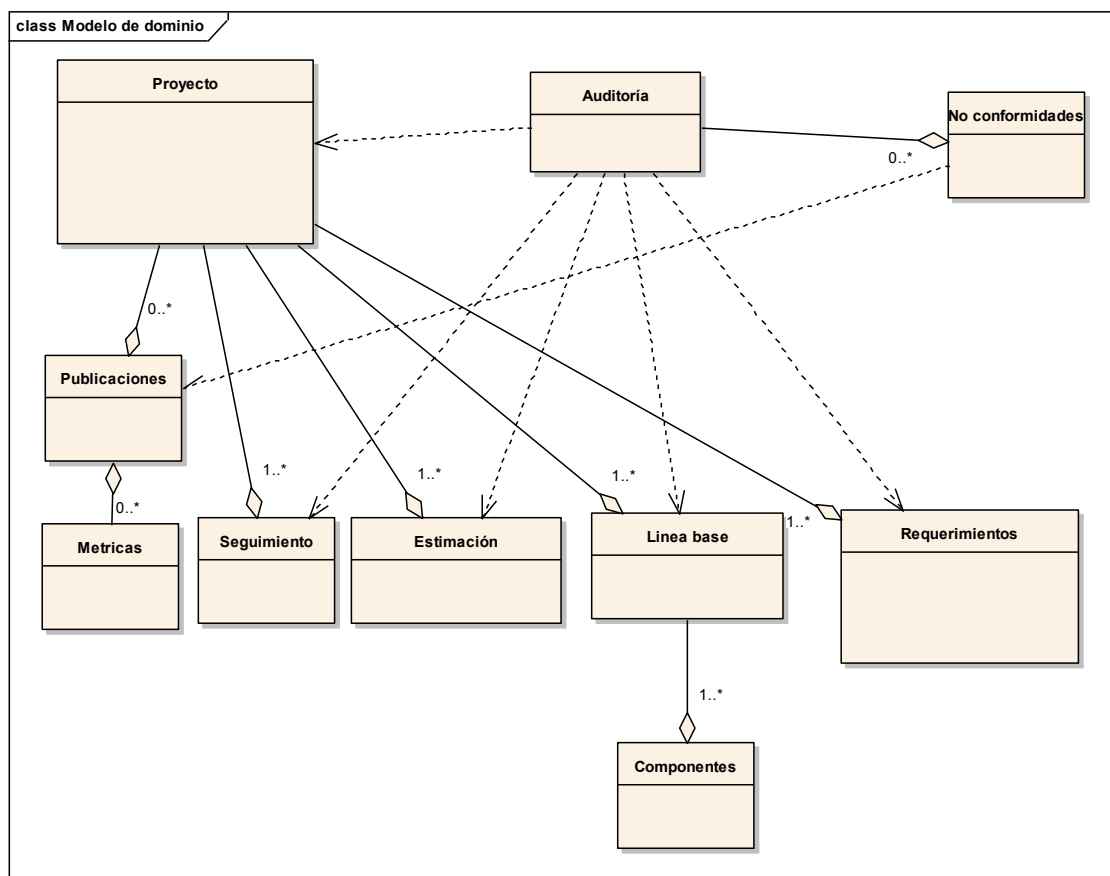


Figura 12 MET Diagrama de dominio.

Supuestos y restricciones

Para el desarrollo de los productos se utiliza la metodología Proceso Unificado – Guiado por casos de uso.

El producto MET, que alcanza a la redacción de las normas y procedimientos, se lo considera un producto más dentro del área.

El producto MET evoluciona por proyectos. Un proyecto de evolución de MET es un proyecto más dentro del centro de desarrollo (tiene un jefe de proyecto, una estimación, seguimientos, etc.)

Requerimientos del Producto

RFMET001 Administrar especificaciones de requerimientos (SRS)

Descripción
Registrar especificaciones de requerimientos (SRS) prescribiendo la forma mediante un template y definiendo el lugar donde persistirlo.

RFMET002 Realizar el Control de los requerimientos

Descripción
Registrar el control de los requerimientos, definiendo la forma, periodicidad y lugar donde persistir el registro del control realizado.

RFMET003 Realizar Estimaciones de proyectos

Descripción
Registrar estimaciones, prescribiendo la forma mediante un template y definiendo el lugar donde persistirlo.

RFMET004. Realizar el Control de Estimaciones

Descripción
Registrar el control de las estimaciones, definiendo la forma, periodicidad y lugar donde persistir el registro del control realizado.

RFMET005 Administrar la planificación de proyectos.

Descripción
Registrar la planificación de proyectos, prescribiendo la forma mediante un template y definiendo el lugar donde persistirlo.

RFMET006 Realizar el Control de la planificación de proyectos.

Descripción

Registrar el control de la planificación, definiendo la forma, periodicidad y lugar donde persistir el registro del control realizado.

RFMET007 Realizar seguimientos de proyectos.

Descripción

Registrar el seguimiento de proyectos, prescribiendo la forma, la periodicidad y definiendo el lugar donde persistirlo.

RFMET008 Realizar el control sobre seguimientos

Descripción

Registrar el control de los seguimientos, definiendo la forma, periodicidad y lugar donde persistir el registro del control realizado.

RFMET009 Realizar la planificación de una línea base

Descripción

Registrar la planificación de una línea base, los artefactos que la forman y el lugar donde se depositan

RFMET010 Realizar el control de la línea base

Descripción

Registrar el control de los artefactos que forman la línea base según su planificación, definiendo la forma, periodicidad y lugar donde persistir el registro del control realizado.

RFMET011 Realizar Métricas

Descripción
Definir un conjunto de métricas, la forma de consolidarlas y la periodicidad de realización.

RFMET012 Realizar Publicación de Métricas

Descripción
Definir la forma, la periodicidad y el lugar donde persistir la publicación del conjunto de métricas.

