



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



*Máster Universitario en Ingeniería del Software, Métodos
Formales y Sistemas de Información*

Universitat Politècnica de València

**Departamento de Sistemas Informáticos y
Computación**

***CMMI aplicado a entornos de
desarrollo software: El caso de
MOSKitt4ME***

Valencia, Septiembre 2013

Author:

Díaz Aspilcueta, Edith

Director:

Vicente, Pelechano Ferragud

Co-directora:

Manuela, Albert Albiol

Curso Académico 2012/2013

"Dime y lo olvido, enséñame y lo recuerdo, involúcrame y lo aprendo"

Benjamin Franklin

Resumen

El estudio está centrado principalmente en la calidad del proceso de desarrollo de software. En concreto, el trabajo busca la integración del estándar CMMI[®] – un modelo para la mejora y evaluación de métodos para el desarrollo y ejecución de sistemas de software – y MOSKitt4ME – una herramienta para la definición de métodos de desarrollo de software y la construcción de las herramientas CASE que les dan soporte.

Para conseguir el objetivo descrito, se ha aplicado a MOSKitt4ME las mejores prácticas de calidad descritas en CMMI[®] para desarrollo v1.3, hasta alcanzar los niveles de madurez 2 y 3, con el fin de obtener entornos de desarrollo de software donde se apliquen métodos o procesos que están bien caracterizados y comprendidos. Este propósito se realizó por medio de la evaluación, análisis y determinación de las relaciones y divergencias que existían entre MOSKitt4ME y CMMI[®] para desarrollo v1.3 en los niveles de madurez 2 y 3.

Para finalizar, una vez identificadas las relaciones o diferencias que existen entre MOSKitt4ME y CMMI[®] para desarrollo v1.3, se realizaron propuestas para extender MOSKitt4ME y de esta forma alcanzar el nivel de calidad descrito en CMMI[®] para desarrollo v1.3 y lograr una mayor sinergia entre ambos.

Abstract

The main focus of present study is the quality of software development processes. In particular, it attempts to integrate the standard CMMI[®] – a model for the improvement and evaluation of methods for the development and enactment of software systems – with MOSKitt4ME – a tool for definition of software development methods and building CASE tools that support them.

To obtain the objective described above, the best quality practices that are described in CMMI[®] for Development v1.3 were applied in MOSKitt4ME until maturity levels 2 and 3 had been reached, in order to obtain software development environments where methods or processes are applied that are well characterized and understood. This aim has been achieved through evaluation, analysis and identification of the existing connections and differences between MOSKitt4ME and CMMI[®] for Development v1.3 maturity levels 2 and 3.

Finally, once the connections or differences between MOSKitt4ME and CMMI[®] for Development v1.3 had been identified, proposals were made to extend MOSKitt4ME in order to the quality level described in CMMI[®] for Development v1.3 and thus accomplish greater synergy between them.

Agradecimientos

A Dios por darme la fuerza para perseverar y alcanzar una de mis metas.

A todos los profesores del máster y compañeros de estudios – de ellos he aprendido tanto e incluso más de las expectativas que me había formulado, antes de llegar a este hermoso país. Es admirable la vocación y profesionalismo de muchos de los docentes que imparten clases.

A Mario Cervera por ser un excelente guía, por compartir sus conocimientos, su tiempo y mucha paciencia – de no contar con su apoyo no habría logrado finalizar este proyecto. A Manuela Albert y Victoria Torres quienes desde el inicio fueron mis guías.

A Markus Hödl (∇) mi esposo, por toda su ayuda incondicional, su disciplina, por toda su comprensión, sus buenos consejos que siempre me acompañan y me guían, sobre todo por su amor que ha sido mi principal pilar ante los buenos y malos momentos.

A mis buenos amigos que he encontrado en este período de estancia, imposible mencionarlos a todos porque llenaría muchas páginas...Sin embargo, de quien he aprendido mucho, indiscutiblemente, es de Angel Cuenca, gracias a todo su apoyo y amistad a lo largo del máster. A Priscila Cedillo que junto con su familia me ha brindado una amistad incondicional.

A mis padres, familiares y buenos amigos que siempre me apoyan a pesar de la distancia.

Edith Díaz

Valencia, Septiembre 2013

Índice general

| | |
|---|------------|
| Resumen | II |
| Abstract | III |
| Agradecimientos | IV |
| Índice de Figuras | XI |
| Índice de Tablas | XII |
| | |
| 1. Introducción | 1 |
| 1.1. Motivación | 2 |
| 1.2. Objetivos | 4 |
| 1.3. Contexto de la tesis | 5 |
| 1.4. Estructura del documento | 5 |
| | |
| 2. Contexto Tecnológico | 7 |
| 2.1. Integración de Modelos de Madurez de Capacidades | 7 |
| 2.1.1. Acerca de CMMI [®] para desarrollo | 8 |
| 2.1.1.1. Niveles de madurez | 8 |
| 2.1.1.2. Niveles de capacidad | 10 |
| 2.1.2. Componentes del área de proceso | 10 |
| 2.1.3. SCAMPI | 13 |
| 2.2. Marco metodológico para construir métodos de desarrollo de software – MOSKitt4ME | 15 |
| 2.2.1. Ingeniería de métodos | 16 |
| 2.2.2. SPEM 2.0 | 18 |
| 2.2.2.1. Características generales de SPEM 2.0 | 19 |
| 2.2.2.2. Conceptos claves SPEM 2.0 | 20 |
| 2.2.2.3. Descripción detallada | 21 |
| 2.3. Conclusiones | 26 |
| | |
| 3. Comparación de CMMI Vs MOSKitt4ME - Nivel de Madurez 2 | 27 |
| 3.1. Gestión de acuerdos con proveedores (SAM) | 27 |
| 3.2. Gestión de requisitos (REQM) | 28 |

| | | |
|----------|---|----|
| 3.2.1. | SG 1 Gestionar los requisitos | 28 |
| 3.2.1.1. | SP 1.1 Comprender los requisitos | 28 |
| 3.2.1.2. | SP 1.2 Obtener el compromiso sobre los requisitos | 28 |
| 3.2.1.3. | SP 1.3 Gestionar los cambios a los requisitos | 29 |
| 3.2.1.4. | SP 1.4 Mantener la trazabilidad bidireccional de los requisitos | 29 |
| 3.2.1.5. | SP 1.5 Asegurar el alineamiento entre el trabajo del proyecto y los requisitos | 30 |
| 3.3. | Monitorización y control del proyecto (PMC) | 30 |
| 3.3.1. | SG 1 Monitorizar el proyecto frente al plan | 30 |
| 3.3.1.1. | SP 1.1 Monitorizar los parámetros de planificación del proyecto | 30 |
| 3.3.1.2. | SP 1.2 Monitorizar los compromisos | 31 |
| 3.3.1.3. | SP 1.3 Monitorizar los riesgos del proyecto | 31 |
| 3.3.1.4. | SP 1.4 Monitorizar la gestión de los datos | 31 |
| 3.3.1.5. | SP 1.5 Monitorizar la involucración de las partes interesadas | 32 |
| 3.3.1.6. | SP 1.6 Llevar a cabo las revisiones del progreso | 32 |
| 3.3.1.7. | SP 1.7 Llevar a cabo las revisiones de hitos | 32 |
| 3.3.2. | SG 2 Gestionar las acciones correctivas hasta su cierre | 33 |
| 3.4. | Planificación del proyecto (PP) | 33 |
| 3.4.1. | SG 1 Establecer las estimaciones | 33 |
| 3.4.1.1. | SP 1.1 Estimar el alcance del proyecto | 33 |
| 3.4.1.2. | SP 1.2 Establecer las estimaciones de los atributos de los productos de trabajo y de las tareas | 34 |
| 3.4.1.3. | SP 1.3 Definir las fases del ciclo de vida del proyecto | 35 |
| 3.4.1.4. | SP 1.4 Estimar el esfuerzo y el coste | 35 |
| 3.4.2. | SG 2 Desarrollar un plan de proyecto | 35 |
| 3.4.2.1. | SP 2.1 Establecer el presupuesto y el calendario | 36 |
| 3.4.2.2. | SP 2.2 Identificar los riesgos del proyecto | 36 |
| 3.4.2.3. | SP 2.3 Planificar la gestión de los datos | 36 |
| 3.4.2.4. | SP 2.4 Planificar los recursos del proyecto | 36 |
| 3.4.2.5. | SP 2.5 Planificar el conocimiento y las habilidades necesarias | 37 |
| 3.4.2.6. | SP 2.6 Planificar la involucración de las partes interesadas | 37 |
| 3.4.2.7. | SP 2.7 Establecer el plan de proyecto | 38 |
| 3.4.3. | SG 3 Obtener el compromiso con el plan | 38 |
| 3.5. | Aseguramiento de la calidad del proceso y del producto (PPQA) | 38 |
| 3.5.1. | SG 1 Evaluar objetivamente los procesos y los productos de trabajo | 39 |
| 3.5.1.1. | SP 1.1 Evaluar objetivamente los procesos | 39 |
| 3.5.1.2. | SP 1.2 Evaluar objetivamente los productos de trabajo | 39 |
| 3.5.2. | SG 2 Proporcionar una visión objetiva | 40 |
| 3.5.2.1. | SP 2.1 Comunicar y resolver las no conformidades | 40 |
| 3.5.2.2. | SP 2.2 Establecer los registros | 40 |
| 3.6. | Gestión de configuración (CM) | 40 |
| 3.6.1. | SG 1 Establecer las líneas base | 40 |
| 3.6.1.1. | SP 1.1 Identificar los elementos de configuración | 41 |
| 3.6.1.2. | SP 1.2 Establecer un sistema de gestión de configuración | 41 |

| | | |
|-----------|--|-----------|
| 3.6.1.3. | SP 1.3 Crear o liberar las líneas base | 41 |
| 3.6.2. | SG 2 Seguir y controlar los cambios | 42 |
| 3.6.2.1. | SP 2.1 Seguir las peticiones de cambio | 42 |
| 3.6.2.2. | SP 2.2 Controlar los elementos de configuración | 42 |
| 3.6.2.3. | SP 3.1 Establecer los registros de gestión de configuración | 43 |
| 3.6.2.4. | SP 3.2 Realizar auditorías de configuración | 43 |
| 3.7. | Medición y análisis (MA) | 43 |
| 3.8. | Resumen | 44 |
| 4. | Comparación de CMMI Vs MOSKitt4ME - Nivel de Madurez 3 | 47 |
| 4.1. | Definición de procesos de la organización (OPD) | 47 |
| 4.1.1. | SG 1 Establecer los activos de proceso de la organización | 47 |
| 4.1.1.1. | SP 1.1 Establecer los procesos estándar | 48 |
| 4.1.1.2. | SP 1.2 Establecer las descripciones de los modelos de ciclo de vida | 48 |
| 4.1.1.3. | SP 1.3 Establecer los criterios y las guías de adaptación | 48 |
| 4.1.1.4. | SP 1.4 Establecer el repositorio de mediciones de la organización | 49 |
| 4.1.1.5. | SP 1.5 Establecer la biblioteca de activos de proceso de la organización | 50 |
| 4.1.1.6. | SP 1.6 Establecer los estándares del entorno de trabajo | 50 |
| 4.1.1.7. | SP 1.7 Establecer las reglas y guías para los equipos | 50 |
| 4.2. | Enfoque en procesos de la organización (OPF) | 51 |
| 4.2.1. | SG 1 Determinar las oportunidades de mejora de procesos | 51 |
| 4.2.1.1. | SP 1.1 Establecer las necesidades de proceso de la organización | 51 |
| 4.2.1.2. | SP 1.2 Evaluar los procesos de la organización | 52 |
| 4.2.1.3. | SP 1.3 Identificar las mejoras de procesos de la organización | 52 |
| 4.2.1.4. | SG 2 Planificar e implementar las acciones de proceso | 52 |
| 4.2.2. | SG 3 Desplegar los activos de proceso de la organización e incorporar las experiencias | 53 |
| 4.2.2.1. | SP 3.1 Desplegar los activos de proceso de la organización | 53 |
| 4.2.2.2. | SP 3.2 Desplegar los procesos estándar | 53 |
| 4.2.2.3. | SP 3.3 Monitorizar la implementación | 53 |
| 4.2.2.4. | SP 3.4 Incorporar las experiencias en los activos de proceso de la organización | 54 |
| 4.3. | Formación en la organización (OT) | 54 |
| 4.4. | Gestión de riesgos (RSKM) | 55 |
| 4.5. | Gestión integrada del proyecto (IPM) | 55 |
| 4.5.1. | SG 1 Utilizar el proceso definido del proyecto | 56 |
| 4.5.1.1. | SP 1.1 Establecer el proceso definido del proyecto | 56 |
| 4.5.1.2. | SP 1.2 Utilizar los activos de proceso de la organización para planificar las actividades del proyecto | 56 |
| 4.5.1.3. | SP 1.3 Establecer el entorno de trabajo del proyecto | 57 |
| 4.5.1.4. | SP 1.4 Integrar los planes | 57 |
| 4.5.1.5. | SP 1.5 Gestionar el proyecto utilizando planes integrados | 57 |
| 4.5.1.6. | SP 1.6 Establecer los equipos | 58 |

| | | |
|-----------|---|----|
| 4.5.1.7. | SP 1.7 Contribuir a los activos de proceso de la organización | 58 |
| 4.5.2. | SG 2 Coordinar y colaborar con las partes interesadas relevantes | 59 |
| 4.5.2.1. | SP 2.1 Gestionar la involucración de las partes interesadas | 59 |
| 4.5.2.2. | SP 2.2 Gestionar las dependencias | 59 |
| 4.5.2.3. | SP 2.3 Resolver las cuestiones de coordinación | 59 |
| 4.6. | Desarrollo de requisitos (RD) | 60 |
| 4.6.1. | SG 1 Desarrollar los requisitos de cliente | 60 |
| 4.6.1.1. | SP 1.1 Educir las necesidades | 60 |
| 4.6.1.2. | SP 1.2 Trasformar las necesidades de las partes interesadas en requisitos de cliente | 60 |
| 4.6.2. | SG 2 Desarrollar los requisitos de producto | 60 |
| 4.6.3. | SG 3 Analizar y validar los requisitos | 61 |
| 4.6.3.1. | SP 3.1 Establecer los conceptos y los escenarios de operación | 61 |
| 4.6.3.2. | SP 3.2 Establecer una definición de la funcionalidad y de los atributos de calidad requeridos | 61 |
| 4.6.3.3. | SP 3.3 Analizar los requisitos | 62 |
| 4.6.3.4. | SP 3.4 Analizar los requisitos para conseguir un equilibrio | 62 |
| 4.6.3.5. | SP 3.5 Validar los requisitos | 62 |
| 4.7. | Integración del producto (PI) | 63 |
| 4.7.0.6. | SG 1 Prepararse para la integración del producto | 63 |
| 4.7.0.7. | SP 1.1 Establecer una estrategia de integración | 63 |
| 4.7.0.8. | SP 1.2 Establecer el entorno de integración del producto | 63 |
| 4.7.0.9. | SP 1.3 Establecer los procedimientos y los criterios de integración del producto | 64 |
| 4.7.0.10. | SG 2 Asegurar la compatibilidad de las interfaces | 64 |
| 4.7.0.11. | SP 2.1 Revisar la completitud de las descripciones de las interfaces | 64 |
| 4.7.0.12. | SP 2.2 Gestionar las interfaces | 65 |
| 4.7.1. | SG 3 Ensamblar los componentes de producto y entregar el producto | 65 |
| 4.7.1.1. | SP 3.1 Confirmar la disponibilidad de los componentes de producto para la integración | 66 |
| 4.7.1.2. | SP 3.2 Ensamblar los componentes de producto | 66 |
| 4.7.1.3. | SP 3.3 Evaluar los componentes de producto ensamblados | 66 |
| 4.7.1.4. | SP 3.4 Empaquetar y entregar el producto o componente de producto | 67 |
| 4.8. | Solución técnica (TS) | 67 |
| 4.8.1. | SG 1 Seleccionar soluciones de componentes de producto | 67 |
| 4.8.1.1. | SP 1.1 Desarrollar soluciones alternativas y los criterios de selección | 67 |
| 4.8.1.2. | SP 1.2 Seleccionar las soluciones de componentes de producto | 68 |
| 4.8.2. | SG 2 Desarrollar el diseño | 68 |
| 4.8.2.1. | SP 2.1 Diseñar el producto o los componentes de producto | 69 |
| 4.8.2.2. | SP 2.2 Establecer un paquete de datos técnicos | 69 |
| 4.8.2.3. | SP 2.3 Diseñar las interfaces usando criterios | 69 |

| | | |
|-----------|--|-----------|
| 4.8.2.4. | SP 2.4 Realizar los análisis sobre si hacer, comprar o re-utilizar | 70 |
| 4.8.3. | SG 3 Implementar el diseño del producto | 70 |
| 4.8.3.1. | SP 3.1 Implementar el diseño | 71 |
| 4.8.3.2. | SP 3.2 Desarrollar la documentación de soporte del producto | 71 |
| 4.9. | Validación (VAL) | 71 |
| 4.9.1. | SG 1 Preparar la validación | 72 |
| 4.9.1.1. | SP 1.1 Seleccionar los productos para la validación | 72 |
| 4.9.1.2. | SP 1.2 Establecer el entorno de validación | 72 |
| 4.9.1.3. | SP 1.3 Establecer los procedimientos y los criterios de validación | 72 |
| 4.9.2. | SG 2 Validar el producto o los componentes de producto | 73 |
| 4.9.2.1. | SP 2.1 Realizar la validación | 73 |
| 4.9.2.2. | SP 2.2 Analizar los resultados de la validación | 73 |
| 4.10. | Verificación (VER) | 74 |
| 4.10.1. | SG 1 Preparar la verificación | 74 |
| 4.10.1.1. | SP 1.1 Seleccionar los productos de trabajo para la verificación | 74 |
| 4.10.1.2. | SP 1.2 Establecer el entorno de verificación | 74 |
| 4.10.1.3. | SP 1.3 Establecer los procedimientos y los criterios de validación | 75 |
| 4.10.2. | SG 2 Realizar las revisiones entre pares | 75 |
| 4.10.3. | SG 3 Verificar los productos de trabajo seleccionados | 75 |
| 4.10.3.1. | SP 3.1 Realizar la verificación | 76 |
| 4.10.3.2. | SP 3.2 Analizar los resultados de la verificación | 76 |
| 4.11. | Análisis de decisiones y resolución (DAR) | 77 |
| 4.12. | Resumen | 77 |
| 5. | Propuestas | 81 |
| 5.1. | Incorporación de un plan de proyecto | 81 |
| 5.1.1. | Conclusiones | 84 |
| 5.2. | Creación de una gestión de cambio | 85 |
| 5.2.1. | Gestión de cambio en MOSKitt4ME | 87 |
| 5.3. | Gestión de riesgos | 88 |
| 5.3.1. | Gestión de riesgos en MOSKitt4ME | 88 |
| 5.4. | Manejo de costos y esfuerzo | 90 |
| 5.5. | Validaciones y Verificaciones | 91 |
| 5.5.1. | Definición de OCL | 91 |
| 5.5.2. | Uso de OCL | 92 |
| 6. | Conclusiones y trabajos futuros | 94 |
| A. | Glosario | 97 |
| A.1. | Artefactos | 97 |
| A.2. | Afirmaciones | 97 |

| | |
|---------------------------------------|----|
| A.3. Línea Base | 97 |
| A.4. Gestión Configuración | 98 |
| A.5. Requisitos | 98 |
| A.6. Revisiones entre Pares | 98 |

Bibliografía

99

Índice de Figuras

| | |
|---|----|
| 1.1. Resultados de las evaluaciones publicadas por CMMI Institute por País . . . | 3 |
| 2.1. Estructura de la representación por etapas y representación continua | 9 |
| 2.2. Niveles de madurez por área de proceso | 11 |
| 2.3. Evaluación de los niveles de madurez [8] | 16 |
| 2.4. Arquitectura general de ambiente CAME[1] | 17 |
| 2.5. Niveles de modelado de MOF | 18 |
| 2.6. Marco de trabajo conceptual de SPEM 2.0 [2] | 19 |
| 2.7. Conceptos claves del Contenido de Método vs. Proceso | 21 |
| 2.8. Estructura del metamodelo SPEM 2.0 | 22 |
| 5.1. Estructura de desglose de trabajo en MOSKitt4ME | 83 |
| 5.2. Plan de proyecto en MS Project 2010 | 84 |
| 5.3. Proceso de gestión de cambio | 86 |
| 5.4. Actividades gestión de riesgos [3] | 89 |

Índice de Tablas

| | |
|---|----|
| 2.1. Asignación peso a las prácticas específicas | 14 |
| 2.2. Resumen de los conceptos de SPEM 2.0 – 1ra parte | 24 |
| 2.3. Resumen de los conceptos de SPEM 2.0 – 2da parte | 25 |
| 3.1. Resumen comparativo entre CMMI-DEV vs. MOSKitt4ME - Nivel 2 | 44 |
| 3.2. Resumen de conceptos entre CMMI-DEV vs. MOSKitt4ME - Nivel 2 . . . | 46 |
| 4.1. Resumen comparativo entre CMMI-DEV vs. MOSKitt4ME - Nivel 3 | 78 |
| 4.2. Resumen de conceptos entre CMMI-DEV vs. MOSKitt4ME - Nivel 3 . . . | 80 |
| 5.1. Resumen comparativo SPEM 2.0 vs. Microsoft Project 2010 | 83 |
| 5.2. Prácticas específicas implementadas al incorporar el plan de proyecto . . . | 85 |
| 5.3. Prácticas específicas implementada al crear la gestión de cambios | 88 |
| 5.4. Prácticas específicas implementadas al crear la gestión de riesgos | 90 |
| 5.5. Prácticas específicas implementada al aplicar métricas | 91 |
| 5.6. Prácticas específicas implementada al usar restricciones OCL | 93 |

Dedicado a mi esposo y padres.

Capítulo 1

Introducción

Los procesos son un conjunto de actividades que se interrelacionan, donde se transforman las entradas en salidas para lograr un propósito. Éstos pueden estar compuestos por subprocesos y por elementos de procesos, y pueden formar jerarquías donde el proceso es el término del más alto nivel, los sub-procesos su siguiente nivel y los elementos de procesos los más específicos [4].

Dentro del ámbito del desarrollo de software, lo ideal es conseguir que los procesos sean reproducibles y predecibles, por lo que deben apuntar a mejorar la productividad y calidad. Para poder llevar a cabo esta meta se debe contar con una efectiva gestión de proyectos, siendo fundamental la evaluación de los procesos.

La evaluación de los procesos de desarrollo de software se realiza para determinar las necesidades, las capacidades, el grado de eficiencia y los errores, ayudando así a encontrar las soluciones de eventuales errores a tiempo. La eficiencia se logra en gran medida por la calidad de un producto (o de un sistema) que es la consecuencia de la calidad de los procesos empleados en su desarrollo y mantenimiento.

Todo proceso de desarrollo de software que desee alcanzar un nivel de calidad deberá implementar modelos de mejoras de procesos o estándares de calidad. Uno de los modelos de mejoras de procesos ha sido proporcionado por el organismo SEI (*Software Engineering Institute*) y consta de un conjunto completo e integrado que combina la madurez de los procesos de software, la calidad y la fiabilidad en los productos entregados. Este modelo se denomina CMMI (*Capability Maturity Model*[®] *Integration*) y es un modelo para la mejora y evaluación de procesos de la industria para el mantenimiento, desarrollo, la adquisición y operación de productos y servicios.

CMMI[®] se basa en las mejores prácticas para la gestión y organización del desarrollo de productos y servicios. En el nivel de madurez 3 del modelo para el desarrollo existe un área de ingeniería del software que proporciona un excelente enfoque que incluye requisitos exigentes, efectividad de los diseños, planificación y seguimiento de proyectos, realización de pruebas estrictas y realización de auditorías.

Mientras CMMI[®] es un modelo de mejoras de procesos, MOSKitt4ME [5] es un marco metodológico para la construcción de métodos de producción de software y ejecución en proyectos. MOSKitt4ME se basa en técnicas de desarrollo dirigido por modelos (MDD) y de ingeniería de métodos (ME).

MDD es un estilo de desarrollo de software donde el artefacto primario son los modelos, a partir de los cuales se obtiene el código y otros artefactos. Por otro lado, el campo de ME se centra en el diseño, la construcción y evaluación de métodos, técnicas y herramientas de apoyo para el desarrollo de sistemas de información [6].

En líneas generales, MOSKitt4ME se basa en meta-modelos y técnicas de transformación de modelos que utilizan las técnicas de MDD, aunados con los conceptos de ME para el diseño y la ejecución del método, el soporte del producto y una parte de los métodos de procesos para el desarrollo de software.

El propósito del trabajo consiste en la integración entre MOSKitt4ME y CMMI[®] para desarrollo v1.3. Para el logro de este objetivo se han evaluado, analizado y determinado las relaciones y divergencias existentes entre MOSKitt4ME y CMMI[®] para desarrollo v1.3 en los niveles de madurez 2 y 3. Una vez determinado el grado de integración se pueden formalizar propuestas para extender MOSKitt4ME.

1.1. Motivacion

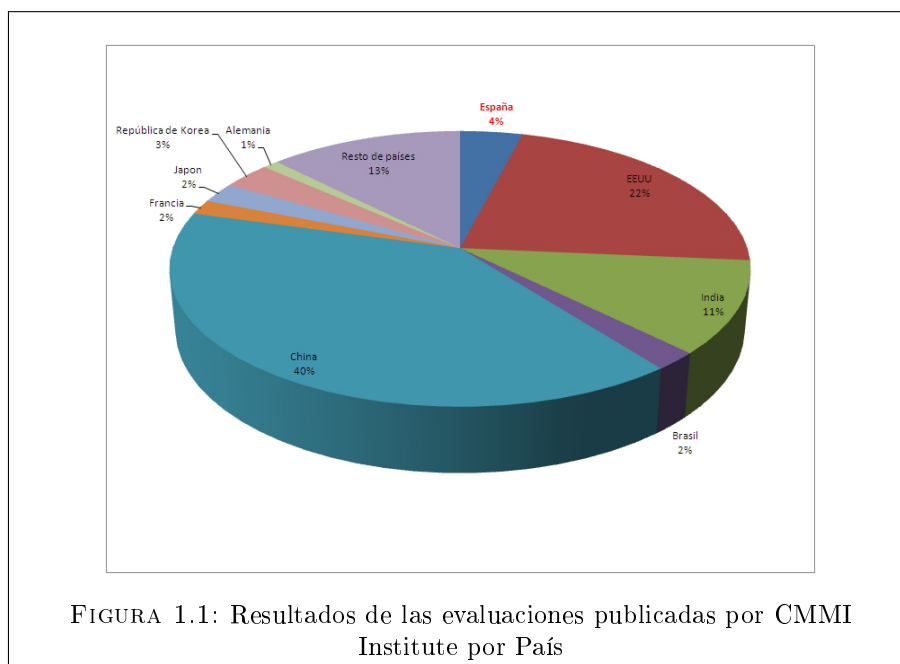
Hoy en día las organizaciones hacen uso de procesos de calidad; estos sientan las bases para la introducción y utilización de las tecnologías emergentes, que sirvan de apoyo para la consecución de los objetivos estratégicos de las organizaciones.

Las empresas tecnológicas, con el fin de mejorar la calidad y los valores agregados a sus productos, evalúan constantemente modelos que les ayuden a obtener técnicas relevantes. Para obtener mejor posicionamiento y mayor competitividad en el mercado, se emplean las certificaciones de calidad que, adicionalmente, le proporcionan a las organizaciones conocimiento acerca de sus procesos.