RFMET013 Realizar el control sobre Publicación de Métricas

Descripción
Registrar el control de la publicación de métricas, definiendo la forma, periodicidad y lugar donde persistir el registro del control realizado.

ANEXO II - MET MODELO DE CASOS DE USO

Breve descripción funcional

El modelo de casos de uso de MET tiene por finalidad identificar el conjunto de actividades desarrolladas en el Centro de desarrollo al gestionar proyectos. Estas actividades se dictan desde el área de Calidad con el objetivo final de estandarizar los procesos de gestión aplicando el concepto de Calidad Total, generando acciones a realizar, registros de las actividades realizadas, control sobre el cumplimiento de los procedimientos y registros sobre el control realizado.

Diagrama de Casos de Uso

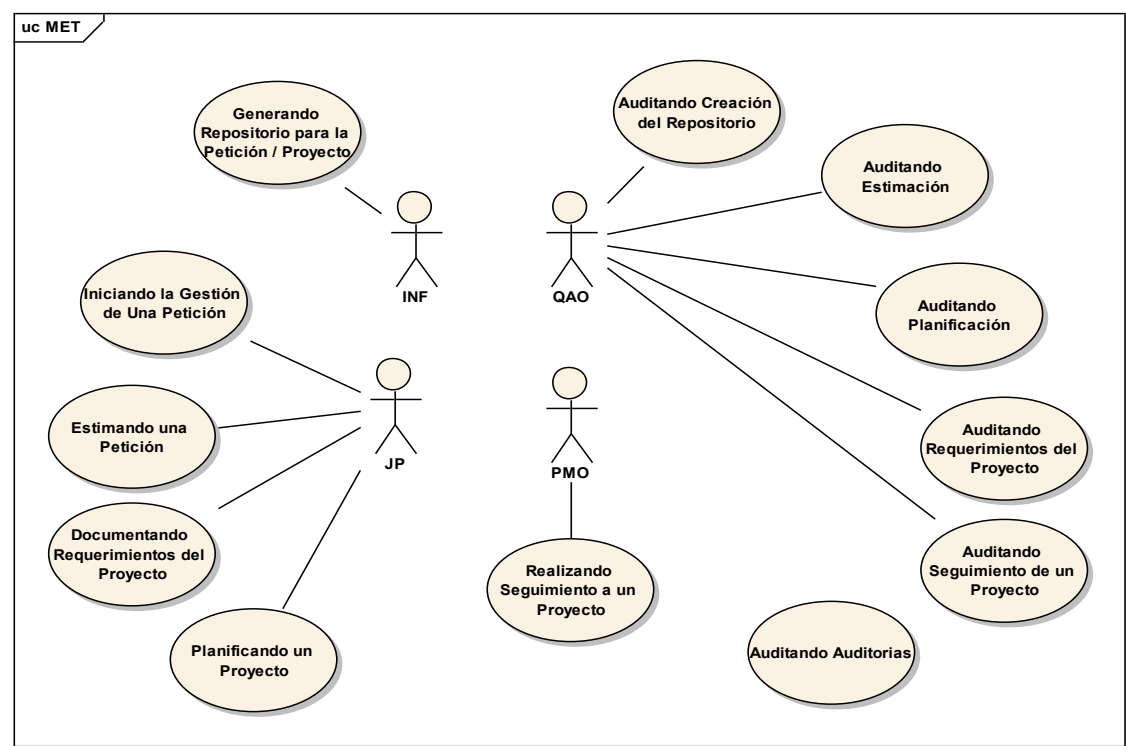


Figura 13 MET Diagrama de casos de uso.

Orden común de ejecución de Casos de Uso

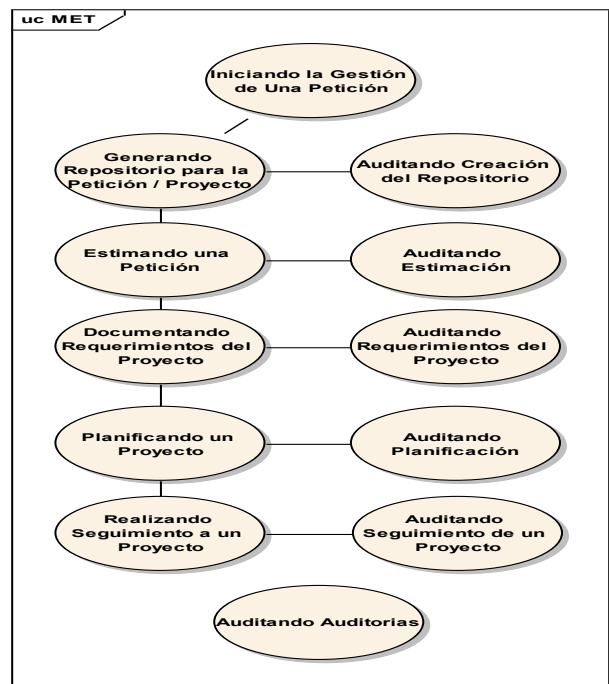


Figura 14 MET Orden común de ejecución de Casos de Uso.

Referencias

El siguiente párrafo tiene por objetivo describir la ubicación física de los entregables, herramientas de gestión y direcciones mencionadas en este documento.

- Documentos del Proyecto: svn://10.204.130.22/PRO_[IdRequest]/trunk
- SRA = <http://bahq55/sra>
- SGP = ITG <http://pluton>
- SCM = SVN System Configuration Management
- Dirección de correo Oficina de Calidad (QAO) = QAO-TMAS
- Dirección de correo Oficina de Proyecto (PMO) = PMO-TMAS
- Dirección de correo Infraestructura = (INF) = INF-TMAS
- Acceso a Template para documentar: <http://bahq55:8090/wiki/Proyectos>

Descripción de Casos de Uso

CU01. Iniciando la Gestión de Una Petición

Breve descripción.

La petición es el disparador que da origen a un proyecto, el siguiente caso de uso permite conocer de manera detallada las acciones correspondientes al inicio de gestión de una petición.

Actores intervinientes

Código	Nombre
JP	Jefe de Proyecto
PMO	Oficina de Proyectos
SGP	Sistema de Gestión de Proyectos

Precondiciones

Código	Descripción
PRE1	Debe existir una petición por parte de un cliente.
PRE2	El JP determinado por la PMO recibió un e-mail con el ID de la petición.
PRE3	La petición esta en estado Pendiente de aceptación.

Flujo principal

Acción Actor	Respuesta del Sistema
--------------	-----------------------

Código	Descripción	Código	Descripción
JP	Accede a la petición	PMO	Confirma aceptación en SGP.
	Identifica el Origen		
	Analiza la petición	SGP	La petición queda pendiente de Estimación.
	Acepta la Petición		
	Comunica a la PMO la aceptación		

Postcondiciones

Código	Descripción
POS1	La petición queda pendiente de estimación.

Flujo Alternativo

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
JP	Accede a la petición	PMO	Confirma el rechazo en SGP.
	Identifica el Origen		
	Analiza la petición	SGP	La petición queda Rechazada
	Rechaza la petición		
	Comunica a la PMO el rechazo		

Postcondiciones

Código	Descripción
POS1	La petición queda rechazada.

CU02. Generando Repositorio para la Petición / Proyecto.

Breve descripción.

Una vez aceptada la petición, comienzan las actividades de gestión, dentro del flujo de trabajo identificamos la necesidad de crear un repositorio para albergar los entregables y utilitarios propios del proyecto, en el siguiente caso de uso se detallan las acciones de la creación del repositorio.

Actores intervinientes

Código	Nombre
INF	Infraestructura
PMO	Oficina de Proyectos
SCM	System configuration Management

Precondiciones

Código	Descripción
PRE1	Debe existir una petición aceptada por parte de un JP

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
PMO	Solicita vía e-mail la creación del repositorio, copia a la QAO	INF	Crea el repositorio
	Informa ID de Request, Código de petición, Descripción y Jefe de proyecto.		Lo denomina: PRO_[NRO], donde el [NRO] representa al ID de Request.
INF	Informa sobre la efectiva creación del repositorio respondiendo el e-mail recibido.	SCM	Crea los subdirectorios: (./Otros) y (./Salida) dentro del repositorio.
			Alta de repositorio.

Postcondiciones

Código	Descripción
POS1	Alta de repositorio para petición / Proyecto.

CU03. Auditando Creación del Repositorio.

Breve descripción.

Paso seguido a la creación del repositorio se desencadena la primera actividad de control. A continuación se detallan las acciones correspondientes a la auditoría sobre la creación del repositorio.

Actores intervinientes

Código	Nombre
QAO	Oficina de Calidad
SRA	Sistema de registro de Auditorías

Precondiciones

Código	Descripción
PRE1	Debe existir una petición aceptada por parte de un JP

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	96 horas posterior a la aceptación de la petición verifica se haya creado el repositorio.	SRA	Alta de Auditoría

	Verifica que el repositorio creado respete la estructura enunciada en el presente documento (ver CU02).		
	Registra en el SRA la auditoría tipo AUD003		

Postcondiciones

Código	Descripción
POS1	Alta de auditoría

Flujo Secundario No - Conformidad

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	Si no encuentra evidencia de la creación del repositorio de Proyecto, envía un e-mail al área de infraestructura (INF) informando la falta del repositorio	SRA	Alta No-Conformidad
	Carga una no conformidad con el siguiente aspecto: Alta de Repositorio.		

Postcondiciones

Código	Descripción
POS1	Alta No-Conformidad

CU04. Estimando una Petición.

Breve descripción.

Permite identificar el momento y las actividades correspondientes a la estimación de una Petición.

Actores intervinientes

Código	Nombre
JP	Jefe de Proyecto

SGP	Sistema de Gestión de Proyectos
SCM	System Configuration Management

Precondiciones

Código	Descripción
PRE1	Debe existir una petición aceptada y pendiente de estimación por parte de un JP

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
JP	Se crea / modifica el documento de estimación. [IdRequest]_Estimacion.doc	SCM	Alta de Documento en Repositorio
	Identifica el origen de la petición.		
	Guarda una copia del documento dentro del directorio raíz del repositorio.		
JP	Cierra la etapa de estimación en el SGP.	SGP	Pendiente de aceptación de Estimación.

Postcondiciones

Código	Descripción
POS1	Pendiente de Aceptación de Estimación

CU05. Documentando Requerimientos del Proyecto.

Breve descripción.

Permite documentar los requerimientos correspondientes a la petición.

Actores intervinientes

Código	Nombre
JP	Jefe de Proyecto
SCM	System Configuration Management

Precondiciones

Código	Descripción
PRE1	Debe existir una petición aceptada por parte de un JP

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción

JP	Documenta los requerimientos correspondientes a la petición.	SCM	Alta de Documento en Repositorio
	Genera una nueva versión del documento [IdRequest]_Requerimientos_de_Proyecto.doc		
	Guarda una copia del documento dentro del directorio raíz del repositorio.		

Postcondiciones

Código	Descripción
POS1	Alta de Documento de requerimientos del Proyecto en Repositorio

CU06. Auditando Estimación.

Breve descripción.

Permite auditar la existencia del documento de estimación identificando que el mismo fue salvado en el repositorio creado a tal fin.