Una aproximación, que está siendo muy utilizada, es CMMI (*Capability Maturity Model*[®] *Integration*) para la evaluación de la calidad en los procesos. El modelo CMMI[®] contiene

las mejores prácticas de ingeniería de sistemas e ingeniería de software que se encuentran en el mercado internacional.

España representa el notable 4 %, mientras que EEUU representan un 23 % de las evaluaciones realizadas a nivel mundial, desde el 2010 hasta la fecha (ver figura 1.1). Recientemente, el interés de las empresas y organizaciones españolas en realizar las evaluaciones del nivel de madurez (1 al 5) o un perfil de rendimiento de nivel de capacidad (1 al 3) experimentó un considerable aumento.



CMMI[®] para desarrollo proporciona una guía para la creación de procesos y la medición de capacidad y madurez en una organización para el desarrollo y la entrega de software de alta calidad. Por otro lado, MOSKitt4ME proporciona un marco metodológico para ayudar en la definición de los métodos de desarrollo de software en requisitos particulares.

Incluyendo modelos de mejoras de procesos en MOSKitt4ME se tendrá el beneficio de producir métodos de forma organizada, sistemática, consistente y efectiva. Es por ello que MOSKitt4ME requiere contar con modelos de mejoras de procesos, al ser un marco metodológico para la construcción de métodos de producción de software y ejecución en proyectos. Para la sinergia con CMMI[®] es necesario un estudio para determinar el grado de integración de MOSKitt4ME con CMMI[®] para desarrollo.

Aprovechando de esta sinergia, se pueden aplicar las mejores prácticas de CMMI[®] en MOSKitt4ME, siendo un proyecto en pleno desarrollo y auge. De esta manera, se podría garantizar que los métodos (o procesos) construidos por medio de MOSKitt4ME se ajustan a las directrices y recomendaciones de CMMI[®] para desarrollo v1.3

Se pretende que las organizaciones de desarrollo de software que hacen uso de MOSKitt4ME obtengan las ventajas de las mejores prácticas para la optimización y evaluación de los procesos, dado que MOSKitt4ME estará integrado con CMMI[®] para desarrollo en los niveles de madurez 2 y 3. Con ello, se proporciona también que las organizaciones obtengan las certificaciones requeridas de los niveles de madurez 2 ó 3 de CMMI[®], lo que conduce a la calidad en sus procesos o métodos.

Las ventajas así obtenidas por las organizaciones se resumen en:

- Los métodos y procesos están bien caracterizados y comprendidos por todos los miembros de la organización. Los proyectos son más visibles, por lo tanto, la directiva y los integrantes del equipo saben en que trabajan y conocen el estado de los proyectos.
- Los procesos se gestionan más proactivamente y se describen mediante estándares, procedimientos, herramientas y métodos.
- Se genera un producto de desarrollo de software con mejor calidad, mediante una apropiada obtención de requerimientos, la detección temprana de errores y la trazabilidad de los requerimientos.
- El proceso estándar de la organización¹ se establece y se mejora a lo largo del tiempo, por lo tanto es más consistente. Adicionalmente, el proceso estándar se utiliza para establecer la integridad en toda la organización.

1.2. Objetivos

El trabajo tiene como objetivo integrar el modelo de mejoras de procesos y métodos CMMI-DEV v1.3 con el entorno de desarrollo de software MOSKitt4ME. La integración consiste en aplicar a MOSKitt4ME las mejores prácticas de calidad descritas en CMMI-DEV v1.3, hasta alcanzar los niveles de madurez 2 y 3. El propósito es obtener entornos de desarrollo de software donde se apliquen métodos o procesos que están bien caracterizados y comprendidos.

Para el logro de este objetivo general se realizan los siguientes objetivos específicos:

- Identificar el grado de implementación de las prácticas específicas del nivel de madurez 2 y 3 de CMMI-DEV v1.3 en relación a MOSKitt4ME.

¹El proceso estándar “es un conjunto de definiciones de procesos que dirigen las actividades de una organización”[4]

- Proponer soluciones que aseguren la implementación de las prácticas específicas “no implementadas” para extender MOSKitt4ME con los niveles de madurez 2 y 3 de CMMI-DEV v1.3.

1.3. Contexto de la tesis

La tesis de máster ha sido desarrollada para el centro de investigación de Métodos de Producción de Software (ProS) de la Universidad Politécnica de Valencia. Específicamente, el análisis y la propuesta de este trabajo han sido realizados dentro del contexto del proyecto MOSKitt4ME.

El proyecto de MOSKitt4ME tiene como objetivo dar apoyo a la ingeniería de métodos, proporcionando un marco metodológico para ayudar a los ingenieros de métodos en la definición de los métodos de desarrollo de software y la construcción de las herramientas de apoyo CASE ² (<http://users.dsic.upv.es/~mcervera/moskitt4me/>). El desarrollo del proyecto, a su vez, se basa en la plataforma de MOSKitt.

Modeling Software KIT (MOSKitt) es una herramienta CASE LIBRE, basada en Eclipse que está siendo desarrollada por la Consellería de Infraestructuras, Territorio y Medio Ambiente para dar soporte a la metodología gvMétrica (una adaptación de Métrica III a sus propias necesidades). (<http://www.moskitt.org/>)

1.4. Estructura del documento

Esta tesis está estructurada de la siguiente manera:

Capítulo I (este capítulo): se indican el contexto, el alcance y la organización del trabajo.

Capítulo II: se presentan los conceptos necesarios en que se enmarca la tesis y los estándares que serán mencionados a lo largo del trabajo.

Capítulo III: se realiza el análisis para determinar el nivel de implementación de las prácticas específicas del nivel de madurez 2 de CMMI-DEV v.1.3 con MOSKitt4ME; al final del capítulo se muestra un resumen del análisis comparativo realizado.

Capítulo IV: se realiza el análisis para determinar el nivel de implementación de las prácticas específicas del nivel de madurez 3 de CMMI-DEV v.1.3 con MOSKitt4ME; al final del capítulo se muestra un resumen del análisis comparativo realizado.

²Computer-Aided Software Engineering; Ingeniería de software asistida por computadora.

Capítulo V: se detallan las propuestas de mejoras a incorporar en MOSKitt4ME para aumentar el grado de implementación de las prácticas específicas de CMMI-DEV v1.3.

Capítulo VII: se resumen las conclusiones generales y se presentan posibles direcciones para trabajos futuros.

Glosario: Al final del documento se incluye un glosario con los términos comúnmente empleados a lo largo de la tesis.

Capítulo 2

Contexto Tecnológico

En este capítulo se presentan los conceptos principales del contexto de la presente tesis, dividido en dos partes principales.

La primera parte consta de la descripción de los modelos de integración de madurez y de capacidades para desarrollo (*CMMI-DEV*, *Capacity Model Maturity Integration for Development*) en la versión 1.3 y sus constelaciones, es decir, la colección de componentes que utiliza CMMI[®] para desarrollo.

Además, se presenta uno de los métodos de evaluación, SCAMPI (*Standard CMMI Appraisal Method for Process Improvement*) clase “A” versión 1.3, que se utiliza comúnmente CMMI[®] para evaluar a una organización con respecto a su grado de adecuación al modelo de mejoras de procesos. No se describe a detalle el estándar de evaluación, solamente los puntos necesarios para realizar la valoración de MOSKitt4ME.

La segunda parte describe los conceptos en que está basado MOSKitt4ME.

2.1. Integración de Modelos de Madurez de Capacidades

El modelo CMMI (*Capability Maturity Model[®] Integration*) surge como una continuación del modelo CMM (*Capability Maturity Model*) que mide el grado de madurez y de capacidad de los procesos, consiste en determinar el rango de resultado esperado que se puede lograr siguiendo un proceso. Para ello, se debe contar de forma explícita y consciente el despliegue de los procesos estando documentados, gestionados, medidos, controlados y mejorados de forma continua. Este modelo ha sido creado por el SEI (*Software Engineering Institute*) que pertenece a la Carnegie Mellon University.

Actualmente, se encuentra en la versión 1.3 de CMMI, y dentro de esta versión existen tres áreas de interés diferentes que fueron liberados en noviembre del 2010.

- CMMI-DEV (*CMMI for Development*) \Rightarrow Este modelo se enfoca al desarrollo, y en él se tratan procesos referidos a desarrollo de productos y servicios.
- CMMI-ACQ (*CMMI for Acquisition*) \Rightarrow Este modelo se enfoca a la adquisición, y en él se tratan procesos relacionados con el gobierno y con la industria, por ejemplo, la gestión de la cadena de suministros, la adquisición y contratación externa.
- CMMI-SVM (*CMMI for Services*) \Rightarrow Este modelo se enfoca a la gestión, el establecimiento y la entrega de servicios.

2.1.1. Acerca de CMMI[®] para desarrollo

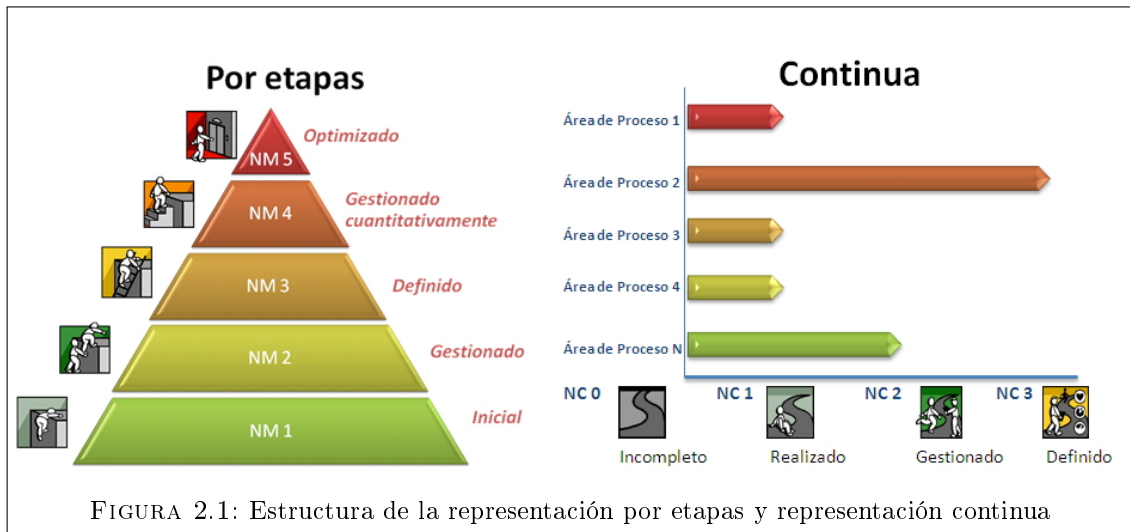
La tesis de máster se centrará en CMMI[®] para desarrollo, en adelante CMMI-DEV, que posee cinco categorías de las áreas de procesos y veintidós áreas de procesos que incluyen la gestión de proyectos, la gestión de procesos, el desarrollo de software y otros procesos de soporte para el desarrollo y mantenimiento.

En CMMI-DEV existen dos caminos o representaciones de mejora usando los niveles, como se muestra en la figura 2.1

1. La representación por etapas, el primer camino, está relacionada con la madurez organizacional que permite a las organizaciones de mejorar un conjunto de procesos relacionados, tratando de forma incremental conjuntos sucesivos de áreas de proceso.
2. El segundo camino, la representación continua, relacionada con la capacidad de cada área de proceso, permite a las organizaciones de mejorar de forma incremental los procesos que corresponden a un área de proceso individual (o un grupo de áreas de proceso), seleccionada por la organización.

2.1.1.1. Niveles de madurez

Los niveles de madurez constan de prácticas específicas y genéricas, relacionadas con un conjunto predefinido de áreas de procesos, y tienen como propósito el mejoramiento del rendimiento global de las organizaciones y, por ende, ser más productivas. Los niveles de madurez poseen la estructura “por etapas” que se muestra en figura 2.1, la imagen izquierda.



- Nivel de madurez 1 – Inicial ⇒ Este nivel es cuando no se realiza ningún tipo de proceso, o que no se consiguen sus objetivos. Son generalmente ad hoc y caóticos, donde las organizaciones no son capaces de repetir sus éxitos.
- Nivel de madurez 2 – Gestionado ⇒ Toda organización que disponga de procesos, planificación y ejecución de acuerdo con los procesos y estándares para el logro de sus objetivos. Se mantienen los procesos y estándares en períodos bajo presión y el establecimiento de los compromisos de las partes interesadas.
- Nivel de madurez 3 – Definido ⇒ A parte de tener gestionados los procesos, éstos se ajustan a la política de procesos marcada por la organización. Se mejoran a lo largo del tiempo. La gestión es más proactiva mediante las actividades del proceso.
- Nivel de madurez 4 – Gestionado cuantitativamente ⇒ Los procesos se controlan utilizando técnicas cuantitativas.
- Nivel de madurez 5 – Optimizado ⇒ Además de cumplir todas las condiciones de los niveles que le preceden, se revisa y modifica o cambia, de forma sistemática, los procesos para adaptarlos a los objetivos del negocio.

El presente trabajo se centrará en los niveles de madurez 2 y 3 de CMMI-DEV v1.3 debido a que existen prácticas que proporcionan efectividad en los diseños, planificación y seguimiento. Al estar integrado MOSKitt4ME con CMMI-DEV v1.3 en los niveles de madurez 2 y 3, es posible crear métodos y ejecución de proyectos con procesos definidos.

2.1.1.2. Niveles de capacidad

Los niveles de capacidad son la consecución de la mejora de los procesos desde el punto de vista de los procesos individuales de una organización, para mejorar de forma incremental para una determinada área.

Los niveles de capacidades, de representación “continua”, son los siguientes de acuerdo con lo mostrado en figura 2.1, del lado derecho:

- Nivel de capacidad 0 – Incompleto \Rightarrow Es un proceso que o no se realiza, o se realiza sólo parcialmente.
- Nivel de capacidad 1 – Realizado \Rightarrow Es un proceso realizado que lleva a cabo el trabajo necesario para producir productos de trabajo. Se satisfacen las metas específicas del área de proceso.
- Nivel de capacidad 2 – Gestionado \Rightarrow Es un proceso gestionado, eso quiere decir, un proceso realizado que se ha planificado y ejecutado de acuerdo con las políticas de la organización, produciendo resultados controlados.
- Nivel de capacidad 3 – Definido \Rightarrow Es un proceso definido donde los procesos se gestionan de forma más proactiva a través de la comprensión de las interrelaciones de las actividades del proceso y de las medidas detalladas del proceso y de sus productos de trabajo.

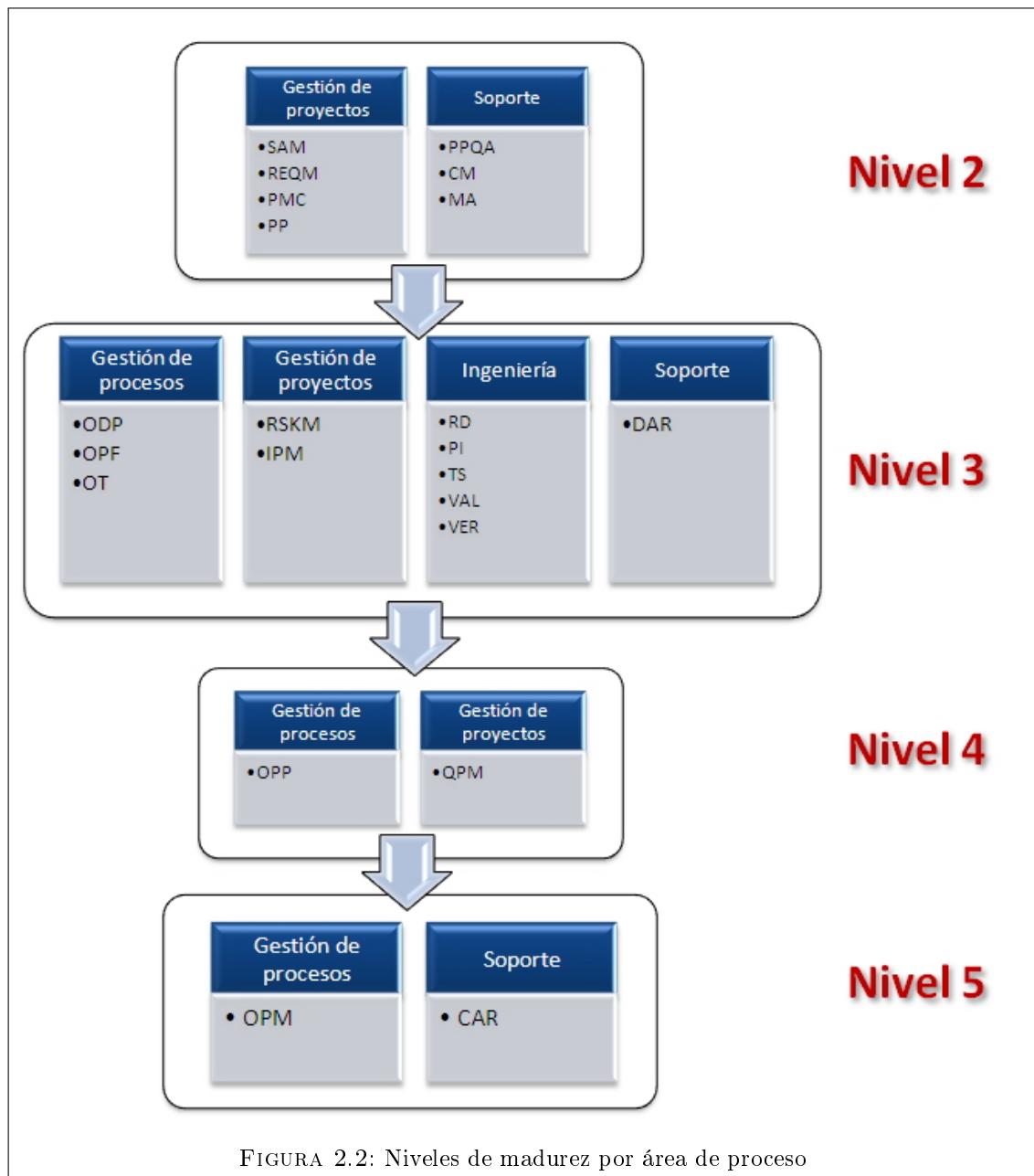
2.1.2. Componentes del área de proceso

Las constelaciones de CMMI-DEV constan de todas las metas y prácticas que se utilizan para generar los modelos CMMI; los modelos se componen de las áreas de procesos que cubren los conceptos de mejoras de los procesos de desarrollo.

CMMI-DEV se compone de 22 áreas de procesos y, a continuación, se muestra figura 2.2 que contiene las siglas de las áreas de procesos y sus categorías por niveles de madurez asociados.

A continuación, se describen los acrónimos de las áreas de procesos de CMMI-DEV, que se usan a lo largo del documento en el Capítulo 3 [Comparación de CMMI Vs MOSKitt4ME - Nivel de Madurez 2](#) y en Capítulo 4 [Comparación de CMMI Vs MOSKitt4ME - Nivel de Madurez 3](#)

- Gestión de acuerdos con proveedores (SAM)



- Gestión de requisitos (REQM)
- Monitorización y control del proyecto (PMC)
- Planificación del proyecto (PP)
- Aseguramiento de la calidad del proceso y del producto (PPQA)
- Gestión de configuración (CM)
- Medición y análisis (MA)
- Definición de procesos de la organización (OPD)

- Enfoque en procesos de la organización (OPF)
- Formación en la organización (OT)
- Gestión de riesgos (RSKM)
- Gestión integrada del proyecto (IPM)
- Desarrollo de requisitos (RD)
- Integración del producto (PI)
- Solución técnica (TS)
- Validación (VAL)
- Verificación (VER)
- Análisis de decisiones y resolución (DAR)
- Rendimiento de procesos de la organización (OPP)
- Gestión cuantitativa del proyecto (QPM)
- Gestión del rendimiento de la organización (OPM)
- Análisis causal y resolución (CAR)

Existen 4 categorías por las cuales están asociadas estas áreas de procesos que se mencionan en figura 2.2, y que consisten en:

- **Gestión de proyectos:** debe integrar las actividades que establecen y mantienen el plan de proyecto, establece y mantiene los compromisos, la monitorización del plan de proyecto y la gestión de acuerdo con los proveedores.
- **Gestión de procesos:** es el conjunto de actividades de planeación, control y ejecución de los elementos de un proceso de una empresa. Además, contiene los elementos que se representan, la capacidad de documentar y compartir las buenas prácticas, los activos de procesos, las normas, los procedimientos u otros.
- **Ingeniería:** es el conjunto de las actividades de desarrollo y mantenimiento de todas las disciplinas de ingeniería. Sin embargo, no solo de ingeniería de software, ya que se utilizan términos genéricos, para el desarrollo de producto.
- **Soporte:** incluye las actividades que dan soporte al desarrollo y mantenimiento del producto. Esta categoría está dirigida hacia el proyecto y puede ser abordada en los procesos más generales de la organización.

Las áreas de procesos están compuestas a su vez por metas y prácticas específicas, así como metas y prácticas genéricas. Dado que se ha mantenido el mismo esquema que presenta CMMI-DEV, Versión 1.3, se procede a explicar en qué consisten:

- La meta específica: describe las características únicas que deben estar presentes para satisfacer el área de proceso. Cada meta específica comienza con el prefijo “SG” (*Specific Goal*).
- La meta genérica: describe las características que deben estar presentes para establecer los procesos que implementan un área de proceso, y se utiliza en las evaluaciones para determinar si se satisface un área de proceso. Las metas genéricas comienzan con el prefijo “GG” (*Generic Goal*).
- La práctica específica: es la descripción de una actividad que se considera importante para lograr la meta específica asociada. Las prácticas específicas describen las actividades que se espera que produzcan el logro de las metas específicas de un área de proceso. Cada práctica específica comienza con el prefijo “SP” (*Specific Practice*), seguido por un número en la forma “x.y” (ejemplo, SP 1.1).
- La práctica genérica: se denomina genérica porque la misma práctica se aplica a múltiples áreas de proceso. Cada práctica genérica comienza con el prefijo “GP” (*Generic Practice*), seguido por un número de la forma “x.y” (ejemplo, GP 1.1).

2.1.3. SCAMPI

SCAMPISM A (Método de Evaluación Estándar de CMMI para la Mejora de Procesos, en inglés *Standard CMMI[®] Appraisal Method for Process Improvement A*), en la versión 1.3, es un método de evaluación que permite medir los procesos de una organización a través de unos estándares: niveles de madurez y de capacidad de un área de proceso[7].

Durante los siguientes dos capítulos se aplicará parcialmente el método de evaluación SCAMPI v1.3 para identificar el grado de implementación que posee MOSKitt4ME en relación a los niveles de madurez 2 y 3.

De acuerdo con el trabajo realizado por [8], quien valoró las áreas de procesos, las prácticas específicas y los niveles de CMMI, obtenemos la siguiente escala con algunas modificaciones realizadas en relación a la versión 1.3:

Las valoraciones posibles de prácticas específicas de acuerdo con el método SCAMPI son:

- CI \Rightarrow Completamente implementada

- AI \Rightarrow Ampliamente implementada
- PI \Rightarrow Parcialmente implementada
- NI \Rightarrow No implementada
- NA \Rightarrow No aún

Completamente implementada: hay suficientes artefactos (ver definición A.1) y/o afirmaciones (ver definición A.2) que fueron juzgados como adecuados para demostrar la implementación de la práctica, sin presentar debilidades.

Ampliamente implementada: hay suficientes artefactos (ver definición A.1) y/o afirmaciones (ver definición A.2) que fueron juzgados como adecuados para demostrar la implementación de la práctica, pero se observan una o más debilidades.

Parcialmente implementada: los datos (artefactos y / o afirmaciones) proporcionados al equipo presentan conflictos: algunos datos implican que la práctica sí se llevó a cabo mientras que otros indican lo contrario; se observan una o más de una debilidad.

No implementada: Algunos o todos los datos necesarios están ausentes o juzgados como insuficientes, los datos suministrados no apoyan las conclusiones de que la práctica se lleva a cabo; existe una o más de una debilidad.

No aún: la unidad base o función de apoyo aún no ha llegado a la etapa de la secuencia de trabajo, o la puesta en marcha para la implementación de la práctica. Este caso no ha sido tomado en cuenta durante la evaluación.

Adicionalmente, se asignó un peso numérico a las prácticas específicas descritas, con el propósito de generar los resultados de la evaluación a realizar:

| Práctica | Peso |
|----------------------------|-------------|
| Completamente implementada | 1,00 |
| Ampliamente implementada | 0,66 |
| Parcialmente implementada | 0,33 |
| No implementada | 0,00 |

TABLA 2.1: Asignación peso a las prácticas específicas

Valoraciones de las áreas de proceso de acuerdo con el método SCAMPI:

- Satisfecha
- Insatisfecha

- No aplicable
- Sin puntaje

La valoración “Satisfecha” para el modelo de representación por etapas puede depender del objetivo del nivel de madurez de la evaluación, es decir si se realiza la calificación de evaluación de nivel de madurez 2, se requerirá dar como satisfecha la meta genérica 2 con el propósito de apoyar el nivel de madurez 2 como resultado de la evaluación.

Si algunas de las metas se clasifican sin valoración y ninguna de las demás se califica “Insatisfecha”, el área de proceso está clasificada como “Sin Valoración”

Cuando un área de proceso se determina que es fuera del alcance de la unidad de organización del trabajo, el área de proceso se designa como “No Aplicable” y no ha sido calificada.

Cuando un área de proceso es aplicable fuera del ámbito de aplicación del modelo utilizado para la evaluación, el área de proceso se designa como “Fuera de Alcance” y no ha sido calificada.

Las valoraciones de los niveles finales del proceso son:

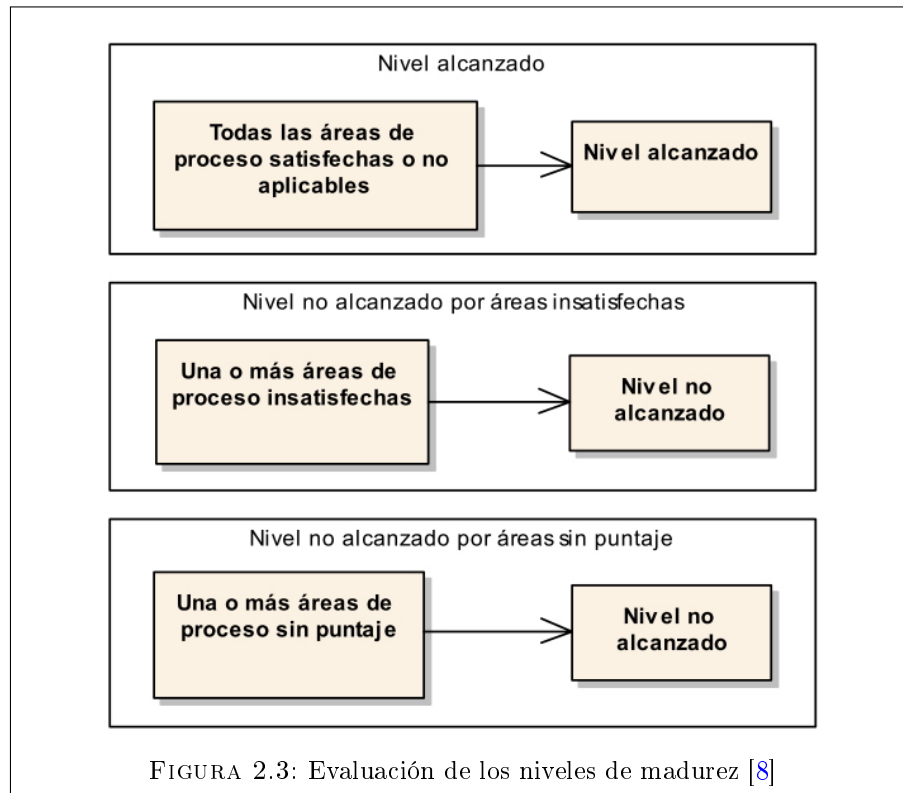
- Nivel alcanzado
- Nivel no alcanzado

En el diagrama que se muestra en la figura 2.3, se indican los niveles de madurez y su evaluación, seguiremos este mismo proceso a lo largo del presente trabajo.