Actores intervinientes

Código	Nombre
QAO	Oficina de Calidad
SRA	Sistema de Registro de Auditorías

Precondiciones

Código	Descripción
PRE1	Debe existir el cierre de la etapa de estimación en el SGP

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	Después de notificado del cierre de etapa de estimación, revisa se haya creado el documento [IdRequest]_Estimación.doc	SRA	Alta de auditoría

	Revisa que el mismo se encuentre en el repositorio creado a tal fin.		
	Registra en el SRA la auditoría tipo AUD006		

Postcondiciones

Código	Descripción
POS1	Alta de auditoría

Flujo Secundario No - Conformidad

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	Si no existe el documento, envía un e-mail al Jefe de Proyecto informando la falta del documento. Carga una no conformidad con el siguiente aspecto: No evidencia Doc. Estimación.	SRA	Alta No-Conformidad
	Si el documento se encuentra vacío, sin datos. Carga una no conformidad con el siguiente aspecto: Documento vacío.		
	Si no se completó el nombre con el [Id_de_Request] correspondiente al Proyecto carga una no conformidad con el siguiente aspecto: Documento con nombre erróneo.		
	Si no se completó el apartado Control de cambios con versión del documento, cambios realizados y actor que lo realizó, carga una no conformidad con el siguiente aspecto: Ctrl. de cambios incompleto.		
	Si no fueron identificadas las horas por fase carga una no conformidad con el siguiente aspecto: Horas no estimadas por fases.		
	Verifica que los apartados que identifican acciones sobre checkbox sean activados: sino se cumple, carga una no conformidad con el siguiente aspecto: Documento incompleto.		

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
	Si se identifica acción de aplica SI por medio de un checkbox, verifica que se complete la descripción correspondiente, sino se cumple, carga una no conformidad con el siguiente aspecto: No completa descripción en apartados.		
	Si no se eliminaron instrucciones de cómo completar el documento carga una no conformidad con el siguiente aspecto: Documento informal.		
	Si no encuentra completo algún apartado para detallar la estimación carga una no conformidad con el siguiente aspecto: Documento sin detalle de estimación.		

Postcondiciones

Código	Descripción
POS1	Alta No-Conformidad

CU07. Auditando Requerimientos del Proyecto.

Breve descripción.

Permite auditar la existencia del documento Requerimiento del proyecto identificando que el mismo fue salvado en el repositorio creado a tal fin.

Actores intervinientes

Código	Nombre
QAO	Oficina de Calidad
SRA	Sistema de Registro de Auditorías

Precondiciones

Código	Descripción
PRE1	Debe el cierre de la etapa de estimación en el SGP

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	Después de notificado del cierre de etapa de estimación en el SGP, revisa se haya creado el documento [IdRequest]_Requerimiento_de_Proyecto.doc	SRA	Alta de auditoría

	Revisa que el mismo se encuentre en el repositorio creado a tal fin.		
	Registra en el SRA la auditoría tipo AUD007		

Postcondiciones

Código	Descripción
POS1	Alta de auditoría

Flujo Secundario No - Conformidad

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	Si no existe el documento, envía un e-mail al Jefe de Proyecto informando la falta del documento. Carga una no conformidad con el siguiente aspecto: No evidencia RQ Proyecto.	SRA	Alta No-Conformidad
	Si el documento se encuentra vacío, sin datos. Carga una no conformidad con el siguiente aspecto: Documento vacío.		
	Si no se completó el nombre con el [Id_de_Request] correspondiente al Proyecto carga una no conformidad con el siguiente aspecto: Documento con nombre erróneo.		
	Si no se completó el apartado Control de cambios con versión del documento, cambios realizados y actor que lo realizó, carga una no conformidad con el siguiente aspecto: Ctrl. De cambios incompleto.		
	Si no fue escrita una breve descripción de objetivos a alcanzar en el documento carga una no conformidad con el siguiente aspecto: Documento sin objetivo.		
	Verifica que los apartados que identifican acciones sobre checkbox sean activados: sino se cumple, carga una no conformidad con el siguiente aspecto: Documento incompleto		

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
	Si se identifica acción de aplica SI por medio de un checkbox, verifica que se complete la descripción correspondiente, sino se cumple, carga una no conformidad con el siguiente aspecto: No completa descripción en apartados.		
	Si no se eliminaron instrucciones de cómo completar el documento carga una no conformidad con el siguiente aspecto: Documento informal.		

Postcondiciones

Código	Descripción
POS1	Alta No-Conformidad

CU08. Planificando un Proyecto.

Breve descripción.

Permite identificar el momento y las actividades correspondientes a la planificación de un proyecto.

Actores intervinientes

Código	Nombre
JP	Jefe de Proyecto
SCM	System Configuration Management
SGP	Sistema de Gestión de Proyectos

Precondiciones

Código	Descripción
PRE1	Debe existir una estimación aceptada y la etapa de estimación cerrada en el SGP.

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
JP	Crea / modifica el documento [IdRequest]_Plan_de_Proyecto.doc	SCM	Alta Plan de Proyecto
	Guarda una copia del documento de Plan de Proyecto dentro en el directorio raíz del repositorio creado para tal fin		
	cierra la etapa de planificación en el sistema de gestión y el mismo informa de manera automática a la PMO y a la QAO que la etapa de planificación ha sido cerrada	SGP	Planificación en estado cerrada

Postcondiciones

Código	Descripción
POS1	Alta del Documento Plan de Proyecto.
POS2	Planificación Cerrada en SGP

CU09. Auditando Planificación.

Breve descripción.

Permite auditar la existencia del documento de plan de proyecto identificando que el mismo fue salvado en el repositorio creado a tal fin.

Actores intervinientes

Código	Nombre
QAO	Oficina de Calidad
SRA	Sistema de Registro de Auditorías

Precondiciones

Código	Descripción
PRE1	Debe el cierre de la etapa de Planificación en el SGP

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	Después de notificado del cierre de etapa de planificación en el SGP, revisa se haya creado el documento [IdRequest]_Plan_de_Proyecto.doc	SRA	Alta de auditoría
	Revisa que el mismo se encuentre en el repositorio creado a tal fin.		
	Registra en el SRA la auditoría tipo AUD009.		
	Genera un TAG al proyecto para guardar el estado a ese momento. el mismo se denomina de la siguiente manera: PRO_NRO_PP. Donde el NRO representa al ID de Request que identifica el proyecto, y PP representa el momento de tag post planificación.	SCM	<p>1 – Si no existe no conformidad para la auditoría de referencia realiza un TAG al Proyecto.</p> <p>2 – Si existe no conformidad abierta para la auditoría de referencia y la misma no escaló a staff, se espera cerrar la no conformidad y se genera un TAG al proyecto.</p> <p>3 – Si existe no conformidad abierta para la auditoría de referencia y la misma escaló a staff realiza un TAG al Proyecto.</p>

Postcondiciones

Código	Descripción
POS1	Alta de auditoría

Flujo Secundario No - Conformidad

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	Si no existe el documento, envía un e-mail al Jefe de Proyecto informando la falta del documento. Carga una no conformidad con el siguiente aspecto: No evidencia Doc Plan de Proyecto.	SRA	Alta No-Conformidad
	Si el documento se encuentra vacío, sin datos. Carga una no conformidad con el siguiente aspecto: Documento vacío.		
	Si no se completó el nombre con el [Id_de_Request] correspondiente al Proyecto carga una no conformidad con el siguiente aspecto: Documento con nombre erróneo.		
	Si no se completó el apartado Control de cambios con versión del documento, cambios realizados y actor que lo realizó, carga una no conformidad con el siguiente aspecto: Ctrl. De cambios incompleto.		
	Si no fue escrita una breve descripción del alcance en el documento carga una no conformidad con el siguiente aspecto: Documento sin alcance.		
	Verifica que los apartados que identifican acciones sobre checkbox sean activados: sino se cumple, carga una no conformidad con el siguiente aspecto: Documento incompleto		
	Si se identifica acción de aplica SI por medio de un checkbox, verifica que se complete la descripción correspondiente, sino se cumple, carga una no conformidad con el siguiente aspecto: No completa descripción en apartados.		
	Si no se eliminaron instrucciones de cómo completar el documento carga una no conformidad con el siguiente aspecto: Documento informal.		

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
	Si no se identifica el equipo de proyecto (roles y personas). Carga una no conformidad con el siguiente aspecto: No evidencia equipo de proyecto		
	Si no se existe el link al cronograma de proyecto con recursos y esfuerzos del SGP. Carga una no conformidad con el siguiente aspecto: No evidencia link al cronograma.		

Postcondiciones

Código	Descripción
POS1	Alta No-Conformidad

CU10. Gestionando No Conformidad.

Breve descripción.

Permite conocer de que manera y en que periodos se deben escalar y gestionar las no conformidades resultantes de las auditorías realizadas.

Actores intervinientes

Código	Nombre
QAO	Oficina de Calidad
SRA	Sistema de Registro de Auditorías

Precondiciones

Código	Descripción
PRE1	Debe existir una no-conformidad en estado abierta en el SRA.

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	La QAO a 96 horas posterior a registrar una no conformidad revisa el estado de la misma	SRA	Alta de Escala de No-Conformidad
	Si no se encuentra evidencia de corrección de la no conformidad se procede a escalar		
	Se envía un e-mail al responsable asignado copiando a un nivel de jerarquía superior informando la falta de solución de la no conformidad.		

	Se registra la acción de escalar en el SRA.		
--	---	--	--

Postcondiciones

Código	Descripción
POS1	Alta de escala de no conformidad

Flujo Alternativo

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	La QAO a 96 horas posterior a registrar una no conformidad revisa el estado de la misma	SRA	No-Conformidad Cerrada.
	Si encuentra evidencia de corrección de la no conformidad se procede a cerrar la misma		

Información adicional Niveles de Escala

Orden de Escala	Descripción
1	Jefe de Proyecto
2	Gerente
3	Staff

Postcondiciones

Código	Descripción
POS1	Cierre de No – Conformidad

CU11. Realizando Seguimiento a un Proyecto.

Breve descripción.

Permite seguir el ciclo de vida del proyecto identificando oportunidades de mejora en la gestión.

Actores intervinientes

Código	Nombre
PMO	Oficina de Proyectos
SCM	System Configuration Management

Precondiciones

Código	Descripción
PRE1	Debe existir una petición / proyecto en gestión.

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
PMO	Identifica el Origen de la petición / Proyecto	SCM	Alta de Informe de Seguimiento
	Origen distinto a SGU		
	Genera un informe semanal con información a gestión de la petición / Proyecto.		
	Publica el Informe http://bahq55:8090/wiki/jsp/Wiki?PMO-TmAs		
	Informa de su publicación vía e-mail a los Jefes de Proyectos y / o Gerentes con copia a la QAO.		

Flujo Alternativo

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
PMO	Identifica el Origen de Proyecto	SCM	Alta de Informe de Seguimiento
	Proyectos SGU		
	Genera un informe quincenal con información de gestión de la petición / Proyecto.		
	Publica el Informe http://bahq55:8090/wiki/jsp/Wiki?PMO-TmAs		
	Informa de su publicación vía e-mail a los Jefes de Proyectos y / o Gerentes con copia a la QAO.		

Información adicional No - generación de Informes

En el caso excepcional de no generarse los informes anteriormente expuestos la PMO informará el motivo de la no generación de los informes de seguimiento, pudiendo registrar la QAO las auditorías correspondientes con la leyenda de la excepción.

Postcondiciones

Código	Descripción
POS1	Alta Informe de Seguimiento

CU12. Auditando Seguimiento de un Proyecto.

Breve descripción.

Permite conocer las acciones de auditoría del seguimiento de un proyecto.

Actores intervinientes

Código	Nombre
QAO	Oficina de Calidad
SRA	Sistema de Registro de Auditorías

Precondiciones

Código	Descripción
PRE1	Debe existir una petición / proyecto en gestión.

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	Revisa que los Proyectos evidencien seguimientos según su origen	SRA	Alta de auditoría
	Verificar que se hayan generado los informes en tiempo y forma		
	Registra en el SRA la auditoría tipo AUD012		
	Revisa que los informes representativos del seguimiento contengan referencia a las peticiones / Proyectos en gestión.		