2.2. Marco metodológico para construir métodos de desarrollo de software – MOSKitt4ME

MOSKitt4ME es una herramienta que se basa en un enfoque metodológico tomando en consideración a dos áreas de la ingeniería. Una de ellas es el desarrollo dirigido por modelos (MDD) fundamentándose en su infraestructura para la definición de meta-modelos y técnicas de transformación de modelos; la otra es la ingeniería de métodos (ME) utilizada para la definición de métodos de desarrollo de software.

Usando el enfoque ME se define los métodos como modelos de método basados en el estándar SPEM 2.0 – en la fase del diseño – y a partir de estos modelos – en la fase



de implementación – se obtienen de forma semiautomática los entornos CASE que da soporte a los métodos.

Adicionalmente, se combinan el estándar SPEM 2.0 con el estándar BPMN 2.0 [9] para mejorar la ejecución del proceso, apoyar en la creación del método al brindar editores gráficos y ayudar en la ejecución del método mediante el uso de un motor de proceso [5][10][11].

2.2.1. Ingeniería de métodos

La ingeniería de métodos “es la disciplina que aborda el diseño, construcción, y adaptación de los métodos, técnicas y herramientas para el desarrollo de sistemas de información” [6]. Dentro de los tipos de ingeniería de métodos se encuentra la ingeniería de método situacional (SME) definido por [12] como “es una disciplina que aborda la adaptación de métodos a la situación específica del proyecto en desarrollo”, proporcionando normas para configurar los métodos partiendo de fragmentos de métodos estándares ya existentes.

Los entornos que generalmente dan soporte al modelado de los procesos SME son las llamadas herramientas Ingeniería de Método Asistido por Computadora (CAME; *Computer Aided Method Engineering*) e Ingeniería de Software Asistida por Computadora en un meta-nivel (Meta-CASE; *Computer Assisted Software Engineering on a Meta-level*).

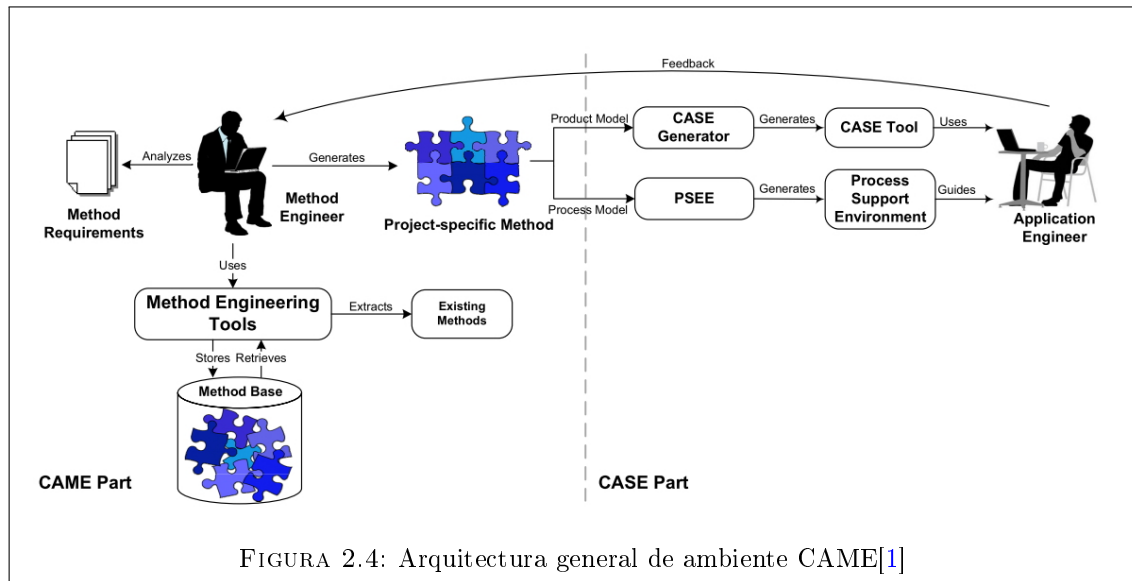


FIGURA 2.4: Arquitectura general de ambiente CAME[1]

En general, la arquitectura de una herramienta CAME se compone de dos partes (ver figura 2.4) y posee las siguientes características generales [1][11][13]:

- La parte CAME cuenta con la habilidad de proporcionar habilidades aportando una amplia gama de actividades para la ingeniería de métodos; esta tecnología posee sus inicios con la ME. Los elementos principales del CAME constituyen las herramientas de ingeniería de método (ofrecen herramientas para facilitar el trabajo a los ingenieros de métodos, por ejemplo, la extracción de componentes del método y el almacenamiento de la base del método) y la base del método (que es el núcleo del ambiente CAME). Por lo general, CAME se enfoca en la fase de diseño del método y ofrece apoyo a la especificación de los métodos del proyecto de desarrollo de software.
- La parte CASE tiene como objetivo la mejor comprensión de los modelos de empresa, sus actividades y el desarrollo de los sistemas de información. Ofrece medios para la generación de herramientas CASE del proyecto en específico, obtiene como entrada los productos y apoya a los entornos de procesos. Su insumo es el proceso centrado de entornos de ingeniería de software (*Process-centered Software Engineering Environments*; PSEEs) y se generan entornos de soporte de procesos, basados en los métodos de proceso. Por lo general, los entornos CASE se enfocan en la implementación del método, permitiendo la personalización de las herramientas CASE mediante especificaciones a un alto nivel.

Los procesos de ingeniería de método definen los aspectos relacionados con los productos y los procesos, que abarcan en líneas generales los métodos y su definición. Los productos representan los artefactos que se producen durante la ejecución del método. El proceso

es la especificación de forma precisa, completa y bien estructurada del producto, siendo útil para facilitar la comprensión del desarrollo del software. Además, se recomienda la ejecución de la especificación del proceso, debido a que es útil para guiar a los ingenieros de software en lo que respecta a los procesos de desarrollo real y la automatización final [10].

2.2.2. SPEM 2.0

SPEM 2.0 (*Software & Systems Process Engineering Meta-Model*; SPEM 2.0) es un estándar para la representación de modelos de procesos de ingeniería del software e ingeniería de sistemas. Por lo que se considera SPEM 2.0 un estándar de meta-modelado que brinda un marco conceptual para el modelado, documentación, presentación, gestión e intercambio de procesos de desarrollo de software y sus componentes, la ejecución de los procesos de desarrollo y los métodos [14].

SPEM 2.0 está basado en la especificación MOF (*Meta Object Facility*) en la cual define una arquitectura de modelado de cuatro niveles conceptuales (ver figura 2.5), desarrollado dentro del marco del OMG¹ (*Object Management Group*). SPEM 2.0 es un metamodelo que sirve para representar los modelos de los procesos del software.

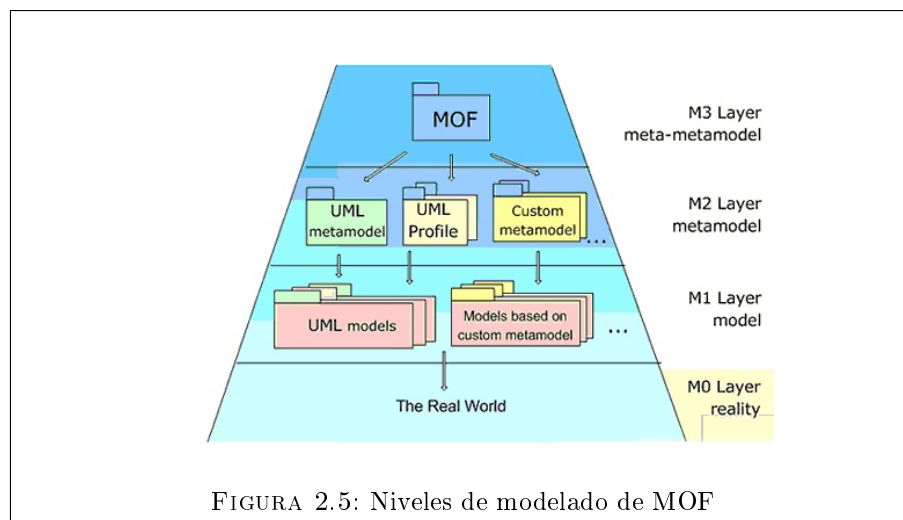


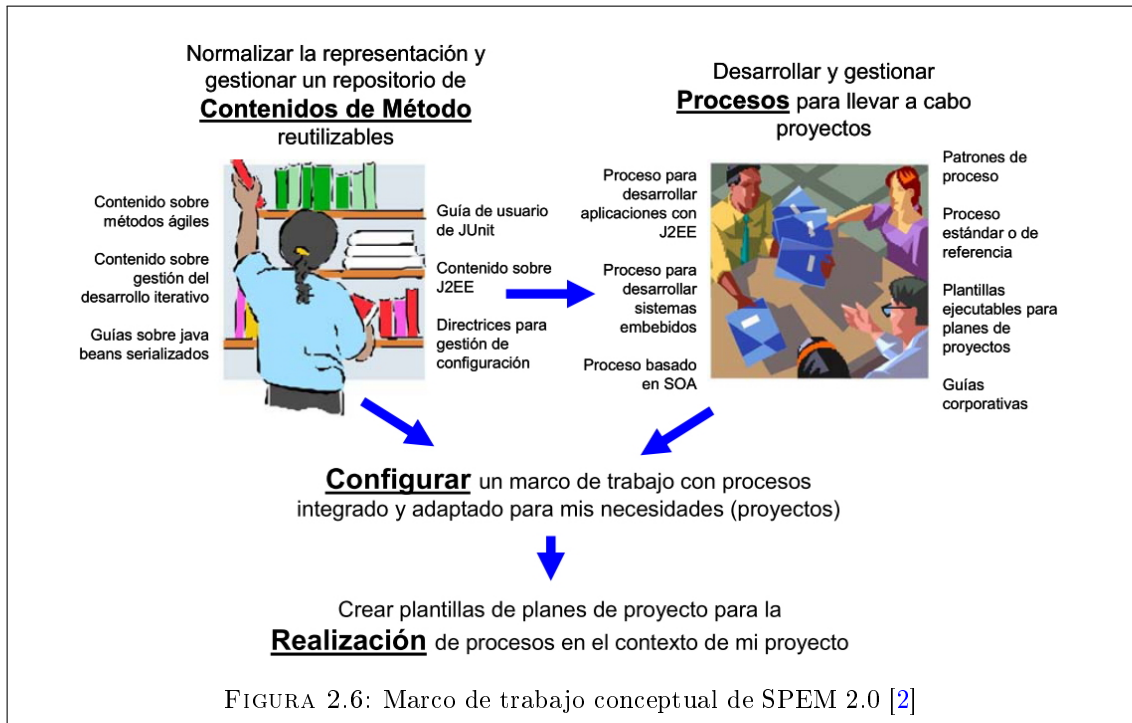
FIGURA 2.5: Niveles de modelado de MOF

Los metamodelos (M2) denotan la definición de modelos para modelos, y los modelos (M1) representan la realidad para un propósito de un cierto dominio, siendo los modelos una abstracción de la realidad (M0), dado que no se puede representar todos los aspectos de la realidad (ver figura 2.5).

¹Para mayor referencias en: <http://www.omg.org/spec/>

2.2.2.1. Características generales de SPEM 2.0

La implementación del marco conceptual del metamodelo se enmarca en las siguientes características comúnmente utilizadas (ver figura 2.6):



- **Proveer una representación estandarizada y una gestión de librerías de métodos reutilizable.** Los desarrolladores de software necesitan conocer los métodos y prácticas para los desarrollos, conociendo las tareas básicas del desarrollo, así como la especificación y gestión de requisitos, las técnicas de análisis y diseño, además de las implementaciones en todas las fases del producto.

Con este fin, SPEM 2.0 permite disponer de un formato estandarizado por el cual se puede crear una base de conocimiento para la ingeniería de procesos de desarrollo de software y sistemas que pueda ser desplegada y gestionada por los propios desarrolladores. El formato estandarizado representa el “contenido de método” reutilizable, es decir, una colección de roles, tareas, productos de trabajos, guías, fragmentos de métodos y procesos – elementos en que se basa SPEM para la representación de los procesos.

- **Soportar el desarrollo, la gestión y el crecimiento de procesos.** SPEM 2.0 admite la creación de procesos basados en el contenido de método reutilizable. El contenido de método es independiente del ciclo de vida del proceso, siendo reutilizado a lo largo del ciclo de vida de un proceso o fuera del mismo, y, dependiendo del contexto de desarrollo de cada proceso, será su adaptación o personalización. Desde

este punto de vista, los procesos pueden ser representados como flujos de trabajo (*Workflows*) y/o estructuras de desglose de trabajo (*Work Breakdown Structures*; WBS). SPEM 2.0 soporta la creación sistemática de procesos basados en la reutilización de contenidos de métodos, gracias a la reutilización de patrones de proceso.

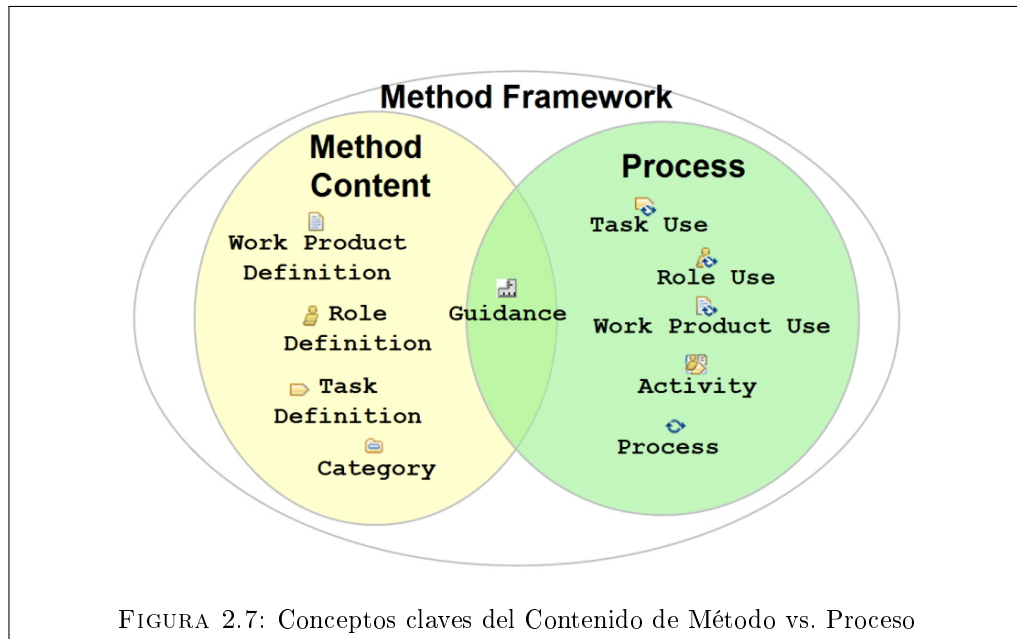
- **Soportar el despliegue sólo de los procesos y componentes de métodos necesarios en un determinado entorno mediante la definición de configuraciones de procesos y componentes de método.** Configurar un marco de trabajo consiste en desplegar los contenidos del método y procesos necesarios que sean requeridos para los equipos de desarrollo. SPEM 2.0, mediante el despliegue del contenido de método y proceso, se ajusta a la necesidad de los proyectos, tomando en consideración que ningún proyecto es exactamente igual a otro y un proceso de desarrollo nunca se ejecuta bajo las mismas condiciones. SPEM 2.0, para cumplir con estas características mencionadas, incorpora los conceptos de reutilización de procesos o patrones de procesos y la variabilidad, que ambos son procesos que incluyen alternativas configurables, y la particularización en donde los usuarios definen sus propias extensiones sobre procesos estándares reutilizables.
- **Soportar la implementación / ejecución o realización de procesos para proyectos en desarrollo.** Se considera un aporte de los procesos cuando afecta el comportamiento del equipo, por lo que las definiciones de proceso deben ser desplegadas en un formato que les permitan su ejecución de forma automática. SPEM 2.0 incluye las estructuras de definición de proceso que permiten expresar cómo un proceso será realizado de forma automática, bien sea a través de sistemas de planificación (ejemplo mediante Microsoft Project) o un motor de proceso (ejemplo mediante BPMN 2.0).

2.2.2.2. Conceptos claves SPEM 2.0

La clave de SPEM 2.0 es la división entre la definición de los métodos y la aplicación de los métodos definidos al desarrollo de un proceso en particular, como se puede apreciar en la figura 2.7:

- Primero, se puebla el contenido de método (*Method Content*) con los elementos de contenido (*Content Elements*), es decir, los elementos primarios o constructores básicos. Se definen las tareas (*Task Definition*), se organizan en distintos pasos (Steps), se define cuáles serán los productos de entrada o de salida para cada uno de ellos (*Work Product Definition*) para luego especificar quién realizará dicha tarea (*Role Definition*).

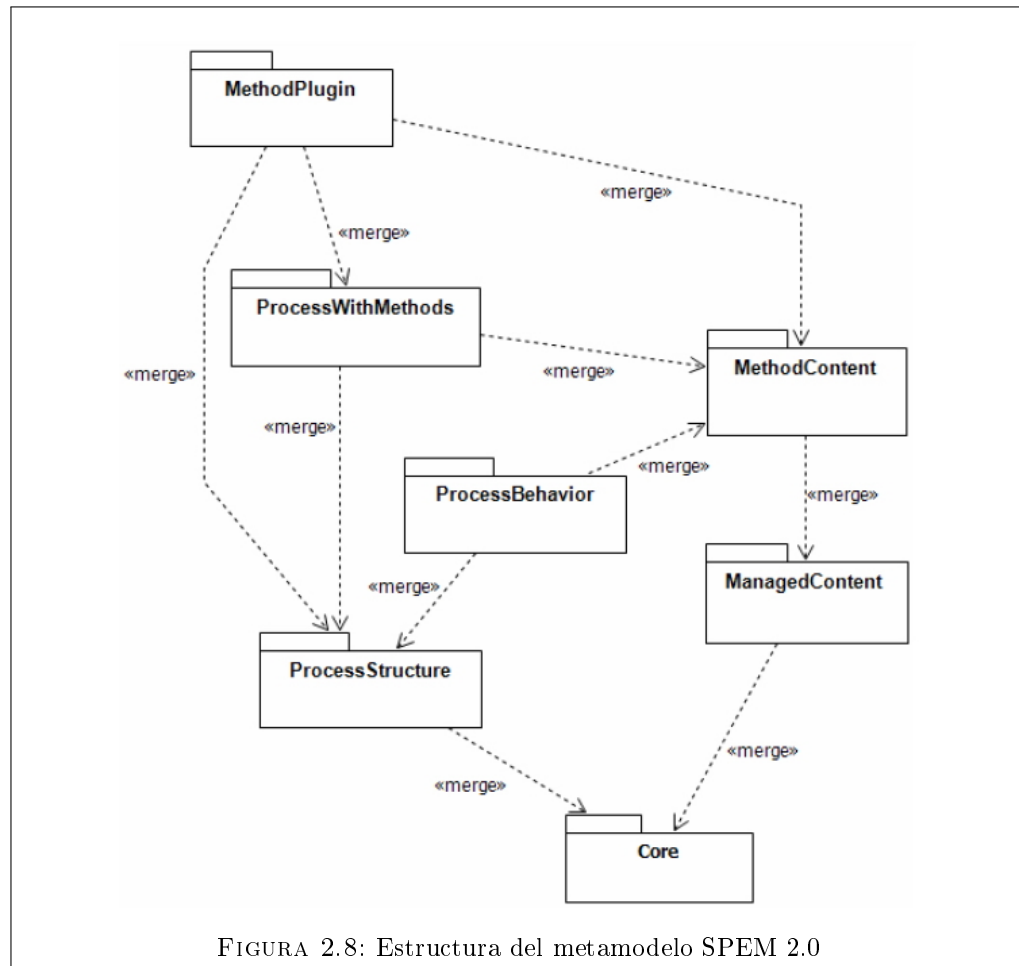
- En la intersección del contenido de método y los procesos existe la guía (*Guidance*) que consiste en plantillas, ejemplos, elementos de comprobación, etc. que dan soporte tanto al método como a los procesos.
- Después, se combinan y reutilizan dichos elementos para obtener los procesos (*Processes*), conformados por un grupo de elementos que están definidos, descritos en una serie de relaciones específicas para un método.



2.2.2.3. Descripción detallada

La división del metamodelo de SPEM 2.0 se estructura en siete paquetes (ver figura 2.8), organizados en unidades lógicas. Estas unidades se extienden, se complementan y le proporcionan estructuras adicionales a las unidades que dependen de ellas. Las unidades de la capa inferior de SPEM 2.0 se pueden implementar parcialmente sin requerir a los paquetes de la capa superior. Regularmente, las clases del metamodelo se definen de forma sencilla en unidades de niveles inferiores, para luego extenderse a unidades superiores, si se desea cumplir con requisitos más complejos del modelado del proceso, e incluso se puede utilizar diferentes niveles de capacidad, conjuntos de conceptos y niveles de formalismo para expresar sus procesos al utilizar diferentes paquetes.

A continuación, se describen las capacidades de los paquetes que se mencionan en la figura 2.8:



- **Core** \Rightarrow es el núcleo de metamodelo de SPEM 2.0 y contiene todas las clases y abstracciones que son la base para las clases de los demás paquetes. Las dos capacidades básicas que posee son:
 - crear cualificaciones definidas por el usuario (*Kinds*) que permiten establecer diferentes tipos de instancias de una clase;
 - definir trabajos expresados como procesos mediante un conjunto de clases abstractas.
- **Process Structure** \Rightarrow Contiene las clases requeridas para la creación de modelos de procesos bien sean flexibles y sencillos. Adicionalmente, proporciona la capacidad de reutilización a través del ensamblado de proceso, usando para ello un conjunto de actividades vinculadas de forma dinámica. La estructura de proceso contiene los elementos principales que permiten definir los procesos de desarrollo, como por ejemplo las actividades.
- **Process Behaviour** \Rightarrow permite la representación de la parte dinámica de los procesos, por ejemplo los diagramas de actividad de UML 2.0 (comportamiento de proceso), las máquinas de estado, etc.

- **Managed Content** \Rightarrow ayuda en la gestión e incorporación de las descripciones en lenguaje natural de los documentos y otras informaciones que son de utilidad a los procesos o a los sistemas de anotaciones. Además de eso, ayudan para capturar ciertos valores y culturas, por lo que existe libertad total para realizar combinaciones de los modelos estructurales de procesos con contenido de lenguaje natural.
- **Method Content** \Rightarrow proporciona los conceptos necesarios para construir una base de conocimientos sobre el desarrollo que puede ser independiente de los procesos específicos y los proyectos de SPEM 2.0. Contiene los principales elementos de método y puede estar organizado a voluntad del usuario mediante una jerarquía de paquetes de contenido (*Content Package*). El objetivo principal de este paquete es definir las tareas (*Task Definition*), organizarlas en distintos pasos (*Steps*), definir cuáles son los productos de entrada-salida de cada una de ellas (*Work Product Definition*) y especificar quién ha sido el que ha realizado dicha tarea (*Role Definition*).
- **Process with Methods** \Rightarrow incorpora nuevas estructuras para la integración del paquete de la estructura de proceso (Process Structure) y los conceptos y elementos del paquete del contenido de método (Method Content). Cuando se asocian los elementos del método en las partes específicas de proceso, se crean nuevas clases que heredarán los elementos del método, aunque con cambios particulares.
- **Method Plugin** \Rightarrow insiere conceptos (por ejemplo, *Method Plugin*, *Process Component*, *Variability...*) para diseñar, gestionar y mantener repositorios y bibliotecas de contenidos de métodos y procesos.

De los paquetes arriba mencionados, se procederá a explicar de forma resumida algunos conceptos relevantes, como muestra la tabla 2.2 y la tabla 2.3:

| CONTENIDO DE MÉTODO (<i>Method Content</i>) | |
|---|--|
| Tarea (<i>Task Definition</i>) | Define una unidad asignable y gestionable de trabajo, que se establece a unos pocos roles y que afecta o es afectada por uno o varios productos de trabajo (<i>Work Product</i>). |
| Rol (<i>Role Definition</i>) | Define un conjunto de habilidades, competencias y responsabilidades de un participante o de un conjunto de participantes. |
| Producto de trabajo (<i>Work Product Definition</i>) | Es un elemento que es consumido, producido o modificado por las tareas, siendo en muchos casos productos de trabajo tangibles. Contiene las asociaciones de tipo de composición (<i>Composition</i>), de agregación (<i>Aggregation</i>) y de impactado por (<i>Impacted by</i>). Adicionalmente, posee los tipos predefinidos de especialización como es de artefacto (<i>Artifact</i>), de entregable (<i>Deliverable</i>) y de resultado (<i>Outcome</i>). |
| Cualificación (<i>Qualification</i>) | Es la habilidad o competencia específica que se utiliza para modelar y representar a las cualificaciones o las condiciones requeridas para el desempeño de una tarea y / o un rol. E incluso se puede usar para encontrar individuos con cualidades específicas. |
| Paso (<i>Step</i>) | Sirve para organizar una tarea en partes o subunidades de trabajo. |
| CONTENIDO ADMINISTRADO (<i>Managed Content</i>) | |
| Guía (<i>Guidance</i>) | Un elemento descriptible que proporciona información adicional relacionada con otros elementos. Dentro los tipos de guías (<i>Guidance Kinds</i>) se encuentran: lista de comprobación (<i>Checklist</i>), concepto (<i>Concept</i>), consideraciones para el cálculo (<i>Estimation Consideration</i>), métrica para la estimación (<i>Estimation Metric</i>), ejemplo (<i>Example</i>), instrucción (<i>Guideline</i>), práctica (<i>Practice</i>), informe (<i>Report</i>), activo reutilizable (<i>Reusable Asset</i>), mapa (<i>Roadmap</i>), material de soporte (<i>Supporting Material</i>), plantilla (<i>Template</i>), definición de término (<i>Term Definition</i>), guía de herramienta (<i>Tool Mentor</i>), documentación (<i>Whitepaper</i>). |
| Categoría (<i>Category</i>) | Es un elemento descriptible que se utiliza para categorizar cualquier elemento, en base a criterios que se definan. Una categoría puede estar asociada a otras categorías creando así las jerarquías de agrupamientos de procesos que pueden ser de tipo estándar (<i>Standard Category</i>) o personalizada (<i>Custom Category</i>). |
| ESTRUCTURA DE PROCESO (<i>Process Structure</i>) | |
| Elemento de desglose de trabajo (<i>Work Breakdown Element</i>) | Es el principal tipo de elemento de desglose, ya que representa la descomposición del trabajo. Puede ser de dos tipos: actividades o hitos. Entre sus propiedades se encuentran: se puede repetir (<i>Is Repeatable</i>), son continuas (<i>Is Ongoing</i>) y están condicionados por sucesos (<i>Is Event Driven</i>). |
| Actividad (<i>Activity</i>) | Es el elemento que sirve para representar la unidad básica de trabajo o para representar un proceso en sí mismo. |

TABLA 2.2: Resumen de los conceptos de SPEM 2.0 – 1ra parte

| ESTRUCTURA DE PROCESO (<i>Process Structure</i>) | |
|--|---|
| Hito (<i>Milestone</i>) | Se trata de un evento significativo dentro del proceso de desarrollo. |
| Secuencia de trabajo (<i>Work Sequence</i>) | Es un elemento que representa una relación entre dos elementos de desglose de trabajo en la que uno de los elementos de desglose de trabajo depende del inicio o el final de otro elemento de desglose de trabajo, con el fin de comenzar o terminar, denominados como predecesor y sucesor. |
| PROCESO CON MÉTODOS (<i>Process with Methods</i>) | |
| Actividad (<i>Activity</i>) | Se define como un grupo de distintos elementos (otras actividades, tarea en uso, rol en uso, e hitos) definidos en un espacio de nombres y para los que se han descrito una serie de relaciones según el método o proyecto particulares. SPEM 2.0 tiene definidos varios tipos de actividades especiales que son la iteración (<i>Iterations</i>), las fases (<i>Phases</i>) y el proceso (<i>Process</i>). |
| Contenido de método en uso (<i>Method Content Use</i>) | Es el concepto clave para entender la separación entre procesos y el contenido de método. El contenido de método posee una serie de elementos y relaciones que son modificados debido a que se usan en un proceso particular para el cuál ha sido creado el contenido de método que pueden ser tareas, roles y productos de trabajo, denominados elementos en uso . |
| PLUG-IN DE MÉTODO (<i>Method Plugin</i>) | |
| Plug-in de método (<i>Method Plugin</i>) | Se define como una unidad de almacenamiento físico de la configuración, modularización, empaquetado e implantación de proceso y el contenido de método. Un plug-in de método se puede extender y ser extendido por otros plug-ins de método. |
| Biblioteca de método (<i>Method Library</i>) | Denominada como repositorio o biblioteca de métodos y procesos se define como una colección de una o más plug-ins y definiciones de configuraciones de métodos (<i>Method Configuration</i>). |
| Configuración de método (<i>Method Configuration</i>) | Se trata de un subconjunto lógico dentro de una biblioteca de método (<i>Method Library</i>) definido mediante una selección de paquetes de método (<i>Method Packages</i>), permitiendo definir una visibilidad dentro de la biblioteca de método, que se puede usar para filtrar contenido de método y proceso. |
| Elemento de Variabilidad (<i>Variability Element</i>) | DEste elemento sirve para proporcionar capacidades de variación y extensión de los elementos de SPEM 2.0. Se contempla 5 tipos de variabilidad entre los elementos de contenido: no aplicable (<i>not assigned</i>), contribuye (<i>contributes</i>), amplía (<i>extends</i>), reemplaza (<i>replaces</i>), amplía y reemplaza (<i>extends-replaces</i>). |

TABLA 2.3: Resumen de los conceptos de SPEM 2.0 – 2da parte

2.3. Conclusiones

Determinar las relaciones o divergencias que existen entre un marco metodológico para construir métodos de desarrollo de software (MOSKitt4ME) y las mejores prácticas de integración de los modelos de madurez y capacidades para desarrollo (CMMI-DEV) es completamente factible y se puede comprobar mediante trabajos previamente realizados.