Postcondiciones

Código	Descripción
POS1	Alta de auditoría

Flujo Secundario No - Conformidad

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	Si no evidencia seguimiento carga una no conformidad con el siguiente aspecto: Falta Documentación de Requerimientos.	SRA	Alta No-Conformidad

	Si no encuentra referencia a proyectos en ejecución carga una no conformidad con el aspecto: Falta Referencia a Proyecto en gestión		
--	---	--	--

Postcondiciones

Código	Descripción
POS1	Alta No-Conformidad

Información adicional Auditorías de Informes

SGU – Quincenal los Miércoles.

Distinto de SGU – Semanal los lunes

CU13. Auditando Auditorías.

Breve descripción.

Permite realizar el control a las Auditorías en curso.

Actores intervinientes

Código	Nombre
QAO	Oficina de Calidad
SRA	Sistema de Registro de Auditorías

Precondiciones

Código	Descripción
PRE1	Debe existir un proyecto en etapa de ejecución.

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	Entre los días 1 y 5, y los días 15 y 20 de cada mes una persona asignada por la QAO debe ingresar a revisar las Auditorías correspondientes a los Proyectos en curso.	SRA	Alta de auditoría
	Ingresa en cada proyecto ingresado en el SRA y analiza la gestión realizada.		

	Identifica que la cantidad de auditorías corresponda al tiempo de vida de los Proyectos en curso.		
	Esta actividad se registra en la base de auditoría, como tipo de auditoría AUD013.		
	Identifica el estado de gestión de las no conformidades.		
	Esta actividad se registra en la base de auditoría, como tipo de auditoría AUD013.		

Postcondiciones

Código	Descripción
POS1	Alta de auditoría

Flujo Secundario No - Conformidad

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	Si no identifica la cantidad de auditorías correspondientes carga una no conformidad con el siguiente Aspecto: No evidencia gestión de Auditoría.	SRA	Alta No-Conformidad
	Si no identifica gestión sobre no conformidades carga una no conformidad con el siguiente aspecto: No evidencia gestión de no conformidad		

Postcondiciones

Código	Descripción
POS1	Alta No-Conformidad

CU14. Liberando entregables.

Breve descripción.

Identifica las acciones a realizar para llevar a cabo la entrega de artefactos según compromiso asumido.

Actores intervinientes

Código	Nombre
JP	Jefe de Proyecto
INF	Infraestructura

Precondiciones

Código	Descripción
PRE1	Debe existir un documento plan de proyecto que identifique la fecha y artefactos a entregar.

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
JP	Identifica en el documento plan de proyecto los artefactos a liberar y la fecha de liberación de dichos artefactos.	SCM	Liberación de Artefactos.
	Ingresa en el directorio salida los artefactos a liberar cumpliendo con la fecha comprometida en el documento plan de proyecto.		

Postcondiciones

Código	Descripción
POS1	Liberación de Artefactos

CU15. Auditando Liberaciones.

Breve descripción.

Permite realizar el control de los artefactos comprometidos a entregar según documento plan de proyecto.

Actores intervinientes

Código	Nombre
QAO	Oficina de Calidad
SRA	Sistema de Registro de Auditorías

Precondiciones

Código	Descripción
PRE1	Debe existir un documento plan de proyecto que identifique la fecha y artefactos a entregar.

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	Ingresa al repositorio del proyecto y verifica que en el directorio salida existan los artefactos comprometidos a entregar según documento plan de proyecto.	SRA	Alta de auditoría
	A esta actividad la registra en el sistema de auditorías como AUD015 Auditando liberación		

Postcondiciones

Código	Descripción
POS1	Alta de auditoría

Flujo Secundario No - Conformidad

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	Si no identifica la existencia de los artefactos a liberar comprometidos en el documento plan de proyecto, carga una no conformidad con el siguiente aspecto: No evidencia liberación de artefactos.	SRA	Alta No-Conformidad

Postcondiciones

Código	Descripción
POS1	Alta No-Conformidad

CU16. Realizando Cambio de Alcance.**Breve descripción.**

Identifica las acciones a realizar cuando es necesario realizar un cambio de alcance.

Actores intervinientes

Código	Nombre
JP	Jefe de Proyecto
INF	Infraestructura
QAO	Oficina de Calidad

Precondiciones

Código	Descripción
PRE1	Debe existir un consenso para la realización de un cambio de alcance.

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	Queda notificada de la existencia de un cambio de alcance por medio de un e-mail generado por el SGP.	SCM	TAG generado al proyecto para cambios de Alcance.
	Genera el TAG denominando el mismo de la siguiente manera: PRO_NRO_SC_NROsc. Donde el NRO representa al ID de Request que identifica el proyecto, y NROsc representa al ID de Request que identifica al cambio de alcance.		
JP	Evoluciona los artefactos Requerimientos del proyecto, Estimación y plan de proyecto con información correspondiente al cambio de alcance.	SCM	Documentación actualizada con información del cambio de alcance.

Postcondiciones

Código	Descripción
POS1	TAG generado al proyecto.
POS2	Evolución de documentos del proyecto.

CU17. Auditando Cambio de Alcance.

Breve descripción.

Permite realizar el control cuando se realiza un cambio de alcance al proyecto.

Actores intervinientes

Código	Nombre
QAO	Oficina de Calidad
SRA	Sistema de Registro de Auditorías

Precondiciones

Código	Descripción
PRE1	Debe existir un proyecto cerrado.

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	Recibe un e-mail correspondiente a un cambio de alcance por el SGP. Identifica a que proyecto corresponde el cambio de alcance.	SRA	Alta de auditoría
	Verifica que se haya creado el repositorio con el formato PRO_NRO_SC_NROsc.		
	Verifica la existencia de los documentos Estimación, Plan de Proyecto y Requerimientos del Proyecto correspondientes al cambio de alcance dentro del repositorio PRO_NRO_SC_NROsc.		
	Verifica que se haya contemplado el cambio de alcance en el apartado control de cambios de los documentos estimación, plan y requerimientos del repositorio origen del proyecto.		
	Registra la auditoría, la identifica como AUD017 Cambio de alcance.		