El estudio comparativo parte del análisis de la semántica [15] y de la aplicación entre los modelos de CMMI[®] y SPEM, evaluando la correlación entre ambos modelos, e incorpora en esta comparación el estándar BPMN 2.0 realizada por [16]. El aporte de estos artículos se resume en cuanto a que el modelo de CMMI[®] se centra en la parte estructural del proceso, mientras que SPEM contiene una mayor representatividad del modelado de los procesos con un mayor grado de concreción. BPMN, por su parte, tiene como objetivo la integración de los procesos organizativos, además de su más fácil comprensión.

Los trabajos se han centrado en realizar un análisis semántico y de la aplicación entre CMMI-DEV 2.0 y SPEM 2.0. Sin embargo, el propósito de la presente investigación pretende realizar un estudio más detallado que establezca la comparación entre CMMI-DEV y MOSKitt4ME. Para ello, [17] analizó las buenas prácticas descritas por MDD propuestas en la literatura y se basó en ellas para realizar la evaluación, determinando si brindan el soporte a las prácticas específicas por CMMI-DEV v1.3 del nivel de madurez 2, cuales son las áreas definidas y el grado de soporte. Su análisis y evaluación llegaron sólo hasta el nivel de madurez 2 de CMMI-DEV v1.3.

A pesar de que no se utilizarán modelos intermedios, ni mejores prácticas como en los trabajos anteriormente mencionados, se ha demostrado que puede existir una clara relación entre las mejores prácticas de CMMI-DEV y los estándares de SPEM 2.0 y BPMN 2.0 (en donde MOSKitt4ME se fundamenta en estos estándares).

A lo largo de los siguientes capítulos, se realizará la comparación entre el marco metodológico para construir métodos de desarrollo de software y los modelos de integración de madurez y de capacidades para el desarrollo, para poder determinar el grado de implementación de las prácticas específicas de los niveles de madurez 2 y 3. A continuación, se realizarán unas propuestas factibles que permitan la integración del marco metodológico con las mejores prácticas en los niveles de madurez 2 y 3.

Capítulo 3

Comparación de CMMI Vs

MOSKitt4ME - Nivel de Madurez 2

En el nivel de madurez 2 de CMMI[®] para desarrollo en la versión 1.3 (CMMI-DEV), los proyectos deben ser gestionados y controlados. En este nivel, se debe garantizar para la ejecución en los proyectos que los procesos sean: planificados, ejecutados de acuerdo con las políticas, monitorizados, controlados y revisados.

A continuación, se realizará una comparación de las áreas de procesos del nivel de madurez 2 de CMMI[®] para desarrollo en su versión 1.3 (CMMI-DEV), con el marco metodológico para la construcción de métodos de producción de software, determinando de esta manera si las prácticas específicas de las áreas de procesos que se analizan son implementadas en MOSKitt4ME o no.

3.1. Gestión de acuerdos con proveedores (SAM)

El propósito del área de gestión de acuerdos con proveedores es gestionar la adquisición de productos y servicios con los proveedores, implicando actividades como: determinar el tipo de adquisición, seleccionando, manteniendo y ejecutando acuerdos con los proveedores, aceptar y asegurar la entrega satisfactoria de los productos adquiridos; un ejemplo de los mismos son los subsistemas.

Se dice que el área de proceso no es aplicable por MOSKitt4ME, debido a que no contempla la adquisición de productos, servicios, componentes de productos o servicios que puedan ser adquiridos y gestionados a través de proveedores.

Valoración: No aplicable.

3.2. Gestión de requisitos (REQM)

El área de gestión de requisitos (ver definición A.3) tiene como propósito gestionar los requisitos y componentes de los productos del proyecto. De esta manera se asegura la alineación entre los requisitos, los planes y los productos de trabajo del proyecto.

3.2.1. SG 1 Gestionar los requisitos

La gestión de requisitos son procesos que realizan la identificación, asignación, validación y modificación de los requisitos. También se encarga de identificar las inconsistencias con los planes y los productos de trabajo.

3.2.1.1. SP 1.1 Comprender los requisitos

Esta práctica específica se basa en analizar los requisitos para asegurar que se alcanza una comprensión compatible y compartida del significado de los requisitos, a medida que maduran los proyectos. Los resultados de estos análisis y diálogos son un conjunto de requisitos aprobados.

En MOSKitt4ME, se puede hacer uso de la primitiva disciplinas (*Disciplines*) para categorizar¹ las tareas y así establecer un conjunto de tareas dedicadas a la comprensión de los requisitos. Las disciplinas tienen un objetivo en común dentro del proyecto completo. En la medida que el proyecto se desarrolla todos los requisitos se han de ir incorporando a esta categoría que se ha definido, para su posterior análisis y comprensión.

Valoración: Completamente implementada.

3.2.1.2. SP 1.2 Obtener el compromiso sobre los requisitos

La práctica específica indica que los participantes involucrados se comprometen con el cumplimiento de los acuerdos y compromisos, lo cual implica llevar a cabo los requisitos actuales y aprobados.

En MOSKitt4ME, no es posible de certificar los acuerdos entre los roles (que representarían las participantes involucradas), en un principio porque no se puede determinar cuándo cambian los requisitos y cuándo se produce un nuevo requisito o llevan una documentación de los cambios. Por ende, no se puede evaluar el impacto de los nuevos requisitos sobre los compromisos de los requisitos ya existentes.

¹Las disciplinas son un tipo predefinidos de categorías

Valoración: No implementada.

3.2.1.3. SP 1.3 Gestionar los cambios a los requisitos

La gestión de cambios a los requisitos consiste en almacenar y documentar de forma eficiente y eficaz todas las posibles modificaciones que puedan ocurrir a lo largo del trabajo en cuanto a los requisitos existentes. Para determinar el impacto de los cambios es necesario saber la fuente de los requisitos, documentando la razón de cualquier cambio que llegue a ocurrir en los requisitos.

A través de MOSKitt4ME no es posible realizar el proceso de gestión de cambios, debido a que no mantiene un histórico, ni documentación de las posibles variaciones que puedan suceder en los requisitos a lo largo del proyecto.

Valoración: No implementada.

3.2.1.4. SP 1.4 Mantener la trazabilidad bidireccional de los requisitos

La práctica específica permite comprobar la completitud de los requisitos, al determinar la asociación que se pueda establecer claramente entre los requisitos de los diferentes niveles inferiores o superiores. La trazabilidad bidireccional se utiliza para los sistemas de seguimiento de requisito fuente hasta sus requisitos de más bajo nivel y viceversa (como también las relaciones a otras entidades, por ejemplo productos de trabajo intermedios y finales).

SPEM 2.0 posee un elemento del método denominado variabilidad (*Variability*) que permite definir ciertos tipos de diferencias con el respecto al elemento original. Aplicando el plugin o no permite incluir o remover estas capacidades de variabilidad en los elementos. Entre los tipos de variabilidad se encuentra la contribución (*Contribution*) que le aporta con sus propiedades al elemento base sin alterar directamente ninguna de las propiedades ya existentes, es decir, solo añade propiedades. Este tipo de variabilidad se puede utilizar para crear la trazabilidad bidireccional al añadir propiedades que le permitan ver la asociación que existen entre los elementos, ya que puede ser creado entre los roles, tareas o productos de trabajo. Por ende, si se definen los requisitos como un grupo de tareas categorizadas, se puede aplicar esta primitiva.

Valoración: Completamente implementada.

3.2.1.5. SP 1.5 Asegurar el alineamiento entre el trabajo del proyecto y los requisitos

La práctica específica consiste en determinar las inconsistencias e iniciar las acciones correctivas que ocurran entre: los requisitos, los planes del proyecto y los productos de trabajo.

En SPEM 2.0, al contar entre sus características con la variabilidad de los elementos de contenidos, específicamente la contribución, por consecuencia, la trazabilidad bidireccional de los requisitos, se puede determinar si existe inconsistencia entre los requisitos haciendo un seguimiento y control. Adicionalmente, en la fase de implementación del método, al verificar los resultados, se puede determinar si los requerimientos concuerdan con lo planificado. Sin embargo, al aplicar las acciones correctivas en MOSKitt4ME, implicaría volver a implementar el método en esta fase.

Valoración: Completamente implementada.

3.3. Monitorización y control del proyecto (PMC)

La monitorización y control del proyecto proporcionan ayuda para determinar el progreso del proyecto, evaluando las posibles variaciones que puedan ocurrir y que desvíen de forma significativa el plan global del proyecto. Esta monitorización servirá para poder tomar las acciones correctivas que se consideren apropiadas.

3.3.1. SG 1 Monitorizar el proyecto frente al plan

La meta específica se encarga de monitorizar el progreso y el avance real del proyecto frente al plan de proyecto.

3.3.1.1. SP 1.1 Monitorizar los parámetros de planificación del proyecto

La práctica específica indica los siguientes puntos: los parámetros de planificación del proyecto que constituyen los indicadores típicos del progreso y del rendimiento del proyecto y, también, incluyen atributos de los productos de trabajo, de las tareas, del coste, del esfuerzo y del calendario. A parte de esto, los atributos de los productos de trabajo, incluyendo el tamaño, la complejidad, el nivel de servicio, la disponibilidad, el peso, la forma, el ajuste, la función y, finalmente, la frecuencia de monitorización de los parámetros. Se deben comparar los valores reales de los parámetros con la planificación del proyecto para así determinar las posibles variaciones que puedan ocurrir.

Existen ciertos parámetros que se pueden monitorizar en MOSKitt4ME - como es el caso del rendimiento del proyecto - a través de la revisión periódica del avance de las actividades e hitos. Dicho rendimiento del proyecto se puede visualizar mediante la aplicación del método, en el ambiente CASE. Los demás parámetros no están contemplados en MOSKitt4ME, por lo tanto, no se puede llevar a cabo una monitorización completa de la planificación del proyecto.

Valoración: Parcialmente implementada.

3.3.1.2. SP 1.2 Monitorizar los compromisos

La monitorización de los compromisos debe realizar en relación con las obligaciones que han sido identificadas en el plan del proyecto, efectuando una revisión, identificación y documentación de los compromisos internos y externos.

Al desarrollar el contenido de método en MOSKitt4ME, se definen las tareas. El contenido de método son los compromisos. Entonces, es posible realizar la monitorización del cumplimiento de las tareas, ya que ellas representan las porciones más pequeñas del trabajo en un modelo de proceso. Al realizar la implementación del método y generarse el CASE, se puede ver claramente, en qué fase o etapa se encuentra la ejecución de las tareas.

Valoración: Completamente implementada.

3.3.1.3. SP 1.3 Monitorizar los riesgos del proyecto

Al igual que los compromisos, los riesgos que se han identificado se deben monitorizar en relación con el plan de proyecto.

El método para construir métodos de desarrollo de software (MOSKitt4ME) no contempla el manejo de los riesgos que consiste en la verificación del comportamiento de los mismos, es decir, la comunicación del cambio de probabilidad que suceda y del cambio de prioridad de los riesgos.

Valoración: No implementada.

3.3.1.4. SP 1.4 Monitorizar la gestión de los datos

Dependiendo de los cambios de requisitos y del estado del proyecto, será necesaria o no la replanificación de los datos del proyecto. Razón por la cual se requiere la monitorización de los datos para lograr una correcta gestión de los mismos.

Para ello, MOSKitt4ME cuenta con los llamados artefactos, los cuales son de naturaleza tangible. Un ejemplo de los artefactos son los modelos, documentos, códigos, archivos, etc. Éstos representan los datos y pueden determinar si se han ejecutado de forma satisfactoria, al ejecutarse los productos de trabajo a los cuales los artefactos se encuentran asociados. Cumpliendo así con la práctica específica de monitorizar la gestión de los datos del proyecto frente al plan de proyecto, en la fase de implementación del método.

Valoración: Completamente implementada.

3.3.1.5. SP 1.5 Monitorizar la involucración de las partes interesadas

La práctica específica se pide la comprobación de las partes interesadas frente al plan de proyecto. En SPEM 2.0 las partes interesadas representarían los roles que se encuentran asociados a las tareas; los mismos se definen como un conjunto de habilidades, competencias y responsabilidades relacionadas de un individuo o de un grupo. El proceso de monitorización y control de las partes interesadas y su participación en el proyecto se pueden llevar a cabo mediante la implementación del método, en la vista de proceso al seleccionar un rol específico, verificando si se ha realizado la tarea.

Valoración: Completamente implementada.

3.3.1.6. SP 1.6 Llevar a cabo las revisiones del progreso

Para determinar el progreso de los proyectos, se requiere saber en un momento determinado el estado de los mismos, con el propósito de mantener informadas a las partes interesadas.

En MOSKitt4ME, se consigue una representación ejecutable del proceso del método mediante la transformación del diseño del método a un conjunto de procesos, usando para ello el estándar de BPMN 2.0. En las vistas de procesos, generadas en la transformación, es posible observar las tareas que han sido ejecutadas y, por ende, el estado actual de los procesos.

Valoración: Completamente implementada.

3.3.1.7. SP 1.7 Llevar a cabo las revisiones de hitos

Los hitos son eventos planificados con antelación, y son revisiones formales del estado del proyecto que se realizan de forma minuciosa y detallada. El propósito es comprender cómo y si se están cumpliendo los requisitos planificados.

Debido a que se pueden definir en SPEM 2.0 - estándar en el que se basa MOSKitt4ME - los hitos de trabajo como los llamados “Milestones” que representan un evento significativo para el desarrollo de un proyecto. Cumpliendo el mismo objetivo para CMMI-DEV como para SPEM 2.0.

Valoración: Completamente implementada.

3.3.2. SG 2 Gestionar las acciones correctivas hasta su cierre

El propósito de la meta específica es gestionar las acciones correctivas ante posibles desviaciones del plan, recopilando y analizando las posibles acciones que se deben tomar, efectuando las acciones correctivas para luego determinar si éstas han sido efectivas para corregir las desviaciones del plan.

El método de desarrollo del software permite realizar la monitorización del proyecto, sin embargo, no permite determinar si han existido desviaciones del plan, ya que no mantiene un histórico u otro medio que indique que haya existido una desviación del plan.

Valoración: No implementada.

3.4. Planificación del proyecto (PP)

La planificación del proyecto consiste en establecer los planes necesarios para llevar a cabo las tareas o actividades que se requieren a lo largo de los proyectos.

Para que todo proyecto tenga éxito, se debe: desarrollar el plan de proyecto, interactuar con las partes interesadas de forma apropiadas, obtener el compromiso del plan y por último mantener el plan.

3.4.1. SG 1 Establecer las estimaciones

Los parámetros de planificación de proyecto deben ser establecidos y mantenidos. Con el propósito mantener toda la información necesaria que el proyecto necesite para su planificación y organización, en la medida que el proyecto se ejecute.

3.4.1.1. SP 1.1 Estimar el alcance del proyecto

La estimación del alcance del proyecto implica que se debe establecer la estructura de descomposición del trabajo (WBS; *Work Breakdown Structure*) de alto nivel. La WBS

ofrece un esquema para la identificación, organización y gestión de estos paquetes de trabajo que resultan de la descomposición mencionada.

La WBS para MOSKitt4ME representa los procesos (*Processes*). En SPEM 2.0, se encuentran dos etapas en la definición del método (*Method Library*). Primero se definen los constructores básicos en el contenido de método, por ejemplo los roles, las tareas, los productos de trabajo, las guías, etc. Segundo, se combinan y se reutilizan estos constructores básicos para ensamblar las actividades y los procesos. Los patrones ensamblados de procesos ayudan a dividir el proyecto global en un conjunto de componentes manejables que están interconectados, como es el caso de las tareas o los productos de trabajo.

Valoración: Completamente implementada.

3.4.1.2. SP 1.2 Establecer las estimaciones de los atributos de los productos de trabajo y de las tareas

El atributo tamaño es una de las entradas principales para realizar las estimaciones de los modelos y, con base en el tamaño, se le asigna a los productos de trabajo y a las tareas un nivel relativo de dificultad y complejidad. El atributo tamaño se utiliza también para estimar el costo y el calendario del proyecto, así como el nivel de servicio, la conectividad, la disponibilidad y la estructura.

En esta práctica específica se dice que es ampliamente implementada, debido a que en SPEM 2.0 existe el elemento de desglose de trabajo (*Work Breakdown Element*) que es el tipo de elemento principal ya que representa la descomposición del trabajo. El elemento de desglose de trabajo posee dos tipos, que son las actividades y los hitos. Entre las propiedades comunes de las actividades y los hitos están:

- Se pueden repetir: existen varias iteraciones
- Son continuos: son trabajos sin duración fija
- Poseen condiciones de sucesos: el inicio es determinado por un evento en especial

Adicionalmente, SPEM 2.0 posee el elemento del método llamado guía que provee información adicional relacionada con otros elementos. Para la estimación de métricas y medidas existe una guía llamada métrica para la estimación (*Estimating Metric*) que se usa comúnmente para calcular el tamaño del esfuerzo, así como el mejor potencial de trabajo.

Estas primitivas determinan el atributo tamaño y, por ende, ayudan a identificar la dificultad, la calidad, el nivel de servicio y la complejidad de los productos de trabajo y de las tareas.

Sin embargo, en MOSKitt4ME, al no contar con las primitivas de costos y tiempos no se permite establecer completamente las estimaciones de los atributos de los productos de trabajo y de las tareas de la práctica específica.

Valoración: Ampliamente implementada.

3.4.1.3. SP 1.3 Definir las fases del ciclo de vida del proyecto

Las fases del ciclo de vida de un proyecto se definen con el objetivo de proporcionar ciertos periodos planificados de revisiones o evaluaciones de los proyectos. Por lo general, son puntos lógicos de decisión, facilitando la oportunidad de tomar acciones correctivas de existir algún debió del plan original.

Dado que en SPEM 2.0 se pueden utilizar la primitiva fase (*Phase*) proporcionada por éste estándar para definir un periodo de tiempo representativo en un proyecto, finalizando generalmente con grupo importante de entregables concluidos o hito. Debido a éste componente que posee MOSKitt4ME se puede concluir que es perfectamente implementada esta práctica específica.

Valoración: Completamente implementada.

3.4.1.4. SP 1.4 Estimar el esfuerzo y el coste

Las estimaciones de esfuerzo y costo se basan en los resultados de análisis de modelos o datos históricos aplicados al tamaño y a las actividades.

Dos atributos importantes que MOSKitt4ME no maneja son los datos históricos y los costos.

Valoración: No implementada.

3.4.2. SG 2 Desarrollar un plan de proyecto

La base de la planificación de un proyecto es plan de proyecto, el cual consiste en los requisitos y las estimaciones establecidas considerando todas las fases del ciclo de vida de los proyectos, para ello se debe establecer y mantener un documento formal y aprobado.

3.4.2.1. SP 2.1 Establecer el presupuesto y el calendario

La práctica específica consiste en constituir y mantener el presupuesto y el calendario del proyecto. El presupuesto y el calendario se deben basar en las estimaciones que se han desarrollado previamente, con una correcta asignación de recursos, una adecuada complejidad y dependencia en las tareas.

A pesar de poder definir en el método la complejidad de las tareas y las dependencias entre las mismas mediante la secuencias de trabajo, no es posible definir las estimaciones de un presupuesto ni el calendario del proyecto mediante MOSKitt4ME.

Valoración: No implementada.

3.4.2.2. SP 2.2 Identificar los riesgos del proyecto

Implica identificar los asuntos potenciales, posibles peligros, amenazas, vulnerabilidades, entre otros que afecten de forma negativa el correcto desenvolvimiento de los planes del proyecto. Los riesgos se descubren y se analizan para dar soporte a la planificación, modificando los riesgos que sean necesarios.

En MOSKitt4ME no es posible realizar la identificación, el análisis, determinar el impacto, la probabilidad de ocurrencia y la priorización de los riesgos.

Valoración: No implementada.

3.4.2.3. SP 2.3 Planificar la gestión de los datos

Los datos son las diferentes documentaciones que posee un proyecto que sirven para darles soporte en todas sus áreas y podrán ser entregables o no entregables.

Para MOSKitt4ME, los datos se representan como los productos de trabajo (*Work Product Definition*), los cuales pueden ser producidos, consumidos o modificados por las tareas. Por lo tanto, pueden ser perfectamente planificados para su gestión.

Valoración: Completamente implementada

3.4.2.4. SP 2.4 Planificar los recursos del proyecto

Los recursos del proyecto se definen como los materiales, talento humano, equipamiento, métodos, etc. Se basan en estimaciones iniciales, proporcionando información adicional para extender la estructura de descomposición de trabajo (*WBS*). La WBS se apoya en

actividades o productos de trabajo, tareas o una combinación de estos elementos y va acompañada de diccionarios que describen el contenido del trabajo que realiza la WBS, así como diagramas de procesos e informes de estado.

La práctica específica, llamada planificar los recursos del proyecto, se cumple en MOSKitt4ME mediante la siguiente relación:

- La estructura de descomposición de trabajo (*WBS*) está representada en SPEM 2.0 por los elementos de desglose de trabajo (*Work Breakdown Element*)
- Los diccionarios describen el contenido del trabajo que realiza la WBS a través de las guías (*Guidance*), específicamente por el tipo de guía denominado concepto (*Concept*)
- El estándar de BPMN 2.0 proporciona los diagramas de procesos y permite ver por completo la planificación de los recursos
- Los informes de estado mediante las guías, en especial con el tipo de guía llamada informe (*Report*)

Valoración: Completamente implementada.

3.4.2.5. SP 2.5 Planificar el conocimiento y las habilidades necesarias

En esta práctica específica se requiere planificar las necesidades de conocimiento y de habilidades para realizar el proyecto; para ello se requiere poseer inventario de las habilidades necesarias, bases de datos con las habilidades y los planes de formación.

En un principio, MOSKitt4ME es desarrollado por los profesionales más experimentados, como el ingeniero del método, quien analiza los requerimientos del mismo y genera los métodos de especificación del proyecto, además de ser un experto del dominio. En segundo lugar, SPEM 2.0 posee las primitivas en los roles y las tareas como son las cualificaciones, habilidades o competencias que permiten incorporar bien sea en las tareas o en los roles los conocimientos requeridos para realizar una actividad en particular. Lo que no posee MOSKitt4ME es el desarrollo de los planes de formación.

Valoración: Ampliamente implementada.

3.4.2.6. SP 2.6 Planificar la involucración de las partes interesadas

Para determinar las partes interesadas e identificarlas en las fases del ciclo de vida del proyecto, determinando las funciones, su grado de importancia y el grado de interacción

existe la definición de los roles (*Role Definition*) en MOSKitt4ME, los cuales se definen como un conjunto de habilidades, competencias y responsabilidades relacionadas de un individuo o de un grupo. Y la matriz bidimensional, planteada por la práctica específica, se puede definir en las llamadas tareas (*Tasks*) donde se describe la unidad atómica de trabajo para los procesos, afectando a unos pocos productos de trabajo y vinculando a unos pocos roles.

Valoración: Completamente implementada.

3.4.2.7. SP 2.7 Establecer el plan de proyecto

El plan de proyecto permite una comprensión del plan general a todas las personas u organizaciones involucradas para dar soporte al plan. Por ello, se requiere un documento que posea todos los componentes relevantes de la planificación.

Para la comprensión del plan general se puede utilizar la vista consolidada (*Consolidated View*), que proporciona el editor de EPF Composer que es el editor de SPEM 2.0, y que está integrada en MOSKitt4ME. Esta vista posee la información del desglose de las actividades e hitos, los roles y los productos de trabajo.

Valoración: Completamente implementada.

3.4.3. SG 3 Obtener el compromiso con el plan

Las prácticas específicas requieren que en la planificación del proyecto las personas responsables adquieran el compromiso para que sean implementadas y den el soporte al plan.

El método de desarrollo de software no permite llevar un control de las posibles modificaciones que se puedan producir, o de los presupuestos que se negocien como resultado de los compromisos realizados.

Valoración: No implementada.

3.5. Aseguramiento de la calidad del proceso y del producto (PPQA)

El área de aseguramiento de calidad del proceso y del producto tiene como objetivo ofrecer y dar soporte a la entrega de productos de alta calidad. Proporcionando a las

personas involucradas del proyecto, la visibilidad idónea sobre los procesos y los productos de trabajo asociados, conforme a las especificaciones.

3.5.1. SG 1 Evaluar objetivamente los procesos y los productos de trabajo

La evaluación consiste en comparar de forma objetiva los estándares y procedimientos contra los productos de trabajo y los procesos realizados.

3.5.1.1. SP 1.1 Evaluar objetivamente los procesos

Se debe realizar una evaluación objetiva de los procesos que se han ejecutado en relación con la documentación de la descripción de los procesos y procedimientos. Para el cumplimiento de esta práctica se recomienda realizar informes de evaluación y de no conformidad y, de ser preciso, tomar las acciones correctivas.