Postcondiciones

Código	Descripción
POS1	Alta de auditoría

Flujo Secundario No - Conformidad

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	Si no identifica la existencia de los artefactos y el repositorio generado para el cambio de alcance, carga una no conformidad con el siguiente aspecto: No evidencia cambio de alcance.	SRA	Alta No-Conformidad
	Si no se completó el apartado control de cambios con versión del documento, cambios realizados, actor que lo realizó y el N° RQ Scope Change carga una no conformidad con el siguiente aspecto: Ctrl. De Cambios incompleto.		

Postcondiciones

Código	Descripción
POS1	Alta No-Conformidad

CU18. Cerrando Proyecto.

Breve descripción.

Identifica el fin de la actividades del Proyecto.

Actores intervinientes

Código	Nombre
JP	Jefe de Proyecto

Precondiciones

Código	Descripción
PRE1	Debe existir un proyecto en gestión

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
JP	Ingresa al sistema de gestión de proyecto e realiza la acción de cerrar proyecto.	SGP	Proyecto en estado cerrado.

Postcondiciones

Código	Descripción
POS1	Proyecto Cerrado.

CU19. Auditando Cierre de Proyecto.

Breve descripción.

Permite realizar el control cuando se realiza el cierre de proyecto.

Actores intervinientes

Código	Nombre
QAO	Oficina de Calidad
SRA	Sistema de Registro de Auditorías

Precondiciones

Código	Descripción
PRE1	Debe existir un proyecto cerrado.

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción

QAO	Recibe un e-mail de cierre de proyecto generado por el SGP.	SRA	Alta de auditoría
	Identifica en el plan de proyecto los compromisos correspondiente a artefactos entregables y verifica que los mismos se encuentren en el directorio salida, según documento plan de proyecto.		
	A esta auditoría se la identifica como AUD019 Cierre de Proyecto.		
	Realiza un TAG al final de proyecto	SCM	TAG al Final de Proyecto

Postcondiciones

Código	Descripción
POS1	Alta de auditoría

Flujo Secundario No - Conformidad

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	Si no identifica la existencia de los artefactos a liberar comprometidos en el documento plan de proyecto, carga una no conformidad con el siguiente aspecto: No evidencia liberación en cierre de proyecto.	SRA	Alta No-Conformidad

Postcondiciones

Código	Descripción
POS1	Alta No-Conformidad

CU20. Generando Métricas.

Breve descripción.

Permite conocer el conjunto de Métricas definidas para la utilización de la corporación.

Actores intervinientes

Código	Nombre
PMO	Oficina de Proyectos
TEST	Área de Testing
SOP	Área de Soporte

Precondiciones

Código	Descripción
PRE1	Debe existir información de gestión.

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
TEST	Periódicamente la oficina de Testing genera dos reportes correspondiente a su gestión y los envía al área de administración de proyecto.	PMO	Consolida la información recibida. Genera informe mensual de Gestión.
SOP	Periódicamente el área de Soporte genera dos reportes correspondiente a su gestión y los envía al área de administración de proyecto.		
PMO	Periódicamente se generan dos reportes correspondiente a la gestión de desarrollo, ellos son enviados al área de administración de proyecto y como parte del seguimiento de proyectos genera reportes de gestión correspondiente al esfuerzo y la entrega comprometida.		

Postcondiciones

Código	Descripción
POS1	Alta de Reportes

CU21. Publicando Métricas.

Breve descripción.

Permite conocer el cronograma de publicación de Métricas definidas para la utilización de la corporación.

Actores intervinientes

Código	Nombre
PMO	Oficina de Proyectos

Precondiciones

Código	Descripción
PRE1	Debe existir métricas generadas.

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
PMO	Periódicamente la oficina de administración de Proyectos recibe los reportes generados desde las áreas involucradas a la gestión de Métricas.	SCM	Alta de Publicación de informe mensual.
	La oficina de administración de proyectos consolida los reportes recibidos.		
	La oficina de administración de proyectos publica un informe mensual elaborando indicadores de gestión con la información recibida. Fecha de publicación: 15 de cada mes. Forma de publicación: e-mail a gerentes y jefes, copia a la QAO. Lugar de publicación: http://bahq55:8090/wiki/jsp/Wiki?PMO-TmAs		

Postcondiciones

Código	Descripción
POS1	Alta de publicación

CU22. Auditando publicación de Métricas.

Breve descripción.

Permite validar si las métricas se encuentran publicadas en tiempo y forma.

Actores intervinientes

Código	Nombre
QAO	Oficina de Calidad
SRA	Sistema de registración de Auditorías

Precondiciones

Código	Descripción
PRE1	Deben existir métricas publicadas

Flujo principal

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	La fecha 16 de cada mes ingresa a verificar la existencia del informe mensual.	SRA	Alta de Auditoría de Métricas
	Carga un registro en el SRA identificando la auditoría.		
	A esta auditoría se la identifica como AUD022 publicación de Métricas.		

Postcondiciones

Código	Descripción
POS1	Alta auditoría.

Flujo Secundario No - Conformidad

Acción Actor		Respuesta del Sistema	
Código	Descripción	Código	Descripción
QAO	Si no identifica la existencia del informe mensual, carga una no conformidad con el siguiente aspecto: No identifica informe mensual.	SRA	Alta No-Conformidad

Postcondiciones

Código	Descripción
POS1	Alta No-Conformidad

BIBLIOGRAFÍA

- [ARG, CRISIS] Información en wikipedia sobre la crisis en Argentina 2001 <http://es.wikipedia.org/wiki/Argentinazo>
- [ARLOW, 2005] UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design (2nd Edition) (The Addison-Wesley Object Technology Series) - Jim Arlow, Ila Neustadt (Jul 7, 2005)
- [BAS, 2002] Empirical Software Engineering Issues. Critical Assessment and Future Directions: International Workshop, Dagstuhl Castle, Germany, June 26-30, 2006, Revised Papers (Lecture Notes in Computer Science) by Victor Basili, Dieter Rombach, Kurt Schneider, and Barbara Kitchenham (May 3, 2007)
- [BRAMBLE, 2001] Bramble Adolph, Patterns for Effective Use Cases, 2001.
- [CMMI, 1.2] CMMI® for Development, Version 1.2 Sitio oficial del sei <http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr008.pdf>
- [CMMI, APPR] Appraisal Requirements for CMMI (ARC). Software Engineering Institute, <http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr011.pdf>
- [CMMI, RESULTADOS] Publicación de los resultados de las evaluaciones CMMI - http://sas.sei.cmu.edu/pars/pars_detail.aspx?a=12207
- [CMMI, SAS] <http://sas.sei.cmu.edu/pars/pars.aspx> Software Engineerig Institute, Published Appraisal Results
- [CMMI-SW, 2002] CMMI(R): Guidelines for Process Integration and Product Improvement (2nd Edition) (The SEI Series in Software Engineering) - Mary Beth Chrissis, Mike Konrad, Sandy Shrum
- [CMU] Sitio oficial de la universidad de Carnegie Mellon. <http://www.cmu.edu>
- [COCKBURN, 2000] Cockburn Alistair, Writing Effective Use Cases., 2000, Addison-Wesley.
- [FOWLER, UML] UML Distilled: A Brief Guide to the Standard Object Modeling Language.
- [IEEE] Sitio oficial de la IEEE <http://www.ieee.org>
- [IEEE, 1990] IEEE Standard Glossary of Software Engineering Terminology, IEEE Std 610.12-1990, 1990.
- [NOTICIAS, SOFTWARE] <http://tecnologia.infobaeprofesional.com/notas/69660-El-software-de-gestion-crecera-17-este-ano-en-la-Argentina.html&cookie>
- [OSTERWEIL, 1987] Software Processes Are Software Too - 1987 ACM 0270-5257/87/0300/0002500.75 - Leon Osterweil
- [PAULK, CMM1995] Paulk, M., et. al., The Capability Maturity Model: Guidelines for Improving the Software Process, Addison-Wesley, 1995.
- [PHILLIPS, 2004] Mike Phillips, CMMI Program Manager. CMMI V1.1 and Appraisal Tutorial, 2004, Carnegie Mellon University.
- [PMI] Página oficial de Project Management Institute. <http://www.pmi.org/>
- [PRESSMAN, SWE] Software Engineering: A Practitioner's Approach - Roger S Pressman

- [SCAMPI, 2005] CMMI(R) SCAMPI Distilled: Appraisals for Process Improvement (The SEI Series in Software Engineering) by Dennis M. Ahern, Jim Armstrong, Aaron Clouse, and Jack R. Ferguson (Mar 12, 2005)
- [SCAMPI] <http://www.sei.cmu.edu/pub/documents/01.reports/pdf/01hb001.pdf>
Standard CMMISM Appraisal Method for Process Improvement (SCAMPISM), Version 1.1: Method Definition Document
- [SCHUL, 1992] Schulmeyer, G. G. and McManus, J. I., eds. Handbook of Software Quality Assurance, 2nd ed, Van Nostrand Reinhold, 1992.
- [SCHWABER, SCRUM] SCRUM Development Process
<http://jeffsutherland.com/oopsla/schwapub.pdf>
- [SCOTT, 2002] Scott, Kendall (2002). The Unified Process Explained.
- [SEDORF, 1997] Software Engineering. Dorfman and R. H. Thayer, editors. IEEE Computer Society Press, 1997
- [SEI, 2009] Sitio oficial del SEI, <http://www.sei.cmu.edu>, página vigente al 01/01/2009.
- [SOMM, 2002] Ingeniería de Software. Ian Sommerville Addison Wesley 2002.
- [SOMMER, 1998] Requirements Engineering: Processes and Techniques (Worldwide Series in Computer Science). Gerald Kotonya and Ian Sommerville.
- [THE STANDISH GROUP, 1995]. Página principal de la empresa. Disponible en <http://www.standishgroup.com>, página vigente al 01/01/2004. Una buena parte de este informe puede verse en <http://net.educause.edu/ir/library/pdf/NCP08083B.pdf>
- [UML&PATTERNS, 2004] Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition) - Craig Larman
- [UML, OMG] Página del Object Management Group, UML <http://www.uml.org/>

ÍNDICE

- Actores, 52
- Appraisal Wizard, 14
- Capítulos, 3
- Certificación, vi
- CM, 9
- CMMI, vi, 8
- CMM-Quest, 14
- Diagrama de casos de uso, 23, 58
- Dominio, 19, 21, 53
- entregables, 10
- GFORGE, 15
- Historia CMM, 5
- IME Toolkit, 14
- ITG, 15
- ITG IT Governance, 10
- Lead Appraiser, vi
- MA, 9
- MET, 17
- MET Casos de uso, 23, 57
- MET Requerimientos, 19
- Metodología, vi
- Microsoft Exchange, 11
- Microsoft Project, 15
- Modelo, vi
- Niveles CMM, 6
- OSTERWEIL, ii, 2, 88
- PMAM, 38, 40
- PMC, 9
- PMO, vii, 27
- PP, 9
- PPQA, 3, 9
- Proceso de desarrollo de software, vi
- QAO, 46, 59
- Quality Assurance, vi
- Quality Control, vi
- Relación con otros Productos, 19, 49
- Relevancia y originalidad, 11
- Remedy Action Request, 15
- REQM, 9
- SAM, 9
- SCAMPI, 1, 9, 42
- SEI, 8
- SRA, 29
- SRA Pantallas, 33
- SVN, 11
- TRAC, 11, 16
- Vista Sumaria, 19, 51