En MOSKitt4ME, se puede realizar la evaluación objetiva de los procesos y procedimientos una vez ejecutada la tarea. Debido a que la información se encuentra documentada en la pestaña que contiene los avances (*Preview*), los procesos de despliegue (*Delivery Process*), los patrones de procesos (*Capability Patterns*), el desglose de las estructuras de trabajo o incluso los diagramas de procesos de BPMN 2.0, es posible visualizar la descripción de los procesos. Al contar con la documentación de los procesos se puede realizar la evaluación y, en el caso de no conformidad, tomar las acciones correctivas. Sin embargo, no es posible mediante MOSKitt4ME de establecer los criterios de evaluación de los procesos.

Valoración: Ampliamente implementada.

3.5.1.2. SP 1.2 Evaluar objetivamente los productos de trabajo

Se debe evaluar objetivamente los productos de trabajo seleccionados frente a las descripciones de proceso, estándares y procedimientos aplicables.

MOSKitt4ME al tener definidos los productos de trabajo (*Work Product Definition*), de tres tipos predefinidos - artefactos (*Artifact*), entregables (*Deliverable*) o de resultado (*Outcome*) -, éstos pueden ser evaluados de forma objetiva con las descripciones de procesos que poseen.

Valoración: Completamente implementada.

3.5.2. SG 2 Proporcionar una visión objetiva

Sí durante la evaluación de los estándares y procedimiento existen no conformidades se debe seguir y comunicar de forma objetiva, asegurando su solución.

3.5.2.1. SP 2.1 Comunicar y resolver las no conformidades

Esta práctica específica busca comunicar los problemas de calidad o las no conformidades con el personal, que se han identificado ante la falta de conformidad con los estándares, y propone acciones correctivas ante las posibles desviaciones.

MOSKitt4ME no posee esta funcionalidad.

Valoración: No implementada.

3.5.2.2. SP 2.2 Establecer los registros

Se deben establecer los registros de las evaluaciones realizadas y las acciones correctivas que se han tomado con el informe del estado en el cual se encuentran las medidas correctivas tomadas y las tendencias de calidad.

Estos puntos anteriormente mencionados, se pueden extraer de MOSKitt4ME, aunque no es posible llevar a cabo los registros de las evaluaciones, ni un control de las acciones correctivas.

Valoración: No implementada.

3.6. Gestión de configuración (CM)

El área de gestión de configuración (ver definición [A.4](#)) se focaliza en un control muy estricto de los elementos de configuración. Ésta área establece y mantiene la integridad de los productos mediante la identificación, el control, el informe de estado y las auditorías de la configuración.

3.6.1. SG 1 Establecer las líneas base

Se deben crear las líneas bases (ver definición [A.3](#)) de los productos de trabajo.

3.6.1.1. SP 1.1 Identificar los elementos de configuración

Los elementos de configuración pueden incluir el hardware, los activos tangibles, el software y la documentación o incluso los resultados de las pruebas. Un elemento de configuración puede consistir en varios productos de trabajo relacionados, agrupados en forma lógica que proporcionan facilidad de identificación y un acceso controlado para formar una línea base. Para ello, se debe identificar los elementos de configuración, los componentes y los productos de trabajo.

En SPEM 2.0 hay un elemento denominado configuración de método (*Method Configuration*) permite definir vistas de una misma biblioteca de plugins, basada en las definiciones de otras configuraciones. Permitiendo de esta manera la definición de subconjuntos lógicos de una biblioteca de métodos, mediante la selección de paquetes deseados, bien sea de contenido de método como de procesos. Permitiendo de esta manera la identificación de los elementos de configuración para formar una línea base.

Valoración:: Completamente implementada.

3.6.1.2. SP 1.2 Establecer un sistema de gestión de configuración

La gestión de configuración incluye en los medios de almacenamiento, procedimientos y herramientas para acceder al sistema, y puede constar de múltiples subsistemas de acuerdo con el entorno de la configuración. Contando con un sistema de gestión de cambios que den soporte a los medios de almacenamiento, los procedimientos, y herramientas.

MOSKitt4ME no dispone de primitivas para registrar y acceder a las peticiones de gestión de cambios.

Valoración: No implementada.

3.6.1.3. SP 1.3 Crear o liberar las líneas base

Una línea base es un conjunto de requisitos, de diseño, códigos fuentes, archivos de construcción y documentación de usuario. La línea base representa a una colección de elementos de configuración que se le asigna un identificador y que representan un momento determinado en el tiempo.

MOSKitt4ME cuenta con la colección de elementos de configuración, desarrollados en la etapa de definición del método, aunque no posee una propiedad importante que es mantener los datos históricos o mantener un sistema de gestión de cambios, que le permita crear o liberar las líneas bases (ver definición [A.3](#)).

Valoración: No implementada.

3.6.2. SG 2 Seguir y controlar los cambios

Una vez liberada la línea base se debe seguir y controlar los productos de trabajo que se han creado en la gestión de configuración.

3.6.2.1. SP 2.1 Seguir las peticiones de cambio

El seguimiento de las peticiones de cambio se refiere a las modificaciones, incorporaciones de requisitos y fallos de los productos de trabajo. Las mismas sirven para medir el impacto que tendrán dichas modificaciones de los productos de trabajo para el proyecto.

En la plataforma de Eclipse (el editor de SPEM EPF Composer) existe en los elementos de contenidos del método como los roles, las tareas, los productos de trabajo, las guías y los artefactos, en la pestaña de descripción la información de la versión (*Version Information*) en ella se puede almacenar la siguiente información:

- Versión
- Fecha de cambio
- Descripción del cambio
- Autores
- Copyright o derechos de autor

En MOSKitt4ME, no son obligatorios ni tampoco están presentes en los procesos. Por lo que es posible realizar un seguimiento a las modificaciones de los elementos del método, pero no a los procesos, bien sea por fallos o nuevos requisitos que se realicen.

Valoración: Ampliamente implementada.

3.6.2.2. SP 2.2 Controlar los elementos de configuración

La práctica específica consiste en mantener la configuración de cada elemento de configuración, controlar la línea base, aprobar nuevas configuraciones y actualizar la línea base cuando se requiera.

En SPEM 2.0 mediante configuración de método, la cual permite identificar y mantener los elementos de la configuración, sin embargo, al no es posible crear las líneas bases, debido a que no maneja las primitivas de tiempo ni datos históricos, por lo tanto no se puede llevar a cabo un historial de revisiones de los elementos de configuración ni archivos de las líneas bases.

Valoración: Parcialmente implementada.

SG 3 Establecer la integridad.

Se debe permitir realizar las auditorías o controles de cambios de los elementos de configuración.

3.6.2.3. SP 3.1 Establecer los registros de gestión de configuración

Esta práctica específica indica que se debe establecer y dar soporte a los registros de los elementos de la configuración. Para ello se debe llevar un historial, registros de cambios o solicitudes de los diferentes elementos de la configuración.

Esta práctica específica indica que se debe establecer y dar soporte a los registros de los elementos de la configuración. Para ello se debe llevar un historial, registros de cambios o solicitudes de los diferentes elementos de la configuración.

Valoración: No implementada.

3.6.2.4. SP 3.2 Realizar auditorías de configuración

En MOSKitt4ME, se puede llevar a cabo un examen objetivo de un producto de trabajo frente a criterios específicos, como son los estándares o requisitos especificados. Por ejemplo, mediante la comprobación de la transformación de modelos, para el caso de modelo a modelo (M2M) o de modelo a texto (M2T), MOSKitt4ME pero no contempla el proceso de evaluación de la integridad de las líneas bases, ni la confirmación de los elementos completos, correctos y consistentes.

Valoración: No implementada.

3.7. Medición y análisis (MA)

Las prácticas específicas del área de proceso de medición y análisis consisten en determinar cuáles son los criterios de medición, análisis, especificación de medidas y técnicas

de análisis de recolección de datos, así como proveer los resultados para poder tomar las acciones correctivas en caso de que existieran desviaciones en relación con los requerimientos de información.

En MOSKitt4ME, no se puede llevar a cabo estas prácticas específicas debido a que no poseen atributos tangibles o medibles como es en el caso del tiempo, de los costos, o de un registro de fallos o defectos, etc.

Valoración: Sin puntaje.

3.8. Resumen

Una vez realizada la comparación entre las áreas de proceso del nivel de madurez 2 de CMMI-DEV 1.3, con MOSKitt4ME, se puede apreciar que muchas de las prácticas específicas son implementadas a completitud, mientras que otras no han sido implementadas. A continuación, se resume en una tabla los resultados obtenidos.

| Área de Proceso | Nro. Prácticas Específicas Implementada | | | | % Prácticas Específicas Implementada |
|---|---|--------|---------|----|--------------------------------------|
| | Completa | Amplia | Parcial | No | |
| Gestión de acuerdos con proveedores (SAM) | - | - | - | - | - |
| Gestión de requisitos (REQM) | 3 | 0 | 0 | 2 | 60 |
| Monitorización y control del proyecto (PMC) | 4 | 0 | 1 | 5 | 53 |
| Planificación del proyecto (PP) | 6 | 2 | 0 | 6 | 52 |
| Aseguramiento de la calidad del proceso y del producto (PPQA) | 1 | 1 | 0 | 2 | 42 |
| Gestión de configuración (CM) | 1 | 1 | 1 | 4 | 28 |
| Medición y análisis (MA) | 0 | 0 | 0 | 8 | 0 |

TABLA 3.1: Resumen comparativo entre CMMI-DEV vs. MOSKitt4ME - Nivel 2

En la tabla 3.1 se realiza una síntesis cuantitativa del análisis realizado a lo largo del presente capítulo. La tabla indica las áreas de proceso, el número total de prácticas específicas que son completamente implementadas, ampliamente implementadas, parcialmente implementadas y las no implementadas, y en la última columna el porcentaje de las prácticas específicas las cuales son implementadas por MOSKitt4ME.

En cuanto al área de proceso de gestión de acuerdos con proveedores (SAM), se puede indicar que no es aplicable para MOSKitt4ME, debido a que dentro de los propósitos o metas que posee el método no es posible de gestionar la adquisición de productos y servicios con los proveedores.

En relación a las áreas de procesos de gestión de requisitos (REQM), monitorización y control del proyecto (PMC), planificación del proyecto (PP), aseguramiento de la calidad del proceso y del producto (PPQA) y gestión de configuración (CM), es posible realizar mejoras al método o incluso aplicar funcionalidades ya existentes de SPEM 2.0. Por lo que estas áreas de proceso se consideran insatisfechas al no ser implementadas por CMMI-DEV al 100 %.

En cuanto al área de proceso medición y análisis (MA) se la considera sin puntaje, debido a que de las ocho prácticas específicas ninguna de ellas es soportada por MOSKitt4ME. Sin embargo, si se aplican mejoras al método incorporando atributos tangibles de medición y análisis, es posible que algunas prácticas específicas sean implementadas por CMMI-DEV.

Existen conceptos y términos que han surgido a lo largo del desarrollo de este capítulo y que se pueden resumir en este cuadro resumen; en algunos se puede ver una clara relación entre CMMI-DEV v1.3 y MOSKitt4ME. Si se desea ver un análisis más detallado se recomienda ver las prácticas específicas descritas anteriormente.

| CMMI-DEV | MOSKitt4ME (SPEM 2.0/BPMN 2.0) |
|--|---|
| Comprensión de requisitos | Definir requisitos como tareas; crear disciplinas para su comprensión |
| Trazabilidad bidireccional | Variabilidad \Rightarrow Contribución |
| Monitorización y control del proyecto: <ol style="list-style-type: none"> 1. Progreso, rendimiento y avance de la tareas 2. Datos 3. Involucración de las partes interesadas 4. Revisión del progreso 5. Revisión de hitos | Se aplica en la fase de aplicación del método: <ol style="list-style-type: none"> 1. El avance de actividades e hitos 2. Artefactos, documentos, códigos y archivos 3. Vistas de roles específicos 4. Vistas del método de proceso 5. Hitos |
| Estimar atributos de productos de trabajo | Métricas para la estimación |
| Fases del ciclo de vida | Fases \Rightarrow Hito |
| Estructuras de descomposición de trabajo (WBS) | Elementos de desglose de trabajo (WBE) |
| Diccionarios | Guías \Rightarrow Concepto |
| Planificación de los recursos | Diagramas de procesos |
| Informe de estado | Guías \Rightarrow Informe |
| Planificar conocimientos y habilidades | Cualificaciones y habilidades |
| Plan de proyecto | Vista consolidadas |
| Productos de trabajo | Definición de productos de trabajo: <ol style="list-style-type: none"> 1. Artefactos 2. Entregables 3. Resultados |
| Identificar elementos de configuración | Configuración del método |

TABLA 3.2: Resumen de conceptos entre CMMI-DEV vs. MOSKitt4ME - Nivel 2

Capítulo 4

Comparación de CMMI Vs

MOSKitt4ME - Nivel de Madurez 3

En el nivel de madurez 3 de integración de modelos de madurez y capacidades de desarrollo (CMMI-DEV) en la versión 1.3, los procesos están definidos, por lo que están debidamente establecidos y documentados, basados en propósitos claramente constituidos para el logro de sus objetivos.

A continuación, se realizará una comparación entre el nivel de madurez 3 de CMMI-DEV 1.3 y el marco metodológico para la construcción de métodos de producción de software, con el propósito de determinar el grado de implementación de las prácticas específicas con respecto a MOSKitt4ME.

4.1. Definición de procesos de la organización (OPD)

El área de proceso de definición de procesos de la organización gestiona el conjunto reutilizable de activos de proceso, los estándares del entorno de trabajo, y las reglas y guías para los equipos.

4.1.1. SG 1 Establecer los activos de proceso de la organización

Los activos de proceso de la organización consisten en los artefactos que contienen la descripción, implementación y mejora de los procesos. La meta específica tiene como propósito establecer y mantener este conjunto de activos de proceso de la organización.

4.1.1.1. SP 1.1 Establecer los procesos estándar

Los procesos estándar son los componentes de los procesos básicos que existen en cualquier proceso definido y su relación entre los componentes del proceso. Los procesos estándar pueden interconectarse a una o más arquitecturas de procesos e incluyen normalmente procesos técnicos, de gestión y organizativos.

Mediante SPEM 2.0, es posible definir los procesos estándar a través del patrón de capacidades (*Capability Pattern*), al poseer las propiedades de reusabilidad y aplicabilidad en la definición de procesos básicos. Éstas características antes mencionadas son esenciales para definir los procesos estándar. El patrón de capacidades representa un fragmento de proceso y puede ser reutilizado más de una vez en los procesos de despliegue.

Adicionalmente en MOSKitt4ME, ha definido los fragmentos del método que definen los elementos atómicos con el cual el método puede ser ensamblado. Estos fragmentos del método son almacenados en un repositorio del método para ser reusados como activos reutilizables. Siendo una colección organizada de roles, productos de trabajo, tareas, guías y procesos y fragmentos de método. El propósito es crear un repositorio de activos reutilizables con un formato estandarizado.

Valoración: Completamente implementada.

4.1.1.2. SP 1.2 Establecer las descripciones de los modelos de ciclo de vida

Los modelos de ciclo de vida se utilizan comúnmente para definir las fases del proyecto y se desarrollan para diferentes situaciones o clientes, debido a que no necesariamente se adecuan a todas las situaciones. MOSKitt4ME posee atributos que permiten especificar aspectos transitorios para los componentes del proceso, para que luego los atributos puedan estar asociados a los planes del proyecto. Por ejemplo, en los dos tipos especiales de actividades llamadas fase e iteración. La fase representa un ciclo de vida evolutivo, debido a que es un período de tiempo significativo para un proyecto, y, cuando finalice, representa un punto de gestión importante. La iteración representa la ejecución de una o varias descripciones de trabajo que se repiten más de una vez.

Valoración: Completamente implementada.

4.1.1.3. SP 1.3 Establecer los criterios y las guías de adaptación

Los criterios y las guías de adaptación indican:

- Cómo utilizar el conjunto de procesos estándar de la organización
- Los requisitos a satisfacer los procesos definidos
- Los procedimientos a seguir para documentar y realizar la adaptación del proceso
En SPEM 2.0 existe la guía (*Guidance*), un elemento del método, que provee información adicional y se relaciona con otros elementos. A manera de ejemplo, los tipos de predefinidos de guías son:
 - Las plantillas (*Template*) que establecen tablas de contenido, secciones y formatos estandarizados predefinidos de un artefacto
 - La práctica (*Practice*) que ayuda a definir estrategias predefinidas para elaborar un trabajo, resumiendo los aspectos importantes que impactan dentro de los procesos
 - La guía de herramienta (*Tool Mentor*) que explica el uso de una herramienta independiente al contexto actual de trabajo

Al contar MOSKitt4ME con estos elementos predefinidos de las guías es posible establecer los criterios y guías de adaptación del proceso.

Valoración: Completamente implementada.

4.1.1.4. SP 1.4 Establecer el repositorio de mediciones de la organización

El repositorio posee medidas de producto y de proceso que están relacionadas con los procesos estándar. El repositorio posee también la información requerida para poder comprender, evaluar e interpretar las medidas.

MOSKitt4ME utiliza el estándar RAS [18] para almacenar los fragmentos del método (los fragmentos conceptuales como las tareas, roles, productos y procesos). Las propiedades de estos fragmentos del método son: situación, tipo, origen y objetivo, siendo posible determinar la medida de esfuerzo de los productos a través de sus propiedades de repetición y múltiples ocurrencias. Sin embargo, entre dichas propiedades no encontramos el tamaño de los productos de trabajo y costos (horas/personas), medidas de fiabilidad o medidas de calidad, aunque son importantes ya que se utilizan frecuentemente para realizar las mediciones en las organizaciones.

Valoración: Parcialmente implementada.

4.1.1.5. **SP 1.5 Establecer la biblioteca de activos de proceso de la organización**

El diseño, los elementos elegidos y el catálogo de elementos establecen la biblioteca de los activos de procesos de las organizaciones.

La biblioteca de los activos de proceso es posible establecerse y mantenerse en MOS-Kitt4ME, a través de las guías o instrucciones (*Guidance*), debido a que son los elementos del método que provee información relacionada con otros componentes. Ejemplo de ello, son los activos reutilizable (*Reusable Asset*) que proporcionan información sobre el funcionamiento de los roles, cómo crear los productos de trabajo, cómo realizar las tareas, ayudas de procesos, entre otros.

Valoración: Completamente implementada.

4.1.1.6. **SP 1.6 Establecer los estándares del entorno de trabajo**

Los estándares del entorno de trabajo permiten a las organizaciones y a los proyectos favorecerse de las herramientas, formación y mantenimientos comunes. El método de desarrollo de software está basado en el desarrollo dirigido por modelos, en donde MOS-Kitt4ME es un CAME¹ que se desarrolló para dar soporte a la construcción de métodos de producción de software. Se fundamenta en los dos estándares SPEM 2.0 (para la construcción del método, fase de diseño del método) y BPMN 2.0 (en la mejora del proceso, específicamente en la ejecución del proceso). En la fase de implementación del método, MOSKitt4ME provee un entorno CASE² (logrado mediante una transformación M2T). Todo esto contribuye a crear un entorno de trabajo estandarizado para las organizaciones en la construcción de métodos.

Valoración: Completamente implementada.

4.1.1.7. **SP 1.7 Establecer las reglas y guías para los equipos**

Quiénes definen y controlan cómo se debe crear e interactuar para el logro común de los objetivos son las guías y reglas para los equipos. Al establecer estas reglas o guías se debe asegurar el cumplimiento de las normas, leyes o estándares que afecten su utilización por los equipos. Así como la definición de los estatutos de los equipos, sus miembros, los líderes y la asignación de los recursos para que cada equipo cumpla con su trabajo.

¹Computer-Aided Method Engineering; Ingeniería de Método Asistido por Computadora

²Computer-Aided Software Engineering, Ingeniería de Software Asistida por Computadora

En SPEM 2.0 existen los elementos, llamados cualificaciones (*Qualification*), que se definen en los roles o las tareas. Las cualificaciones o habilidades de los roles representan el conjunto de destrezas y capacidades que poseen en común un grupo o individuo. Las cualificaciones de las tareas son las destrezas requeridas para realizar una tarea en particular. Se puede definir y controlar las reglas mediante esta primitiva que posee SPEM 2.0, para el cumplimiento del trabajo.

Valoración: Completamente implementada.

4.2. Enfoque en procesos de la organización (OPF)

El área de enfoque en procesos de la organización cuenta entre sus objetivos con el plan de mejora de procesos de la organización. Para ello debe planificar, implementar y ejecutar mejoras en la organización, tomando en cuenta sus fortalezas y debilidades actuales, además de los activos de procesos de la organización.

4.2.1. SG 1 Determinar las oportunidades de mejora de procesos

Consiste en la identificación de forma periódica de las fortalezas, debilidades y oportunidades de los procesos de la organización.

4.2.1.1. SP 1.1 Establecer las necesidades de proceso de la organización

La práctica específica indica que se debe establecer y mantener la descripción de las necesidades de procesos. Los objetivos, necesidades y limitaciones del negocio afectan en forma directa a los procesos de la organización determinando sus objetivos y necesidades. Dentro del entorno del negocio que afecta a los procesos se puede mencionar la satisfacción del cliente, de las finanzas, la tecnología y los recursos humanos.

MOSKitt4ME contribuye a los procesos de la organización, dado que permite a los ingenieros de métodos crear métodos de desarrollo de software, incidiendo directamente en la tecnología de la organización y, por consecuencia, en sus finanzas.

Valoración: Parcialmente implementada.

4.2.1.2. SP 1.2 Evaluar los procesos de la organización

La evaluación de los procesos se debe realizar de forma periódica, manteniendo una comprensión de sus fortalezas y debilidades. El propósito de realizar la evaluación de los procesos es:

- Identificar las mejoras de procesos
- Confirmar el progreso a los beneficios de las mejoras
- Satisfacer las necesidades de una relación Cliente-Proveedor
- Motivar y facilitar que se involucren los procesos

En MOSKitt4ME, no se realiza una evaluación de los procesos de forma periódica.

Valoración: No implementada.

4.2.1.3. SP 1.3 Identificar las mejoras de procesos de la organización

Esta práctica específica requiere realizar un análisis e identificación de las mejoras a los procesos en la organización. Para el logro de dichas mejoras se debe determinar, identificar, priorizar y documentar las mejoras en los procesos candidatos que se implementarán.

En SPEM 2.0, los procesos organizativos se relacionan con los procesos de despliegue (*Delivery Process*) ayudando para la identificación de mejoras de los procesos. Sin embargo, la categorización y la priorización dependerán de las políticas de la empresa.

Valoración: Parcialmente implementada.

4.2.1.4. SG 2 Planificar e implementar las acciones de proceso

La meta específica, en primer lugar, establece los planes de acción de proceso, los cuales consisten en mejorar los procesos y los activos de procesos. Al involucrar a las partes interesadas y obtener el compromiso en las mejoras de proceso se incrementa la probabilidad de un desarrollo eficaz. Y en segundo lugar, implementa el establecimiento de los planes de acción de proceso.

En MOSKitt4ME, no se contempla el manejo de los planes de acción, debido a que no se realizan evaluaciones que permitan descubrir las debilidades y, por ende, no se pueden mejorar mediante la implementación de los planes de acción.

Valoración: No implementada.

4.2.2. SG 3 Desplegar los activos de proceso de la organización e incorporar las experiencias

La meta específica tiene como propósito expandir e incorporar las experiencias adquiridas a los activos de procesos en toda la organización.

4.2.2.1. SP 3.1 Desplegar los activos de proceso de la organización

Se debe garantizar el despliegue de los activos de proceso de forma organizada y su incorporación (según sea necesario) en algunas áreas de la organización.

En MOSKitt4ME, al contar con procesos estándar, modelos de ciclos de vida, criterios y guías de adaptación, bibliotecas de activos y estar basado en estándares como los “Procesos de Ingeniería de Software” (SPEM 2.0) y “Modelado y Notación de Procesos de Negocios” (BPMN 2.0), se garantizan los materiales de formación de despliegue y los materiales de soporte de los activos de procesos. No obstante, en dicho método no están contempladas la planificación del despliegue o la documentación de los cambios de los activos de procesos.

Valoración: Ampliamente implementada.

4.2.2.2. SP 3.2 Desplegar los procesos estándar

Los procesos estándar no sólo consisten en crearlos y desplegarlos, sino que deben ser actualizados de acuerdo con los cambios que puedan surgir durante el transcurso de vida de los proyectos. Estos procesos estándar deben ser adaptados y, con el transcurrir del tiempo, han de ser cada vez más eficaces, sobre todo en las actividades más críticas de los procesos.

En MOSKitt4ME, al definirse los procesos estándar mediante el patrón de capacidad, estos bien podrían ser adaptados, modificados o servir de base para crear unos patrones de capacidad, y así incluir los beneficios de los otros proyectos aprendidos.

Valoración: Completamente implementada.

4.2.2.3. SP 3.3 Monitorizar la implementación

Para determinar si los activos de procesos se adecuan y se acoplan a los proyectos, se requiere monitorizar la implementación y determinar cuáles son los activos de procesos que se están manejando y en dónde, dentro de la organización.

En MOSKitt4ME, una vez generado el ambiente CASE de forma automática - en la vista de proceso -, es posible realizar la monitorización, para un momento determinado, de los activos de procesos, siempre que se encuentren involucrados a los procesos que se están monitorizando. Mediante MOSKitt4ME, es factible cumplir con la práctica específica de monitorizar la implementación de los activos de procesos.

Valoración: Parcialmente implementada.

4.2.2.4. SP 3.4 Incorporar las experiencias en los activos de proceso de la organización

Los resultados que se han obtenido de la planificación y ejecución del proceso en cuanto a las experiencias concernientes al proceso se deben incorporar a los activos de procesos. Además, se debe registrar las actividades de mejoras de los procesos en la organización.

Para el cumplimiento de esta práctica específica sería necesario que MOSKitt4ME cumpla con:

- Contar con revisiones periódicas
- Obtener retroalimentación
- Listar y poner en práctica las lecciones aprendidas
- Contar con registros históricos de las actividades de los procesos

Valoración: No implementada.

4.3. Formación en la organización (OT)

El propósito de las prácticas específicas concernientes al área de formación en la organización consiste en aumentar las habilidades y destrezas de las personas para que puedan efectuar sus roles de una forma eficaz y eficientemente durante los proyectos. En su primera etapa, se debe establecer lo siguiente:

- Las necesidades estratégicas de formación
- Cuáles son las necesidades de formación que dependen de la organización
- El plan de formación táctico

- La capacidad de formación

Como última etapa, impartir, establecer los registros y evaluar la formación de forma eficaz.

El área de procesos de formación en la organización se considera no aplicable para MOSKitt4ME, debido a que el método no fue creado con propósitos formativos o de entrenamiento.

Valoración: No aplicable.

4.4. Gestión de riesgos (RSKM)

El área de gestión de riesgos implica reconocer los problemas más trascendentes y que pueden influir de forma significativa en el transcurso de la ejecución de los proyectos, con el propósito de mitigar el impacto que puedan tener, para lograr los objetivos del proceso.

En MOSKitt4ME, no se realizan las siguientes prácticas específicas:

- Determinar las fuentes y las categorías de riesgos
- Definir los parámetros
- Establecer una estrategia de gestión de riesgos
- Identificar, evaluar, clasificar y priorizar los riesgos
- Desarrollar los planes de mitigación de riesgo e implementarlos

Valoración: Sin puntaje.

4.5. Gestión integrada del proyecto (IPM)

El área de gestión integrada del proyecto tiene como objetivo la creación y gestión de los proyectos. Optimiza los recursos necesarios, alcanza el nivel de eficiencia y el nivel de eficacia de los beneficios previstos, contando para ello con un conjunto de procesos estándar definidos en la organización.

4.5.1. SG 1 Utilizar el proceso definido del proyecto

Mediante los procesos definidos se llevan a cabo los proyectos. Los procesos definidos - basados en los procesos estándar - tratan todos aquellos métodos necesarios para adquirir, desarrollar y mantener el producto.

4.5.1.1. SP 1.1 Establecer el proceso definido del proyecto

Los procesos definidos se basan en los requisitos de las partes interesadas, los compromisos y las necesidades de la organización, los procesos estándar, las guías de adaptación de la organización, así como su entorno operacional y de negocio. Por lo tanto, los procesos definidos se deben instaurar desde la fase de inicio de vida de los proyectos, como a lo largo de los mismos y deben estar integrados de forma coherente. Los procesos definidos traen como beneficios los planes del proyecto y permiten instaurar eficientemente el conjunto inicial de requisitos.

Se relacionan los procesos definidos con MOSKitt4ME mediante el proceso de despliegue (Delivery Process) que sirven de plantilla en la planificación y ejecución de los proyectos, abarcando un ciclo de vida integrado y completo. Los patrones de capacidades (asociados como procesos estándar) están acoplados a un proceso de despliegue, permitiendo de esta manera adaptar los procesos estándar para que se ajusten a las necesidades de los proyectos mediante el proceso definido (ver sección [4.1 SP 1.1 Establecer los procesos estándar](#)).

Valoración: Completamente implementada.

4.5.1.2. SP 1.2 Utilizar los activos de proceso de la organización para planificar las actividades del proyecto

Los resultados de las mediciones y los activos de procesos son utilizados para estimar los parámetros de planificación y evaluar las actividades del proyecto respectivamente.

En MOSKitt4ME, es posible utilizar los procesos de la organización para estimar las actividades del proyecto o estimar los parámetros de planificación, a través de los elementos de desglose de trabajo (Work Breakdown Element) mediante:

- Las guías
- La definición de fases e iteraciones
- Algunos parámetros de planificación como la medición del esfuerzo

Lo que no se puede estimar con el método son los parámetros de medición para la planificación del proyecto; por ejemplo: datos válidos históricos del proyecto, costo, calendario.

Valoración: Ampliamente implementada.

4.5.1.3. SP 1.3 Establecer el entorno de trabajo del proyecto

El entorno de trabajo se fundamenta en las instalaciones, las herramientas y los equipos que requieren las personas para el cumplimiento correcto de los objetivos del proyecto. Para que pueda existir un mantenimiento y establecimiento del ambiente de trabajo idóneo, dicho ambiente se debe basar en los estándares del entorno de trabajo.

En primer lugar, MOSKitt4ME se desarrolló con base en el resultado de un marco metodológico que combinaba las técnicas de ingeniería de métodos y el desarrollo dirigido por modelos bajo un entorno de ECLIPSE. En segundo lugar, MOSKitt4ME utiliza una serie de estándares ya preestablecidos - con base en SPEM 2.0 y BPMN 2.0 - para que los ingenieros de métodos puedan diseñar e implementar métodos para el desarrollo de software, permitiendo de esta manera establecer un entorno de trabajo adecuado para el desarrollo del proyecto.

Valoración: Completamente implementada.

4.5.1.4. SP 1.4 Integrar los planes

Los planes del proyecto se deben integrar a los otros planes que influyen en los proyectos y que describen los procesos definidos. Los otros planes se pueden considerar como las actividades adicionales de la planificación, como por ejemplo coordinar las partes interesadas relevantes, incorporar el proceso definido, establecer los criterios de entrada y salida de las tareas.

Los planes adicionales que se requieran en la planificación de los proyectos a través de los elementos de desglose de trabajo (*Work Breakdown Element*), relacionados previamente como procesos definidos en las prácticas específicas de CMMI-DEV, pueden integrarse con SPEM 2.0.

Valoración: Completamente implementada.

4.5.1.5. SP 1.5 Gestionar el proyecto utilizando planes integrados

La práctica específica tiene como objetivo gestionar el proyecto, por lo que se requieren el proceso definido, el plan y los otros planes que afecten al proyecto.

En SPEM 2.0, al contar con los procesos de despliegue como los procesos definidos y los planes como los elementos del desglose de trabajo, es posible realizar la gestión del proyecto mediante los planes integrados.

Valoración: Completamente implementada.

4.5.1.6. SP 1.6 Establecer los equipos

Los equipos se establecen tomando en consideración sus habilidades y experiencias, para que en conjunto puedan alcanzar los objetivos planteados en el proyecto. En los equipos se deben identificar las responsabilidades, los papeles que desempeñan en el proyecto, siendo preciso, para poder medir y gestionar los resultados que se puedan obtener como grupo de trabajo.

Todos estos aspectos se encuentran en SPEM 2.0 a través de las cualificaciones de los roles y las tareas, quienes pueden estar asociados con los productos de trabajo y las habilidades que provee el rol. Los roles representan a un grupo o un individuo que posee un conjunto de competencias y responsabilidades. Las tareas poseen la opción de las cualificaciones siendo las habilidades necesarias para desarrollar una tarea. Estas características de los roles y de las tareas hacen factible el establecimiento de los equipos.

Valoración: Completamente implementada.

4.5.1.7. SP 1.7 Contribuir a los activos de proceso de la organización

La contribución de información se realiza desde los procesos definidos del proyecto hasta los activos de procesos de la organización.

En SPEM 2.0 existe la posibilidad de realizar la documentación del proceso de despliegue (relacionado con el proceso definido) a través de las guías, por ejemplo las guías de comprobación (*Checklist*) o las descripciones de ejemplares de procesos (*Guideline*). SPEM 2.0 ejecuta el apoyo a la monitorización de la implementación (mediante la fase de implementación del método), específicamente a los artefactos que se encuentran asociados con los procesos estándar del método. Lo que no es posible realizar en SPEM 2.0 son las mejoras a las propuestas de los activos de procesos, ni proporcionar las medidas reales de los métodos y del producto en el proyecto.

Valoración: Ampliamente implementada.

4.5.2. SG 2 Coordinar y colaborar con las partes interesadas relevantes

La meta específica describe la coordinación y colaboración entre el proyecto y las partes interesadas.

4.5.2.1. SP 2.1 Gestionar la involucración de las partes interesadas

Se debe garantizar la gestión para el logro del compromiso de las partes interesadas en el desenvolvimiento del proyecto.

Las partes interesadas se relacionan con MOSKitt4ME mediante la definición de los roles (*Role Definition*), y las tareas se vinculan a unos pocos roles. Al tener asociados las tareas y los roles a los procesos de despliegue (como procesos definidos), es posible gestionar la involucración de las partes interesadas para un desarrollo correcto del plan de proyecto.

Valoración: Completamente implementada.

4.5.2.2. SP 2.2 Gestionar las dependencias

Se debe hacer un seguimiento, control e identificación de las dependencias críticas (de los recursos) con las partes interesadas del proyecto.

Mediante SPEM 2.0 no es posible determinar si existen dependencias críticas a lo largo de la planificación del proyecto, por lo que no se puede realizar una gestión de dependencias con las partes interesadas.

Valoración: No implementada.

4.5.2.3. SP 2.3 Resolver las cuestiones de coordinación

Resolver los asuntos con las partes interesadas relevantes. Estos asuntos pueden ser los defectos en los requisitos o los defectos en el diseño, los problemas a nivel de producto o la falta de disponibilidad de los recursos críticos para el proyecto.

Mediante MOSKitt4ME no existen procedimientos preestablecidos que canalicen para coordinar y dar una solución a los posibles asuntos arriba mencionados.

Valoración: No implementada.

4.6. Desarrollo de requisitos (RD)

El área de desarrollo de requisitos deduce, analiza y determina tres tipos de requisitos que se describen en esta área de clientes, de productos y de componentes del producto.

Los requisitos son la base de todo diseño, por lo que se considera que todos los proyectos de desarrollo tienen requisitos.

4.6.1. SG 1 Desarrollar los requisitos de cliente

Los requisitos del cliente representan las necesidades, expectativas, restricciones e interfaces de las partes interesadas.

4.6.1.1. SP 1.1 Educir las necesidades

La deducción de las necesidades implica la identificación de los requisitos que no han sido explícitamente proporcionados por el cliente.

Mediante MOSKitt4ME, no es posible educir las necesidades no proporcionadas por el usuario final.

Valoración: No implementada

4.6.1.2. SP 1.2 Trasformar las necesidades de las partes interesadas en requisitos de cliente

Implica realizar la transformación de las necesidades, las expectativas, las restricciones y las interfaces de las partes interesadas en requisitos de cliente priorizados.

En MOSKitt4ME, al no poder educir las necesidades, tampoco se puede realizar la priorización de los requisitos de cliente.

Valoración: No implementada

4.6.2. SG 2 Desarrollar los requisitos de producto

Los requisitos de clientes deben ser elaborados, analizados o clasificados para poder desarrollar los requisitos de productos y de componentes de productos.

Mediante MOSKitt4ME, es posible establecer, analizar y clasificar los requisitos de los productos o los componentes de los productos mediante las disciplinas.

Valoración: Completamente implementada.

4.6.3. SG 3 Analizar y validar los requisitos

El análisis y la validación se realizan con el propósito de determinar los requisitos conceptuales que optarán a satisfacer las necesidades expectativas y las restricciones de las partes interesadas, que luego se transformarán en requisitos funcionales.

4.6.3.1. SP 3.1 Establecer los conceptos y los escenarios de operación

Un escenario representa la secuencia de eventos que podrían ocurrir en el desarrollo o uso del producto. Dicho escenario se usa para describir las necesidades funcionales o de calidad. Un concepto operacional para un producto depende de la solución del diseño como del escenario. La práctica específica establece y mantiene estos conceptos.

En SPEM 2.0, se puede dar soporte mediante las primitivas de guías, para el escenario mediante la lista de comprobación (*Checklist*) que representa una serie de pasos a ser comprobados y el concepto mediante la documentación (*Whitepaper*) que es una versión especial del concepto que ha sido revisada y publicada.

Valoración: Completamente implementada.

4.6.3.2. SP 3.2 Establecer una definición de la funcionalidad y de los atributos de calidad requeridos

Define la funcionalidad y los atributos de calidad analizando los escenarios para la descripción del producto.

Mediante MOSKitt4ME, es posible la definición de los atributos de calidad y funcionalidad a través de los procesos de despliegue. Adicionalmente, se permite realizar diagramas de actividad, diagramas de caso de uso e incorporar el análisis orientado a objetos.

Valoración: Completamente implementada.

4.6.3.3. SP 3.3 Analizar los requisitos

Se analizan los requisitos para determinar si son necesarios y suficientes para cumplir con los objetivos. En la medida que se realiza el análisis de los requisitos, se debe comprender la relación entre los atributos de calidad, la funcionalidad de alto nivel y los requisitos de más alto nivel.

En proceso de analizar los requisitos dependerá sustancialmente del ingeniero de método y la organización o categorización de los requisitos del proyecto que haya realizado. A través de los patrones de capacidades, roles, tareas y guías, MOSKitt4ME le proporciona todas las primitivas necesarias para la definición y organización de los requisitos, bien sean funcionales o de calidad.

Valoración: Ampliamente implementada.

4.6.3.4. SP 3.4 Analizar los requisitos para conseguir un equilibrio

Consiste en analizar los requisitos para mantener un equilibrio entre las necesidades y las restricciones de las partes interesadas.

Al consistir en un proceso, el análisis dependerá del ingeniero de método para mantener este equilibrio.

Valoración: No implementada

4.6.3.5. SP 3.5 Validar los requisitos

La validación de los requisitos está relacionada con el área de validación, y se debe garantizar el funcionamiento correcto del producto, de acuerdo con lo previsto en el entorno del usuario final.

En MOSKitt4ME, es posible realizar el proceso de validación de los requisitos, bien sea mediante la creación de prototipos, demostraciones o simulaciones en el entorno CASE de métodos.

Valoración: Completamente implementada

4.7. Integración del producto (PI)

El área de integración del producto tiene como propósito garantizar que el producto sea ensamblado, funcione correctamente de acuerdo con los atributos de calidad requeridos y sea entregada debidamente.

4.7.0.6. SG 1 Prepararse para la integración del producto

La meta específica se refiere a la preparación del ambiente o al entorno donde se realizará la integración del producto, así como los procedimientos y criterios que se deben definir antes de efectuarse la integración.

4.7.0.7. SP 1.1 Establecer una estrategia de integración

El establecimiento y mantenimiento de una estrategia de integración consiste en una aproximación que describe, ensambla y evalúa los componentes del producto. Dichas estrategias de integración deben estar alineadas con las aproximaciones técnicas y la selección de las soluciones de diseño y sus componentes de producto.

Como MOSKitt4ME se basa en el estándar SPEM 2.0 se permite establecer las estrategias de integración de los productos y sus componentes del método con las aproximaciones técnicas y la selección de las soluciones técnicas que se puedan proponer, gracias al marco conceptual de SPEM 2.0 que provee los conceptos necesarios para modelar, documentar, presentar, publicar y realizar métodos y procesos de software.

Valoración: Completamente implementada.

4.7.0.8. SP 1.2 Establecer el entorno de integración del producto

El entorno mediante el cual se realiza la integración del producto se debe establecer y mantener bien sea que se haya adquirido o desarrollado. Puede incluir el equipamiento de pruebas, simuladores y dispositivos de grabación.

Actualmente, MOSKitt4ME está siendo soportado por el Centro de Investigación en Métodos de Producción de Software, de la Universidad Politécnica de Valencia, siendo una extensión de MOSKitt (<http://www.moskitt.org/>) que da soporte al trabajo de investigación.

Valoración: Completamente implementada.

4.7.0.9. SP 1.3 Establecer los procedimientos y los criterios de integración del producto

Establecimiento y mantenimiento de los criterios de los procedimientos para la integración del producto pueden incluir el número de iteraciones incrementales, el detalle de las pruebas y otras evaluaciones.

Los procedimientos y criterios pueden indicar:

- La disponibilidad de un componente
- La definición de los criterios de verificación de componentes de productos y su comportamiento
- El grado de simulación permitido o el entorno a utilizar para las pruebas de integración
- El calendario y las pruebas de ensamblaje, para evitar la aparición de retrasos y fallas de componentes

Mediante las guías que se definen en MOSKitt4ME es posible establecer los procedimientos y los criterios de integración de los productos. Un ejemplo para el tipo de guía es la práctica (*Practice*) que es una manera predefinida de hacer un trabajo, y que resume los aspectos que pueden impactar en diferentes partes de un método o proceso.

Valoración: Completamente implementada.

4.7.0.10. SG 2 Asegurar la compatibilidad de las interfaces

Se debe garantizar que las interfaces, tanto internas como externas, de los componentes de producto sean completas y compatibles.

4.7.0.11. SP 2.1 Revisar la completitud de las descripciones de las interfaces

Las interfaces deben incluir la cobertura y completitud con sus descripciones adecuadas. Se debe garantizar que los componentes de producto y las interfaces sean etiquetadas para asegurar una conexión fácil y correcta para la unión de los componentes de producto.

MOSKitt4ME provee soporte al uso de herramientas externas del contexto de Eclipse. Para ello, SPEM 2.0 utiliza el tipo de elemento llamado guía de herramienta (*Tool Mentor*), que permite explicar el uso de ciertas herramientas en el contexto de un trabajo

o de forma independiente. Con el uso de la guía de herramienta es posible realizar las descripciones adecuadas de las interfaces.

Valoración: Completamente implementada.

4.7.0.12. SP 2.2 Gestionar las interfaces

Mantener la consistencia de las interfaces durante toda la vida del producto, cumplir con las limitaciones de la arquitectura, solucionar los conflictos, cambios y las no conformidades están en la responsabilidad de la gestión de las interfaces. Por ello, se deben considerar las definiciones y diseños de las interfaces internas y externas, para los productos y componentes de productos.

La interfaz externa en MOSKitt4ME se define a través de las guías de herramientas. La interfaz interna se define como la relación que se mantiene entre la tarea y los productos de trabajo. Una tarea está asociada a los productos de trabajo como salidas y entradas obligatorias u opcionales. Los productos de trabajo están predefinidos como:

- Artefactos (*Artifact*): pueden ser modelos, documentos, códigos, archivos, etc.
- Entregables (*Deliverable*): están asociados con cero o más de un componente entregable
- Resultados (*Outcome*): son de naturaleza tangible

Al existir estas especializaciones de los productos de trabajo en SPEM 2.0 es posible que un producto de trabajo de salida de una tarea pueda ser el producto de trabajo de entrada de otra tarea. Por esta razón, se considera a las relaciones entre las tareas y los productos de trabajo como las interfaces internas.

Por lo anteriormente expuesto, se considera posible gestionar en MOSKitt4ME las interfaces, tanto internas como externas, de los productos y los componentes de los productos.

Valoración: Completamente implementada.

4.7.1. SG 3 Ensamblar los componentes de producto y entregar el producto

Los componentes de producto deben estar ensamblados y verificados. Además, se debe proporcionar un producto integrado, verificado y validado.

4.7.1.1. SP 3.1 Confirmar la disponibilidad de los componentes de producto para la integración

Se debe verificar antes de realizar la integración de los componentes de producto si estos han sido identificados de forma correcta, si su comportamiento es de acuerdo con lo descrito y si las interfaces del componente cumplen con las descripciones que poseen.

En MOSKitt4ME, a través de la ejecución del método, en la fase de implementación, es posible realizar las evaluaciones de los componentes de los productos. Sin embargo en la fase de diseño, no existe un proceso que permita determinar en forma automática la verificación de los componentes de producto y así determinar si cumplen o no con la funcionalidad que se ha descrito.

Valoración: Parcialmente implementada.

4.7.1.2. SP 3.2 Ensamblar los componentes de producto

La práctica específica tiene como propósito ensamblar los componentes de producto de acuerdo con los procedimientos de integración.

Los componentes de producto en SPEM 2.0 representan el contenido de método (*Method Content*), y el proceso de ensamblaje de dichos componentes se realiza en el marco metodológico de los procesos (*Processes*) donde se reutilizan y combinan dichos elementos.

Valoración: Completamente implementada.

4.7.1.3. SP 3.3 Evaluar los componentes de producto ensamblados

Los componentes de producto, una vez ensamblados, deben someterse a una validación en cuanto a la compatibilidad de las interfaces.

Se considera parcialmente implementada la práctica específica de evaluación debido a que no se realizan procesos de validación formales en MOSKitt4ME. No obstante, existe la vista consolidada (*Consolidated View*) en los procesos ya ensamblados que permite realizar una evaluación visual de la jerarquía o la arquitectura de producto.

Valoración: Parcialmente implementada.

4.7.1.4. SP 3.4 Empaquetar y entregar el producto o componente de producto

La entrega y empaquetado del producto o componente de producto al cliente.

Esta práctica específica se relaciona con MOSKitt4ME mediante la fase de implementación del método que consiste en la generación de forma semiautomática de un entorno CASE que provee soporte a la creación de productos de método y también a la ejecución del proceso.

Valoración: Completamente implementada.

4.8. Solución técnica (TS)

El área de solución técnica se focaliza en la selección de la solución, el desarrollo detallado y la implementación del diseño. Las soluciones técnicas engloban los productos, los componentes de producto y los procesos del ciclo de vida relativos al producto, bien individualmente o en conjunto.

4.8.1. SG 1 Seleccionar soluciones de componentes de producto

Las soluciones alternativas y sus ventajas relativas son parte de la solución del componente, en donde las soluciones alternativas son establecidas mediante los requisitos clave, los asuntos de diseño y las limitaciones.

4.8.1.1. SP 1.1 Desarrollar soluciones alternativas y los criterios de selección

La identificación de soluciones alternativas permite la selección de una solución equilibrada en términos de coste, de calendario, de rendimiento y de riesgo. Las soluciones alternativas y los criterios de selección se basan en:

- Las arquitecturas de productos de trabajos que posean atributos críticos de calidad
- Un diseño factible de soluciones
- La asignación de alternativas de requisitos para diferentes componentes de producto
- Un rango aceptable de coste, de calendario y de rendimiento

Normalmente podrían tratar costes (p. ej., tiempo, personal, dinero), beneficios (p. ej., prestación del producto, capacidad, eficacia) y riesgos (p. ej., técnicos, de coste, de calendario).

Mediante SPEM 2.0 se pueden proporcionar las soluciones alternativas y los criterios de selección mediante el concepto de variabilidad (*Variability*). Esta última se organiza en plug-ins (*Plugins*) en donde los procesos incluyen partes alternativas que son configurables, y es un mecanismo que permite modificar los elementos de método o de proceso sin realizar la modificación del componente original. A los criterios de selección se puede dar soporte mediante los tipos de guías llamadas instrucción (*Guideline*) que proveen información adicional para la realización de un grupo de tareas y recomendación sobre productos de trabajo.

Como MOSKitt4ME se basa en el estándar SPEM 2.0, se puede incluir los criterios de selección. Sin embargo, para la identificación de soluciones alternativas, el método no posee las primitivas del calendario o de la gestión de riesgos.

Valoración: Ampliamente implementada.

4.8.1.2. SP 1.2 Seleccionar las soluciones de componentes de producto

Con base en los criterios de selección previamente definidos, se debe seleccionar los componentes de productos. Debe existir una relación entre los requisitos y los componentes de productos, en donde la decisión de la solución y evaluación deben estar debidamente documentadas e incluir un análisis razonado.

En SPEM 2.0, una vez identificado los componentes de producto dentro de la fase de contenido de método (*Method Content*) y establecido los criterios de selección, con el uso de las guías (*Guidance*), se puede establecer la selección de las soluciones y su debida documentación de la decisión mediante el uso de las guías, ya que sirven de soporte o ayuda a los demás elementos del método. Así, por ejemplo, es el caso del tipo de guía llamada informe (*Report*) que es una plantilla predefinida de un resultado que se obtiene de forma automática.

Valoración: Completamente implementada.

4.8.2. SG 2 Desarrollar el diseño

El desarrollo del diseño debe ser apropiado para todas las fases del ciclo de vida del producto que incluye la modificación, el reaprovisionamiento, el mantenimiento, el soporte

y la instalación. Los diseños de producto deben contar también con una documentación apropiada que den referencia para la comprensión del diseño y mantenimiento de sus componentes.

4.8.2.1. SP 2.1 Diseñar el producto o los componentes de producto

La fase del diseño del producto comprende dos grandes fases: el diseño preliminar y el diseño detallado. El diseño preliminar establece las capacidades y la arquitectura del producto. El diseño detallado define por completo el detalle de la estructura y las capacidades de los componentes de producto.

SPEM 2.0 es un estándar de metamodelado que sirve para representar procesos de ingeniería de software, un metamodelo describe un conjunto de conceptos genéricos y sus interrelaciones para la definición de modelos de un cierto dominio. A través de la definición del metamodelo se define la primera fase para la arquitectura del diseño preliminar. Y, mediante la ejecución del método, MOSKitt4ME brinda todas las directrices que permiten el desarrollo de un diseño detallado del método.

Valoración: Completamente implementada.

4.8.2.2. SP 2.2 Establecer un paquete de datos técnicos

Los paquetes de datos técnicos son la descripción adecuada de un producto o de los componentes de un producto que sirve de apoyo a una estrategia de adquisición, o fase de implementación, producción, ingeniería y soporte logístico del ciclo de vida del producto. Es particularmente útil usar la arquitectura del producto como un medio para organizar estos datos, y para abstraer vistas que son claras y relevantes para una característica de interés.

En SPEM 2.0 los paquetes de datos técnicos se pueden establecer mediante las guías en el tipo de definición de término (*Term Definition*) que sirve para generar una especie de glosario automático y se relaciona a los elementos de contenido.

Valoración: Completamente implementada.

4.8.2.3. SP 2.3 Diseñar las interfaces usando criterios

Mediante los criterios establecidos se debe diseñar la interfaz de componentes.

En SPEM 2.0, al poderse desarrollar las interfaces en sus componentes y definir los criterios mediante las guías, es posible aplicar los criterios que se hayan establecido, sobre todo con el tipo de guía llamada práctica (*Practice*).

Valoración: Completamente implementada.

4.8.2.4. SP 2.4 Realizar los análisis sobre si hacer, comprar o reutilizar

El proceso de evaluación sobre si hacer, comprar o reutilizar, se basa en las necesidades del proyecto y depende de muchos factores, por ejemplo:

- Funcionalidad de los productos y su ajuste al proyecto
- Recursos y habilidades disponibles del proyecto
- Costes de adquisición frente a costes de desarrollo interno
- Fechas críticas de integración y de entrega
- Funcionalidad y calidad de los productos disponibles
- Impacto en las competencias básicas
- Licencias, garantías, responsabilidades y limitaciones asociadas con los productos que están siendo adquiridos
- Disponibilidad del producto
- Reducción del riesgo
- Correspondencia entre las necesidades y los activos básicos de la línea de producto

Mediante SPEM 2.0 es posible realizar la evaluación y el uso de los componentes de productos bien se haga, compre o se reutilice, dado que da el soporte requerido.

Valoración: Completamente implementada.

4.8.3. SG 3 Implementar el diseño del producto

Una vez realizado el diseño de los productos, se procede a implementarlos. Se debe incluir las pruebas unitarias de los componentes de producto antes de realizar la integración del mismo.

4.8.3.1. SP 3.1 Implementar el diseño

La actividad de implementación de los componentes de producto parte del diseño e incluye la asignación, el refinamiento, la coordinación de los esfuerzos de desarrollo y la verificación de cada componente del producto.

Mediante MOSKitt4ME, se puede realizar la implementación de diseño por las siguientes razones:

- Se realiza una asignación entre los componentes en la fase de diseño del método, por ejemplo, cuando se establece la relación entre los artefactos, las tareas y las guías.
- El proceso de refinamiento se realiza mediante la configuración del método, que consiste en una selección de contenidos de los plug-ins de una librería, de forma que se limita la vista de la librería a sub-conjuntos seleccionados.
- La coordinación de los esfuerzos de desarrollo se hace bajo un ambiente de desarrollo de ECLIPSE que está unificado y estandarizado.
- La verificación se lleva a cabo de forma manual o mediante la documentación que poseen los componentes de productos. Aunque quizás sea posible mejorar dicho proceso.

Valoración: Ampliamente implementada.

4.8.3.2. SP 3.2 Desarrollar la documentación de soporte del producto

La práctica específica requiere el desarrollo y mantenimiento de la documentación que se elabora en la instalación, operación y mantenimiento del producto.

SPEM 2.0 cuenta con los descriptores en todos sus componentes de productos que se desarrollan. Adicionalmente, posee elementos como las guías que le permiten incluir la documentación suplementaria necesaria a lo largo del ciclo de vida del proyecto.

Valoración: Completamente implementada.

4.9. Validación (VAL)

La validación significa que un producto cumple con su uso deseado, es decir, la validación asegura que se «construyó lo correcto». En la mayoría de los casos, los usuarios

finales y las partes interesadas están involucrados en las actividades de validación. Estas actividades se aplican en todos los aspectos del producto y en cualquier entorno previsto.

4.9.1. SG 1 Preparar la validación

Consiste en preparar el entorno, seleccionar los productos o los componentes de producto, crear los procedimientos y los criterios de validación. Cualquier producto o componente de producto puede estar sujeto a validación.

4.9.1.1. SP 1.1 Seleccionar los productos para la validación

Se puede seleccionar los requisitos, los productos y los componentes de productos para realizar la validación con relación a las necesidades del usuario final. Se deben crear los métodos, requisitos y las restricciones a considerar para realizar el proceso de validación en esta práctica específica.

En MOSKitt4ME, es factible realizar la selección de los requisitos, los productos y de los componentes, y con los procesos de despliegue (*Delivery Processes*) se llevaría a cabo la creación de los métodos, las restricciones y los requisitos del proceso de validación.

Valoración: Completamente implementada.

4.9.1.2. SP 1.2 Establecer el entorno de validación

Es el ambiente donde se realizan las validaciones de los diseños, los prototipos y de la versión final. El entorno de validación se debe establecer y mantener a lo largo del ciclo de vida del proyecto.

MOSKitt4ME permite la creación de prototipos, por lo tanto, se puede realizar las validaciones con el usuario final, así como también en la fase de ejecución del método (Ambiente CASE) ejecutar las validaciones.

Valoración: Completamente implementada.

4.9.1.3. SP 1.3 Establecer los procedimientos y los criterios de validación

Son los procedimientos, criterios y estándares de validación que se definen para asegurar que el producto cumpla con los objetivos que se hayan definido para su uso. Incluyen las pruebas y evaluaciones.

Los procedimientos y los criterios de validación se pueden establecer en SPEM2.0 mediante las guías (*Guidance*), específicamente con el tipo de guía llamada práctica (*Practice*).

Valoración: Completamente implementada.

4.9.2. SG 2 Validar el producto o los componentes de producto

Para realizar la validación del producto o los componentes de producto se usa los métodos, los procedimientos y los criterios de validaciones que se han definido previamente.

4.9.2.1. SP 2.1 Realizar la validación

Se ejecuta el proceso de validación de los productos o componentes con los usuarios o partes interesadas, de acuerdo con los procedimientos preestablecidos.

En MOSKitt4ME el procesos de validación se puede realizar en la fase de ejecución del método, creando previamente procesos de validación y ejecutándolos en esta fase (mediante la herramienta CASE) para así comprobara si los procesos de desarrollo de software cumplen con lo preestablecido. Sin embargo, el proceso de validación se debe realizar en todas las fases del ciclo de vida del producto de trabajo, y, durante la fase del diseño del método, no se realiza el proceso de validación.

Valoración: Ampliamente implementada.

4.9.2.2. SP 2.2 Analizar los resultados de la validación

Los resultados de las pruebas se analizan frente a los criterios de validación definidos. Los informes de análisis por lo general contienen:

- Si cumplieron con las necesidades
- Cuando ocurren las deficiencias, documentan el grado de éxito o fallo y clasifican las causas del fallo
- Si los malos resultados de las pruebas son debido a problemas de procedimiento de validación o del entorno.

Con MOSKitt4ME, es posible realizar el análisis de los resultados mediante la ayuda de las guías para los resultados, específicamente el tipo de guía llamada informe (*Report*).

Valoración: Completamente implementada.

4.10. Verificación (VER)

La verificación es la confirmación de que los productos reflejan los requisitos conforme se han especificado, es decir la verificación asegurar que «lo construyó correctamente». Al aplicar la verificación de los productos de trabajo es directamente proporcional de que el producto satisfaga los requisitos del cliente, de producto y de componente de producto. El área específica de verificación incluye a los productos y de los productos de trabajo intermedio, los requisitos de cliente, de producto y de componente de producto frente a los requisitos seleccionados.

4.10.1. SG 1 Preparar la verificación

La preparación de la verificación consiste en la selección de los productos a verificar, la preparación del entorno y la creación de los procedimientos a utilizar durante el proceso de verificación que incluye todo el ciclo de vida de producto o los componentes de producto de trabajo.

4.10.1.1. SP 1.1 Seleccionar los productos de trabajo para la verificación

La práctica específica de procesos implica la selección de los productos de trabajo y métodos para la verificación. Para seleccionar los componentes se toma en consideración su contribución con el logro de los objetivos y los requisitos del proyecto.

En MOSKitt4ME, es factible realizar la selección de los productos de verificación, debido a que se pueden seleccionar los componentes y crear los métodos de verificación mediante los procesos de despliegue (*Delivery Processes*).

Valoración: Completamente implementada.

4.10.1.2. SP 1.2 Establecer el entorno de verificación

El entorno de trabajo se puede adquirir, desarrollar, reutilizar, modificar o ser una mezcla de todos. Esto dependerá de las necesidades del proyecto. Algunos entornos pueden requerir simuladores, emuladores, generadores de escenarios, herramientas de reducciones de datos, interfaces con otros sistemas o simplemente los revisores y una sala (para el caso de revisores entre pares; ver definición [A.6](#)).

Es posible establecer entornos de trabajo mediante MOSKitt4ME, ya que al estar desarrollado en la plataforma de un entorno ECLIPSE³, este permite la integración o implementación pública, abierta y gratuita con otros sistemas.

Valoración: Completamente implementada.

4.10.1.3. SP 1.3 Establecer los procedimientos y los criterios de validación

Los procedimientos y los criterios de verificación se definen para asegurar que los productos de trabajo cumplen sus requisitos.

Los procedimientos y los criterios de verificación se pueden establecer en SPEM2.0 mediante las guías (*Guidance*), específicamente con el tipo de guía llamada práctica (*Practice*).

Valoración: Completamente implementada.

4.10.2. SG 2 Realizar las revisiones entre pares

La revisión entre pares es un método usado para la validación de la calidad de un trabajo, por lo general entre autores de rango semejante o superior al autor o creador, en este caso de los productos desarrollados en el proyecto. La realización de las revisiones entre pares (ver definición A.6) es altamente recomendada entre las prácticas específicas, debido a que es un mecanismo probado para eliminar defectos eficazmente, de tal forma que se pueden prevenir los defectos y se pueden identificar las oportunidades de mejoras.

En la metodología descrita por MOSKitt4ME, no se contempla el uso de este método de revisión entre pares para la verificación de los productos desarrollados. Sin embargo, su uso y aplicación se consideran factibles y consisten en identificar al personal que participará en la revisión entre pares, realizar el proceso de revisión con el propósito de encontrar y eliminar defectos en fases tempranas, y analizar los resultados de los datos obtenidos.

Valoración: No implementada.

4.10.3. SG 3 Verificar los productos de trabajo seleccionados

Verificar los productos y productos de trabajo incrementalmente promueve la detección temprana de problemas y puede dar como resultado la eliminación de defectos a tiempo.

³Más información se puede encontrar en: www.eclipse.org/epf/. Consultado al 08-05-2013

Los resultados de la verificación ahorran un coste considerable de aislamiento de fallos y re-trabajo asociados con la resolución de problemas.

Realizar el proceso de verificación de acuerdo con los métodos y procedimientos establecidos en los requerimientos.

4.10.3.1. SP 3.1 Realizar la verificación

Realizar la verificación sobre los productos de trabajo seleccionados. Tiene como beneficio la detección de los problemas si se aplica desde las primeras fases del proceso ahorrando costes y re-trabajos asociados a la resolución de problemas. Por lo general, la verificación del desarrollo de software incluye prototipado, modelado y simulación para verificar la idoneidad del diseño del sistema (y la asignación).

En MOSKitt4ME, se puede realizar la verificación por las siguientes razones:

- Se pueden crear prototipos
- Su objetivo principal es el diseño y desarrollo de métodos de desarrollo de software
- Se permite la ejecución de proyectos (con uso de herramientas CASE), con ello se puede realizar la simulación, de ser requerido

No obstante, el proceso de verificación se debe realizar en todas las fases del ciclo de vida del producto de trabajo, en la fase de diseño del método para MOSKitt4ME, no se realiza la verificación.

Valoración: Ampliamente implementada.

4.10.3.2. SP 3.2 Analizar los resultados de la verificación

El análisis de los resultados se debe comparar con los criterios establecidos previamente determinando su validez y registrar para guardar como evidencia del proceso de verificación.

Con MOSKitt4ME, es posible realizar el análisis de los resultados de la verificación con respecto a la ejecución de los otros procesos, mediante la ejecución del método, con la generación de la herramienta CASE, y se pueden apoyar mediante la ayuda de las guías, específicamente el tipo de guía llamada informe (*Report*).

Valoración: Completamente implementada.

4.11. Análisis de decisiones y resolución (DAR)

El área de análisis de decisiones y resoluciones consiste en establecer guías y poder determinar mediante un proceso de evaluación formal las posibles decisiones, que evaluarán las alternativas frente a unos criterios establecidos. El proceso de evaluación formal es un enfoque estructurado e implica las siguientes acciones:

- Establecer los criterios para evaluar alternativas
- Identificar soluciones alternativas
- Seleccionar métodos para evaluar alternativas
- Evaluar soluciones alternativas utilizando los criterios y los métodos establecidos
- Seleccionar soluciones recomendadas a partir de las alternativas identificadas con base en los criterios de evaluación

Se establecen las evaluaciones formales para cuestiones que lo requieran, por ejemplo la selección entre distintas alternativas de diseño o de arquitectura, la utilización de componentes reutilizables o de componentes de productos comerciales (COTS), los entornos de prueba, las alternativas de entrega.

Para MOSKitt4ME, se considera no aplicable esta área de análisis de decisiones y resolución, debido a que no es requerido establecer guías o procesos formales que evalúen alternativas de diseño o arquitectura.

Valoración: No aplicable.

4.12. Resumen

Una vez realizado el análisis comparativo entre las áreas de proceso del nivel de madurez 3 de CMMI-DEV 1.3 con MOSKitt4ME, se puede apreciar que muchas de las prácticas específicas son implementadas a completitud, mientras muy pocas no han sido implementadas. A continuación, se resumen los resultados obtenidos.

En tabla 4.1 se encuentra sintetizado el análisis que se efectuó a lo largo del presente capítulo. La tabla está comprendida por las siguientes columnas:

- “Área de Proceso” comprende el nivel de madurez 3 de CMMI-DEV

| Área de Proceso | Nro. Prácticas Específicas Implementadas | | | | % Prácticas Específicas Implementadas |
|---|--|--------|---------|----|---------------------------------------|
| | Completa | Amplia | Parcial | No | |
| Definición de procesos de la organización (OPD) | 6 | 0 | 1 | 0 | 90 |
| Enfoque en procesos de la organización (OPF) | 2 | 1 | 2 | 4 | 37 |
| Formación en la organización (OT) | -- | -- | -- | -- | -- |
| Gestión de riesgos (RSKM) | 0 | 0 | 0 | 7 | 0 |
| Gestión integrada del proyecto (IPM) | 6 | 2 | 0 | 2 | 73 |
| Desarrollo de requisitos (RD) | 6 | 1 | 0 | 3 | 67 |
| Integración del producto (PI) | 7 | 0 | 2 | 0 | 85 |
| Solución técnica (TS) | 6 | 2 | 0 | 0 | 92 |
| Validación (VAL) | 4 | 1 | 0 | 0 | 93 |
| Verificación (VER) | 4 | 1 | 0 | 3 | 58 |
| Análisis de decisiones y resolución (Dar) | -- | -- | -- | -- | -- |

TABLA 4.1: Resumen comparativo entre CMMI-DEV vs. MOSKitt4ME - Nivel 3

- En “Nro. Prácticas Específicas Implementadas”, el número de prácticas específicas que han sido implementadas completamente, ampliamente, parcialmente o no implementadas, de acuerdo con el estudio realizado
- “ % Prácticas Específicas Implementadas” da el porcentaje del número de prácticas específicas que se consideran implementadas en relación con el total de prácticas que posee el área de proceso. Este porcentaje se obtuvo mediante la siguiente fórmula:

$$(\sum, PesosrelacionPracticasEspecificas) / \text{Número de Prácticas Específicas}$$

En cuanto a las áreas de formación en la organización (OT) y al análisis de decisiones y resolución (DAR), se puede indicar que no son aplicables para MOSKitt4ME, debido a que no están dentro de los propósitos o metas que posee el método.

En relación con las áreas de procesos de definición de procesos de la organización (OPD), gestión integrada del proyecto (IPM), integración del producto (PI), solución técnica (TS), validación (VAL) y verificación (VER), se consideran que han alcanzado casi al 100 % la implementación de las prácticas específicas con respecto al método. De estas 42 prácticas evaluadas, sólo estos puntos que se mencionan de las prácticas específicas no se han implementado o se consideran parcialmente implementada por MOSKitt4ME:

- Repositorio de mediciones: específicamente las propiedades de tamaño de los productos de trabajo y costos (horas/personas), medidas de fiabilidad o medidas de calidad

- Resolver las cuestiones de coordinación: dar garantía de los defectos en los requisitos o los defectos en el diseño, los problemas a nivel de producto o la falta de disponibilidad de los recursos críticos para el proyecto
- Confirmar la disponibilidad de los componentes de producto para la integración: determinar en forma automática la verificación de los componentes de producto y así determinar si cumplen o no con la funcionalidad acordada
- Coordinación entre pares: un método de verificación del producto
- Verificación: la verificación automática del producto o los componentes del producto en todas las fases del ciclo de vida.

En cuanto al área de gestión de riesgos (RSKM), desarrollo de requisitos (RD) y enfoque en procesos de la organización (OPF), se consideran sin puntaje o con un bajo puntaje, debido a que ninguna de sus prácticas específicas o muy pocas son soportadas por MOSKitt4ME. Sin embargo, en su mayoría son procedimientos para la gestión de los riesgos, de los requisitos y de los procesos de la organización por lo que mediante SPEM 2.0, al ser un estándar de metamodelado, les permitirá representar e implementar sus procedimientos y procesos de ingeniería de software enmarcados en estas áreas.

A continuación, se realizó una tabla resumen de algunos conceptos o términos que han surgido a lo largo del presente capítulo en donde se ha encontrado una relación entre MOSKitt4ME y CMMI-DEV 1.3 (ver la tabla 4.2). Si se desea ver mayor detalle se pueden consultar las prácticas específicas en donde han sido implementadas con respecto al método.

El estándar de SPEM 2.0 y, por ende, MOSKitt4ME provee primitivas que ayudan con el cumplimiento de madurez del nivel 3 de CMMI-DEV, debido a que poseen características que se han analizado a lo largo de este capítulo, confirmando lo indicado en la especificación de SPEM 2.0. [14]

- La primera característica son los mecanismos de adaptación y reutilización de sus elementos, por ejemplo, los patrones de procesos o capacidades.
- Otro elemento descrito es la variabilidad o adaptabilidad de los elementos del método. Las guías y sus diferentes tipos sirven de ayuda en lo que respecta a la documentación, la definición del glosario de términos, la definición de los procedimientos para los demás elementos.
- Los elementos de descomposición de trabajo se relacionan con la creación de procesos definidos.

| CMMI-DEV | MOSKitt4ME (SPEM 2.0/BPMN 2.0) |
|--|--|
| Procesos estándar | Patrón de capacidades, fragmentos del método (activos reutilizables) |
| Modelos de ciclo de vida | Fases e iteraciones |
| Guías de adaptación | Guías ⇒ Plantillas |
| Bibliotecas de activos | Guías ⇒ Activos reutilizable |
| Reglas y guías de equipos | Cualificaciones |
| Procesos definidos | Procesos de despliegue |
| Interfaces internas | Relaciones entre las tareas, roles, artefactos |
| Interfaces externas | Guías ⇒ Guía de herramienta |
| Todos los procedimientos | Guías ⇒ Práctica |
| Ensamblar los componentes del producto | En el proceso de SPEM 2.0 |
| Soluciones alternativas | Variabilidad |
| Criterios de selección | Guías ⇒ Directrices |
| Refinamiento | Configuración del método |
| Entorno | Lo establece la plataforma ECLIPSE |

TABLA 4.2: Resumen de conceptos entre CMMI-DEV vs. MOSKitt4ME - Nivel 3

Todo lo anteriormente mencionado es fundamental a lo largo del establecimiento del mapeo entre CMMI-DEV y MOSKitt4ME.

Capítulo 5

Propuestas

En el análisis comparativo realizado entre las prácticas específicas del nivel 2 y 3 de madurez de CMMI-DEV y MOSKitt4ME, se pudo apreciar que existen prácticas específicas que no han sido implementadas por MOSKitt4ME. Las propuestas que a continuación se describe proponen alternativas o soluciones que tienen como objetivo mejorar MOSKitt4ME para que de soporte a las diferentes áreas de procesos, que se han mencionado en el nivel 2 y 3 de CMMI-DEV.

5.1. Incorporación de un plan de proyecto

Al no implementar por completo las prácticas específicas del área de Planificación del Proyecto (PP) se propone una extensión en MOSKitt4ME que se basa en la incorporación de un plan de proyecto. Para ello, SPEM 2.0 ofrece diferentes mecanismos para la ejecución de los procesos del método:

- Se mapean los procesos a planes del proyecto, y se realiza la ejecución de los planes mediante herramientas de gestión de proyectos, tales como IBM Rational Portfolio Manager o Microsoft Project.
- Se mapean los procesos a representaciones ejecutables, por ejemplo BPMN 2.0, para luego ejecutar la representación de los procesos mediante un motor de proceso implementadas de las prácticas.

Se debe analizar la primera opción de ejecución de proyectos mediante la herramienta de gestión de proyectos, con el propósito de poder establecer las equivalencias conceptuales entre SPEM 2.0 y MS Project 2010, para dar un mayor porcentaje de implementación

a las prácticas específicas y así dar por satisfecha el área de Planificación del Proyecto (PP).

A continuación se realizará un mapeo entre SPEM 2.0 y Microsoft Project 2010:

- En SPEM 2.0 los procesos de entrega (*DeliveryProcess*), describen el enfoque completo e integrado para la realización de un tipo de proyecto específico. En MS Project 2010 este concepto se representa por medio del archivo del proyecto principal, cuya función es la administración de los proyectos.
- La iteración (*Iteration*) de SPEM 2.0 representa un período de tiempo significativo en los proyectos, se relaciona con MS Project 2010 mediante la tarea llamada resumen, que ayuda a organizar la lista de tareas.
- La actividad (*Activity*) en SPEM 2.0 son unidades de trabajo general de un proceso. En MS Project 2010 se representa como la tarea que agrupa a una serie de tareas con un mismo objetivo.
- La tarea (*Task*), tanto en SPEM 2.0 como en MS Project 2010, se denomina de la misma manera, representando la porción más pequeña del trabajo. La diferencia principal radica que en MS Project 2010 se debe indicar la duración y su delimitación (fecha de inicio y fecha de fin) a la tarea. Lo que SPEM 2.0 y MS Project 2010 tienen en común es la propiedad de dependencia o predecesoras entre las tareas.
- En SPEM 2.0 el rol (*Role*) define las competencias y responsabilidades de un individuo o grupo y está asociado a los productos de trabajo o a las habilidades que el rol provee. En MS Project 2010 la llamada información del recurso (aunque no solamente se utiliza para el manejo de personal, sino también para los materiales y los costos) permite crear, asignar o eliminar los roles o personas al/del proyecto. Al gestionar los recursos, MS Project 2010 permite asignar los valores de costos bajo ciertas premisas. A las tareas o actividades se le asignan los nombres de los recursos.
- Los hitos (*Milestone*), que se denomina de la misma manera tanto en SPEM 2.0 como en MS Project 2010, representan un evento significativo para el desarrollo del proyecto.

A continuación se muestra la tabla 5.1 que menciona los términos utilizados para establecer la relación entre SPEM 2.0 y Microsoft Project 2010:

En la figura 5.1, se puede ver a manera de ejemplo un proceso desarrollado en SPEM 2.0, desde la estructura de desglose de trabajo y en la figura 5.2, de forma similar el mismo

| SPEM 2.0 | Microsoft Project 2010 |
|---------------------|--|
| Procesos de entrega | Archivo del proyecto principal |
| Iteración | Resumen |
| Actividad | Tarea que agrupa a una serie de tareas |
| Tarea | Tarea |
| Rol | Información del recurso |
| Hitos | Hitos |

TABLA 5.1: Resumen comparativo SPEM 2.0 vs. Microsoft Project 2010

plan de proyecto desde la vista diagrama de Gantt con el objetivo de ver las diferencias que poseen.

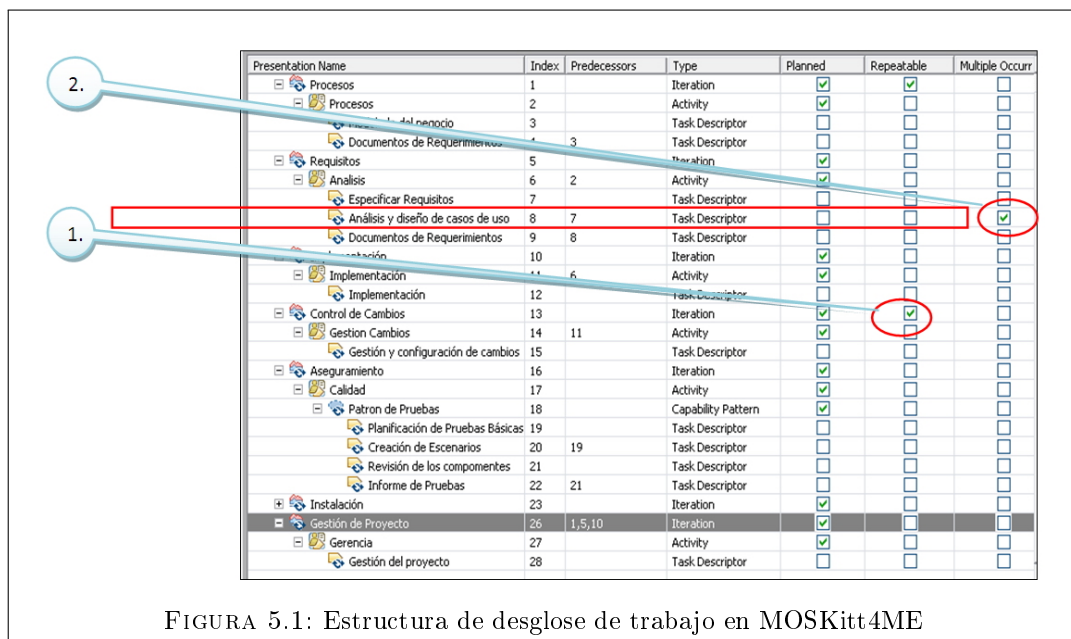


FIGURA 5.1: Estructura de desglose de trabajo en MOSKitt4ME

1. La primera diferencia que se puede apreciar en la 5.1, en la tarea de “Análisis y diseños de casos de uso”, donde el atributo de múltiples ocurrencias se encuentra activado ($has_multiple_occurrences \equiv true$) en la estructura de desglose de trabajo (*Work Breakdown Element*), es el hecho de que existe una sola tarea. Comparando con la figura 5.2 se nota que existen dos tareas.
2. La segunda diferencia se muestra en la iteración en SPEM 2.0, “Control de Cambios”. Para que este grupo de tareas se repita es necesario tener activado el atributo de que es repetible ($Is_repeatable \equiv true$), mientras en MS Project 2010 se debe planificar en dos oportunidades las actividades de “Control de Cambios”.

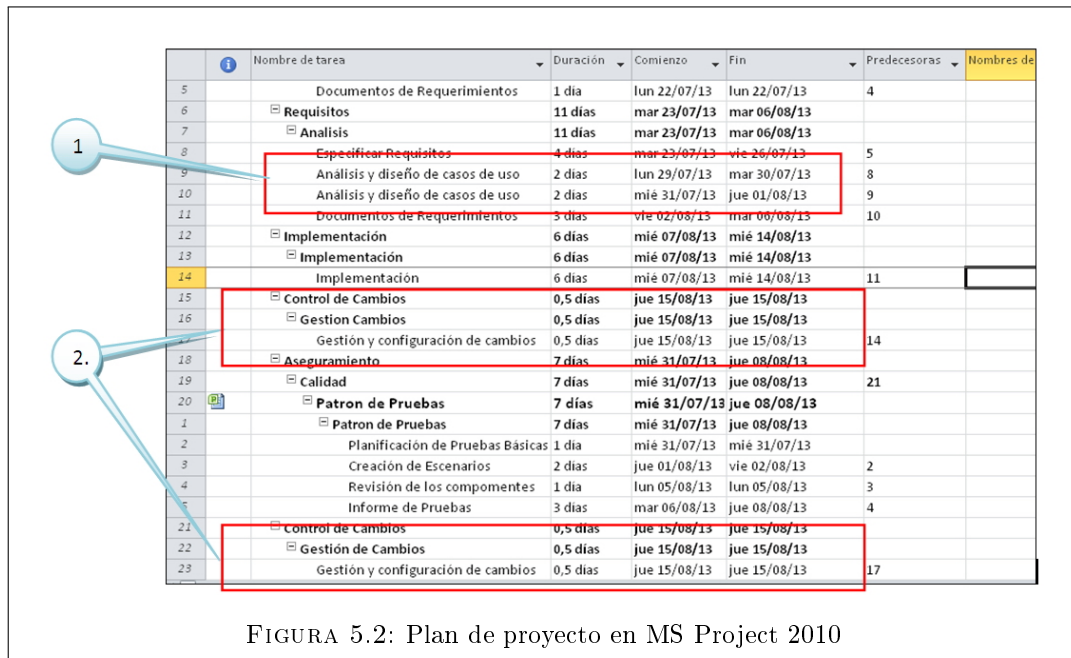


FIGURA 5.2: Plan de proyecto en MS Project 2010

- En MS Project 2010 existe una opción que se llama “Herramientas de Organizador de equipo” que presenta los recursos con las tareas asignadas para una fecha determinada. Estas vistas de recursos ayudan a determinar la criticidad de los mismos, y, por consecuencia, a realizar una mejor gestión y coordinación de dependencias.
- Otra funcionalidad que posee MS Project 2010 y que no posee SPEM 2.0 es la creación de líneas bases que consisten en un plan actual previamente acordado, siendo una referencia inicial, antes de comenzar los proyectos, para poder comparar los cambios que puedan surgir a lo largo de dichos proyectos.

5.1.1. Conclusiones

Después de haber realizado el mapeo entre SPEM 2.0 y MS Project 2010, se puede concluir que el plan de proyecto (a diferencia del proceso representado en SPEM 2.0) define los siguientes puntos:

- Incluye los parámetros de planificación de los proyectos, entre ellos el coste, el esfuerzo y el calendario del proyecto.
- Se crea la línea base, con la cual es posible medir las desviaciones del plan, efectuando las acciones correctivas.
- Ayudan a determinar la criticidad de los recursos mediante las herramientas de organización de equipo. Al incorporar los planes de proyecto es posible que sean implementadas las siguientes prácticas específicas descritas en la tabla 5.2 .

| Nivel de madurez 2 | |
|---|---|
| Monitorización y control del proyecto (PMC) | |
| SG 1 Monitorizar el proyecto frente al plan. | SP 1.1 Monitorizar los parámetros de |
| SG 2 Gestionar las acciones correctivas hasta su cierre. | SP 2.1 Analizar las cuestiones. |
| | SP 2.2 Llevar a cabo las acciones correctivas. |
| | SP 2.3 Gestionar las acciones correctivas. |
| Planificación del proyecto (PP) | |
| SG 1 Establecer las estimaciones. | SP 1.2 Establecer las estimaciones de los |
| | SP 1.4 Estimar el esfuerzo y el coste. |
| SG 2 Desarrollar un plan de proyecto. | SP 2.1 Establecer el presupuesto y el calendario. |
| SG 3 Obtener el compromiso con el plan. | SP 3.1 Revisar los planes que afectan al |
| | SP 3.2 Conciliar los niveles de trabajo y de |
| | SP 3.3 Obtener el compromiso con el plan. |
| Gestión de configuración (CM) | |
| SG 1 Establecer las líneas base. | SP 1.3 Crear o liberar las líneas base. |
| Nivel de madurez 3 | |
| Definición de procesos de la organización (OPD) | |
| SG 1 Establecer los activos de proceso de la | SP 1.4 Establecer el repositorio de mediciones |
| Gestión integrada del proyecto (IPM) | |
| SG 2 Coordinar y colaborar con las partes interesadas relevantes. | SP 2.2 Gestionar las dependencias. |
| | SP 2.3 Resolver las cuestiones de coordinación. |

TABLA 5.2: Prácticas específicas implementadas al incorporar el plan de proyecto

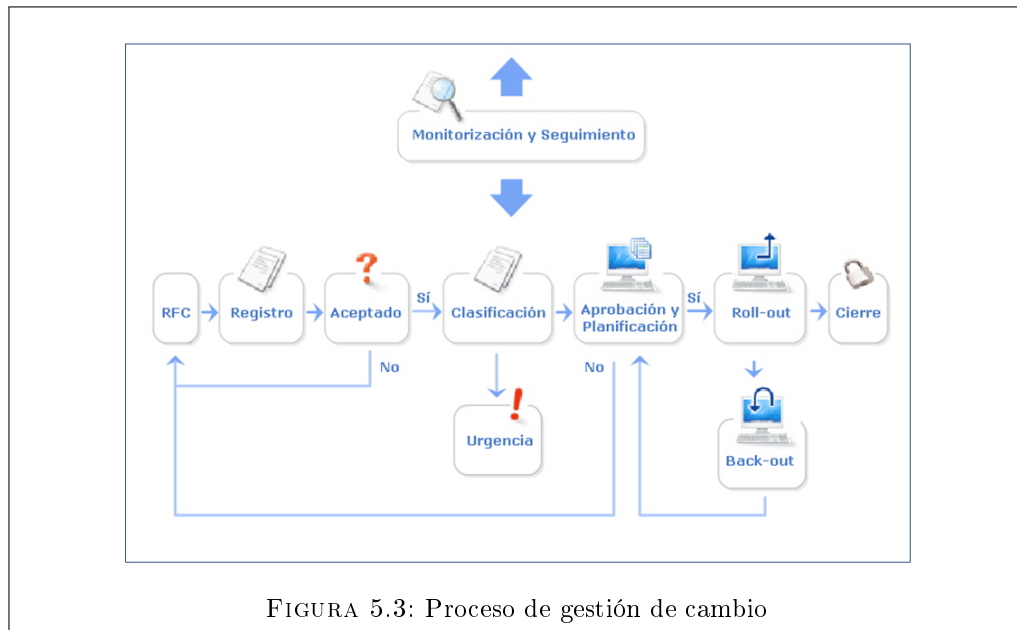
5.2. Creación de una gestión de cambio

A lo largo del ciclo de vida de un proyecto, es altamente probable que ocurran cambios. Para llevar de forma más eficiente y seguir los procedimientos establecidos, asegurando en todo momento la calidad y continuidad, es requerido poseer una correcta evaluación y planificación del proceso de cambio.

A continuación se mencionan algunos beneficios de una correcta gestión de cambio:

- Reducción del número de fallos que se encuentren asociados a los cambios.
- En caso de fallos que se encuentren asociados a los cambios, es posible volver a la configuración anterior.
- Permite el desarrollo de procedimientos de controles de cambio estándares, facilitando una rápida actualización de sistemas no críticos.
- Maximiza la productividad y minimiza los errores que puedan ocurrir en el desarrollo de los proyectos. A continuación se muestra la figura 5.3, que describe en forma gráfica como deberían llevarse las iteraciones y funcionalidades de una gestión de cambio¹:

¹Fuente tomada de http://itil.osiatis.es/Curso_ITIL/. Consultada el 14/07/2013



A continuación, se describe el proceso de gestión de cambio basado en ITIL ²:

- La monitorización y el seguimiento indican que todo proceso debe ser monitorizado, realizar informes de rendimiento y elaborar las métricas de rendimiento
- Las solicitudes de cambios se realizan mediante los RFC que son registros que poseen las especificaciones funcionales y técnicas de una versión, las cuales pueden ser aceptados o no, y pueden estar clasificadas en:
 - Versiones de emergencia que son modificaciones que reparan de forma rápida un error conocido
 - Versiones para su aprobación y planificación
- El proceso de distribución o implementación de la versión es conocido también como Rollout, y puede ser completo o fragmentado.
- De ocurrir algún fallo, con la nueva versión implementada, se requiere tener previsto planes o procesos de recuperación llamados Backout. Se reinicia el proceso para analizar la causa de la falla y tomar las correcciones respectivas.

La gestión de configuración debe contar con los siguientes aspectos funcionales:

- Manejo de versiones; debe ser capaz de guardar todas las versiones de los productos que se manejen o se produzcan. Permitiendo realizar un reverso de los cambios, cuando se requiera.

²ITIL (*Information Technology Infrastructure Library*) proporciona un marco práctico y sensato para identificar, planificar, entregar y apoyar a los servicios de TI para el negocio. Fuente tomada de <http://www.itil-officialsite.com> y consultada el 10/09/2013

- Gestión de seguimiento y dependencia de los cambios; cuando se realiza una modificación que pueda afectar a otro componente y este componente falla, es de suma importancia llevar un estricto control de dependencia entre los mismos, así se mantienen la integridad de la información y la generación de productos o artefactos que se produzcan.
- Rutas de auditoría; toda la información adicional que pueda servir de utilidad para realizar un rastreo o seguimiento a los cambios que se han realizado, por ejemplo, cuando, por quién, por qué, cuantas veces se han realizado los cambios.

5.2.1. Gestión de cambio en MOSKitt4ME

Se detalla los aspectos funcionales de la gestión de cambio en MOSKitt4ME.

1. El control de versiones se puede realizar de la siguiente manera: Actualmente, para la plataforma de Eclipse (“EPF Composer”), en los elementos de contenido del método, en la descripción de información de la versión (*Version Information*), existe la siguiente información:
 - Versión
 - Fecha de cambio
 - Descripción del cambio
 - Autores
 - Copyright o derechos de autor

Al contar con la información de versión de los cambios que se realizan, en la plataforma de Eclipse Process Framework Composer, que contiene la información mínima y mediante la aplicación, programación o adaptación de los plugins es posible poseer un completo control de versiones.

2. Para determinar el control de dependencias es de suma importancia la trazabilidad bidireccional que se realiza en SPEM 2.0 mediante la primitiva de variabilidad en los elementos de contenido, explicada en la sección 3.2 de la práctica específica “SP 1.4 Mantener la trazabilidad bidireccional de los requisitos”. No solo es aplicable a los requisitos, sino también a los elementos del contenido del método. El seguimiento y control permite mantener un estricto control de dependencia.
3. El resultado de la trazabilidad bidireccional y el histórico de un control de versiones, permiten realizar procesos de auditorías. A continuación se muestra en la tabla 5.3:

| Nivel de madurez 3 | |
|--------------------------------|---|
| Gestión de requisitos (REQM) | |
| SG 1 gestionar los requisitos. | SP 1.2 Obtener el compromiso sobre los requisitos |
| | SP 1.3 Gestionar los cambios a los requisitos. |
| | SP 1.4 Mantener la trazabilidad bidireccional de los requisitos |

TABLA 5.3: Prácticas específicas implementada al crear la gestión de cambios

5.3. Gestión de riesgos

Los riesgos son amenazas para los proyectos, por lo que deben ser minimizadas. Sin embargo, existen riesgos positivos llamados oportunidades. Los riesgos son eventos no certeros, expresados como impactos y con cierta probabilidad de ocurrencia.

A continuación se muestra en la figura 5.4) en donde:

- Las actividades están relacionadas con el desarrollo del plan de gestión de riesgos e implica cómo se va a realizar dicha gestión.
- Incluyen identificar riesgos que afectan al proyecto
- Se procede a analizar los riesgos bien sean un análisis cuantitativo de los riesgos o por priorización
- El plan de respuesta de riesgos que consiste en un seguimiento y control de los riesgos de un proyecto
- Por último el cierre de la gestión de los riesgos que consiste en documentar las lecciones aprendidas

5.3.1. Gestión de riesgos en MOSKitt4ME

De acuerdo con las actividades de una gestión de riesgo descrita por [3], se puede integrarlas con MOSKitt4ME mediante las siguientes consideraciones:

1. La planificación de una gestión de riesgo consiste en una serie de actividades que se pueden definir mediante los procesos de despliegue, y que poseen asociados una serie de tareas, roles, artefactos y guías.

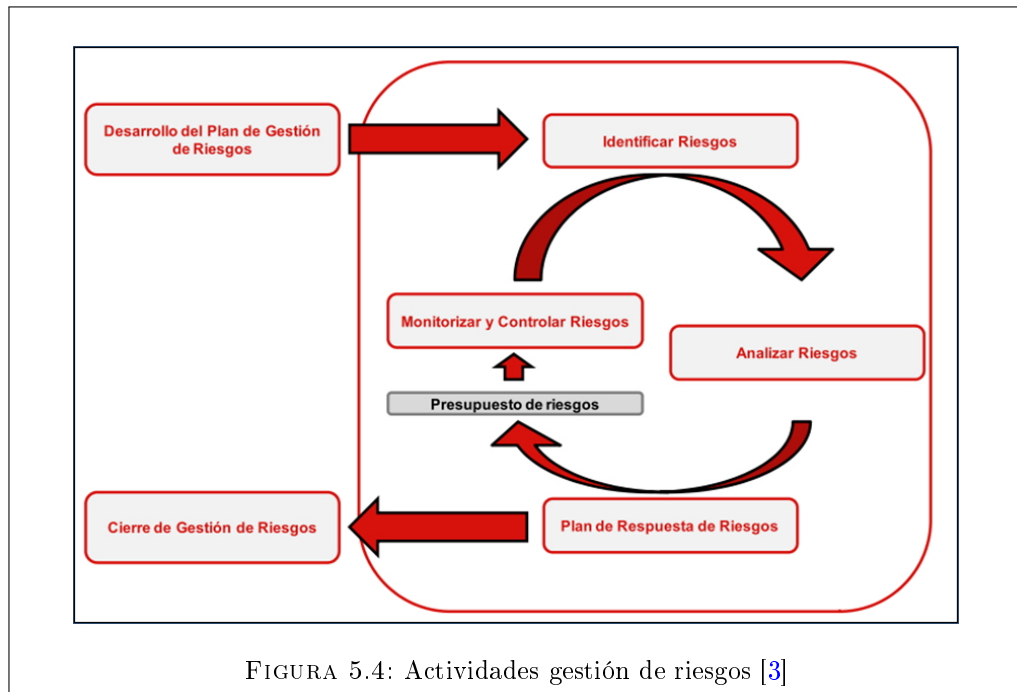


FIGURA 5.4: Actividades gestión de riesgos [3]

2. La identificación de los riesgos es una labor que dependerá de cada proyecto que se aborde. Sin embargo, con MOSKitt4ME, al estar basado en el enfoque de un desarrollo dirigido por modelos, se reducen los problemas de riesgo que ocurren comúnmente a nivel tecnológico. Entre los riesgos que se mitigan están:

- Las estrategias de desarrollo
- El conocimiento de tecnología

Ventajas:

- Al estar basado en metamodelos y técnicas de transformación de modelos, las implementaciones que se realicen representan el conocimiento del dominio y por lo tanto son reutilizables
 - Hasta cierto punto es posible la portabilidad y la interoperabilidad
3. Para asignar la prioridad a los riesgos identificados existe un campo descriptivo llamado nivel de riesgo (Risk Level), que se encuentra en el editor de SPEM llamado Eclipse Process Framework Composer, en las propiedades de los elementos de los procesos de despliegue.
4. El plan de respuesta y el cierre de los riesgos ya dependerán del tipo de riesgo o su nivel, y se pueden monitorizar mediante las dos opciones de ejecución de SPEM 2.0, bien sean mediante planes de proyecto o representaciones ejecutables.

De aplicarse la gestión de riesgos en MOSKitt4ME, ella incidiría de forma positiva en las siguientes áreas que se describen en la tabla 5.4.

| Nivel de madurez 2 | |
|--|--|
| Monitorización y control del proyecto (PMC) | |
| SG 1 Monitorizar el proyecto frente al plan. | SP 1.3 Monitorizar los riesgos del proyecto. |
| Planificación del proyecto (PP) | |
| SG 2 Desarrollar un plan de proyecto. | SP 2.2 Identificar los riesgos del proyecto. |
| Nivel de madurez 3 | |
| Gestión de riesgos (RSKM) | |
| SG 1. Preparar la gestión de riesgos. | SP 1.1 Determinar las fuentes y las categorías de riesgos. |
| | SP 1.2 Definir los parámetros de riesgos. |
| | SP 1.3 Establecer una estrategia de gestión de riesgos. |
| SG 2 Identificar y analizar los riesgos. | SP 2.1 Identificar los riesgos. |
| | SP 2.2 Evaluar, clasificar y priorizar los riesgos. |
| SG 3 Mitigar los riesgos. | SP 3.1 Desarrollar los planes de mitigación de riesgos. |
| | SP 3.2 Implementar los planes de mitigación de riesgos. |

TABLA 5.4: Prácticas específicas implementadas al crear la gestión de riesgos

5.4. Manejo de costos y esfuerzo

En SPEM 2.0 existen las métricas que son elementos especiales, que contienen una o más restricciones, proporcionando mediciones a cualquier elemento que sea descriptible. Se pueden definir métricas en la definición de trabajo; por ejemplo, los costos por hora, promedios de calidad, errores, etc. Las métricas son documentadas con descripciones de contenido asociadas a la métrica.

El uso requerido del tipo de guía para la estimación de métricas es imprescindible, ya que dicha guía provee el tamaño de las medidas o estándares de tamaño de esfuerzo y está asociada a una pieza de trabajo. E, incorporando adicionalmente las descripciones de contenido a las métricas en MOSKitt4ME, permitirá la definición de grupos de métricas de productividad, calidad y escala en las diferentes etapas de definición e implementación del método.

Gestionando a estos dos elementos antes mencionados de forma correcta, es posible implementar las siguientes prácticas específicas 5.5:

| Nivel de madurez 2 | |
|--|---|
| Monitorización y control del proyecto (PMC) | |
| SG 1 Monitorizar el proyecto frente al plan. | SP 1.1 Monitorizar los parámetros de planificación del proyecto. |
| Planificación del proyecto (PP) | |
| SG 1 Establecer las estimaciones. | SP 1.4 Estimar el esfuerzo y el coste. |
| SG 2 Desarrollar un plan de proyecto. | SP 2.1 Establecer el presupuesto y el calendario. |
| Medición y análisis (MA) | |
| SG 1 Alinear las actividades de medición y análisis. | SP 1.1 Establecer los objetivos de medición. |
| | SP 1.2 Especificar las medidas. |
| | SP 1.3 Especificar los procedimientos de recogida y de almacenamiento de datos. |
| | SP 1.4 Especificar los procedimientos de análisis. |
| SG 2 Proporcionar los resultados de la medición. | SP 2.1 Obtener los datos de la medición. |
| | SP 2.2 Analizar los datos de la medición. |
| | SP 2.3 Almacenar los datos y los resultados. |
| | SP 2.4 Comunicar los resultados. |
| Nivel de madurez 3 | |
| Definición de procesos de la organización (OPD) | SP 1.4 Establecer el repositorio de mediciones de la organización. |

TABLA 5.5: Prácticas específicas implementada al aplicar métricas

5.5. Validaciones y Verificaciones

Las validaciones y verificaciones son las confirmaciones de que los productos o servicios se han proporcionado con su uso deseado y reflejan adecuadamente los requisitos que se han especificado. A través de MOSKitt4ME es posible realizar las confirmaciones requeridas de los productos, no obstante mediante las restricciones OCL, se propone dar un mayor aporte para realizar el cumplimiento de forma automática a estas áreas de procesos.

5.5.1. Definición de OCL

Object Constraint Language (OCL) es un lenguaje notacional para el análisis y diseño de sistemas software. Fue desarrollado como un lenguaje de modelado de negocios y tiene sus raíces en el método Syntropy.

Está definido como lenguaje estándar para dar soporte al Unified Modeling Language (UML), que es un estándar de OMG para el análisis y diseño orientado a objetos. OCL da soporte a UML para la especificación de restricciones y consultas sobre modelos, permitiendo definir y documentar los modelos UML de forma más precisa [19]

Algunas de las características de las restricciones OCL son:

- Al ser un lenguaje de especificación pura, se garantiza que no existan efectos secundarios. Al evaluarse las expresiones, simplemente devuelve un valor
- No se puede cambiar nada en el modelo, las OCL se pueden utilizar para especificar un cambio de estado (por ejemplo, en un post-condiciones)
- Por no ser un lenguaje de programación, no es posible escribir lógicas de programación, invocar procesos, activar operaciones no consultadas en OCL
- Las expresiones OCL deben ajustarse a las normas de conformidad del estándar definido
- La evaluación de la expresión es instantánea, es decir, el estado del objeto no puede cambiar durante su evaluación

Las instrucciones OCL pueden ser usadas por UML y otros lenguajes basados en MOF, para especificar restricciones, expresiones de navegación, expresiones booleanas, consultas y otras expresiones incluidas en estos modelos

5.5.2. Uso de OCL

- OCL puede ser usado con distintos fines:
 - Para especificar invariantes sobre clases y tipos en el modelo de clases
 - Para especificar pre-condiciones y post-condiciones sobre operaciones y métodos
 - Para describir guardas (en statecharts y otros elementos)
 - Como lenguaje de navegación
 - Como lenguaje de consultas y propiedades derivadas
- OCL es utilizado también para especificar la semántica de UML. Uso de las restricciones OCL en MOSKitt4ME. Se podría complementar el conjunto de primitivas que actualmente posee SPEM 2.0 para mejorar y automatizar las áreas de procesos de validación y de verificación mediante la incorporación de las restricciones OCL de la siguiente manera:
- Asegurar que todos los requisitos posean identificadores únicos.

- Los parámetros de los requisitos que sean reutilizables posean un tipo válido, por ejemplo los tipos de activos reutilizables soportados por MOSKitt4ME: «Model Transformation», «Graphical Editor», «Form-based Editor», «Textual Editor» y «Meta-Model»
- La verificación de las relaciones de trazabilidad que se hayan definido entre los elementos de contenido del método, cuando se haya activado la variabilidad por contribución.

Al incorporar las restricciones de OCL se pueden implementar las siguientes prácticas específicas 5.6:

| Nivel de madurez 3 | |
|--|--|
| Integración del producto (PI) | |
| SG 3 Ensamblar los componentes de producto y entregar el producto. | SP 3.1 Confirmar la disponibilidad de los componentes de producto para la integración. |
| Solución técnica (TS) | |
| SG 3 Implementar el diseño del producto. | SP 3.1 Implementar el diseño. |
| Validación (VAL) | |
| SG 2 Validar el producto o los componentes de producto. | SP 2.1 realizar la validación. |
| Verificación (VER) | |
| SG 3 Verificar los productos de trabajo seleccionados. | SP 3.1 Realizar la verificación. |

TABLA 5.6: Prácticas específicas implementada al usar restricciones OCL

Capítulo 6

Conclusiones y trabajos futuros

En este trabajo se ha presentado la integración de CMMI-DEV v1.3 del nivel de madurez 2 y 3 con MOSKitt4ME. Para ello, se ha realizado una evaluación completa y detallada de la relación que existe entre CMMI-DEV v1.3 y MOSKitt4ME. Adicionalmente, se realizaron una serie de propuestas para extender MOSKitt4ME y de esta forma alcanzar las mejores prácticas descritas por CMMI®.

Con el propósito de realizar una evaluación y un análisis objetivos se siguió el método de SCAMPI v1.3 descrito en la Subsección 2.1.3, en donde se han evaluado las 140 prácticas específicas que comprenden los niveles de madurez 2 y 3. Se logró establecer las relaciones y divergencias que existen entre MOSKitt4ME y CMMI-DEV. Se pueden ver resumidos los resultados de forma cuantitativa en la tabla 3.1 y la tabla 4.1.

Durante la evaluación se ha determinado que las áreas de procesos que han sido implementadas representan el 61 % aproximadamente. Esto corresponde a 13 áreas de procesos que mantienen relación con MOSKitt4ME (de las 18 que presenta CMMI-DEV v1.3 del nivel de madurez 2 y 3) que se consideran “Satisfechas”.

Se han logrado determinar importantes relaciones entre CMMI-DEV v1.3 y MOSKitt4ME para los niveles de madurez 2 y 3. Entre ellas se pueden mencionar:

- La trazabilidad bidireccional que se puede llevar a cabo mediante MOSKitt4ME mediante la variabilidad, específicamente mediante el tipo de variabilidad llamado contribución.
- La monitorización de los proyectos que se puede realizar mediante la fase de aplicación del método (ambiente CASE), lo que permite llevar un control del progreso del proyecto.

- El proceso estándar (los componentes de los procesos básicos) que se define en SPEM 2.0 como el patrón de capacidades (al ser reusables).
- Los diccionarios, bibliotecas, procedimientos, criterios e informes que se pueden definir mediante las guías de SPEM 2.0 y sus diferentes tipos de guías.

Las relaciones anteriormente descritas entre CMMI-DEV v1.3 y MOSKitt4ME son de vital importancia en las organizaciones, debido a que proporcionan calidad en sus procesos y métodos en el desarrollo de software. Por ejemplo:

- La trazabilidad bidireccional \Rightarrow permite la generación de productos con mejor calidad mediante una apropiada obtención de requerimientos.
- La monitorización de los proyectos \Rightarrow garantiza que los procesos están bien caracterizados y comprendidos al hacerlos más visibles.
- El proceso estándar \Rightarrow da soporte a los procesos estándares de la organización, les ayuda a mantener la integridad y la consistencia en los métodos y productos que se definen mediante MOSKitt4ME.

Existen otras relaciones conceptuales y semánticas que han sido fundamentales para determinar el grado de implementación entre las prácticas específicas del nivel de madurez 2 y 3 de CMMI-DEV v1.3 y MOSKitt4ME. Las cuales se pueden ver resumidas en la tabla 3.2 y la tabla 4.2.

Para lograr disminuir el grado de implementación que es el 39 % restante e incluso incorporar áreas de proceso que no presentan puntaje, se han propuesto algunas alternativas teóricas que pueden ser implementadas en MOSKitt4ME, y así estar alienadas a las mejores prácticas propuestas por CMMI-DEV v1.3.

Las propuestas que han surgido se han basado en las prácticas específicas de CMMI-DEV v1.3 del nivel de madurez 2 y 3 que no han sido implementadas en MOSKitt4ME. El propósito fue eliminar las brechas que se detectaron y así extender las funcionalidades que actualmente presenta MOSKitt4ME.

Algunas ventajas en las organizaciones al implementar las propuestas serían las siguientes:

- Durante la ejecución de los procesos del método, de poseer un plan de proyecto la organización puede generar sus planes, indicadores y estimaciones que les sirven de base de conocimiento:

- Se pueden crear las líneas bases, fundamentales para determinar las posibles desviaciones del plan y por ende tomar las acciones correctivas.
 - Se incluyen los parámetros en la planificación de los proyectos, como son las definiciones del calendario y de los costos.
 - Otro beneficio es que existen términos que son comunes entre SPEM 2.0 y Microsoft Project 2010 (herramienta de gestión de proyecto que se analizó), ver Tabla 7 Resumen comparativo SPEM 2.0 vs. Microsoft Project 2010.
- Contar con una gestión de cambio; para ello se debe aprovechar de las funcionalidades de la plataforma de Eclipse con la información de la versión y la variabilidad por contribución (primitiva de SPEM 2.0) y adicionalmente incorporar las auditorías. Estos puntos traerían como consecuencia la reducción de número de fallas, controles de cambio estándar, maximizan la productividad y minimizan los errores.
 - Al incorporar validaciones y verificaciones de forma automática mediante el uso de las restricciones de OCL, se garantiza que desde todas las fases del desarrollo del método se realicen los procesos de validaciones y verificaciones, por ejemplo “Asegurar que todos los requisitos posean identificadores únicos”. Por consecuencia, productos y métodos con mejor calidad.

Aspectos a considerar para trabajos futuros serán implementar las propuestas teóricas que han surgido a lo largo del análisis realizado.

Por otro lado, se plantea la creación de una gestión de cambio, como una propuesta o un control de las modificaciones que puedan surgir durante la creación de métodos de desarrollo de software en MOSKitt4ME. Sin embargo, existen técnicas o propuestas que describen tipos de propagación de cambio en el contexto de la ingeniería de método [20] que podría ser evaluado.

Apéndice A

Glosario

A.1. Artefactos

Representan de forma tangible la evidencia del trabajo que se realiza y pueden incluir las políticas de la organización u otros productos de trabajo a nivel de aplicación. Para verificar la implementación del modelo de la práctica asociada son necesarios suficientes artefactos que demuestran y corroboran el trabajo que se realiza.

A.2. Afirmaciones

Son declaraciones orales o escritas que confirman o apoyan la implementación (o la falta de implementación) de una práctica de modelo, proporcionadas por quienes implementan la práctica.

A.3. Línea Base

El conjunto de especificaciones que sean claves o productos de trabajo, que se han revisado y acordado en un momento del tiempo, servirán para comparar la planificación original del proyecto con el desarrollo posterior, en donde sólo se podrá modificar dichos conjuntos de especificaciones mediante acuerdos previamente discutidos por las personas involucradas.

A.4. Gestión Configuración

“Una disciplina que aplica la dirección técnica y administrativa, y la supervisión para (1) identificar y documentar las características funcionales y físicas de un elemento de configuración, (2) controlar los cambios a esas características, (3) registrar e informar sobre el proceso de cambios y su estado de implementación, y (4) verificar el cumplimiento de los requisitos especificados.” [8]

A.5. Requisitos

Son condiciones o capacidades que son necesitadas para dar solución a un problema o para lograr un objetivo.

A.6. Revisiones entre Pares

Son las verificaciones de los productos de trabajo que efectúan personas con el mismo conocimiento que los autores de los desarrollos de los productos de trabajo, con el propósito reconocer y eliminar posibles defectos.

Bibliografía

- [1] A. Niknafs and R. Ramsin, “Computer-aided method engineering: An analysis of existing environments,” in *Advanced Information Systems Engineering* (Z. Bellahsene and M. Léonard, eds.), vol. 5074 of *Lecture Notes in Computer Science*, pp. 525–540, Springer Berlin Heidelberg, 2008.
- [2] F. Ruiz and J. Verdugo, *Guía de Uso de SPEM 2 con EPF Composer*. Universidad de Castilla-La Mancha, Abril 2008.
- [3] INTECO, “Guía avanzada de gestión de riesgo,” 2008.
- [4] SEI, *CMMI® for Development, Version 1.3*. Carnegie Mellon, Noviembre 2010.
- [5] M. Cervera, M. Albert, V. Torres, and V. Pelechano, “A model-driven approach for the design and implementation of software development methods,” in *International Journal of Information System Modeling and Design (IJISMD)*, 2012.
- [6] S. Brinkkemper, “Method engineering: engineering of information systems development methods and tools,” *Information and Software Technology*, vol. 38, no. 4, pp. 275 – 280, 1996.
- [7] SEI, *Standard CMMI® Appraisal Method for Process Improvement (SCAMPI SM) A v1.3*. Carnegie Mellon, Marzo 2011.
- [8] M. L. Peralta, “Asistente para la evaluación de cmmi-sw,” Master’s thesis, ITBA, 2004.
- [9] OMG, *Business Process Model and Notation (BPMN) (v2.0)*. OMG Object Management Group, 2011.
- [10] M. Cervera, M. Albert, V. Torres, and V. Pelechano, “The moskitt4me approach: Providing process support in a method engineering context,” in *Conceptual Modeling* (P. Atzeni, D. Cheung, and S. Ram, eds.), vol. 7532 of *Lecture Notes in Computer Science*, pp. 228–241, Springer Berlin Heidelberg, 2012.

-
- [11] M. Cervera, “Model driven method engineering. a supporting infrastructure,” Master’s thesis, Universidad Politécnica de Valencia. Departamento de Sistemas Informáticos y Computación., 2010.
- [12] F. Harmsen, S. Brinkkemper, and J. L. H. Oei, “Situational method engineering for informational system project approaches,” in *Proceedings of the IFIP WG8.1 Working Conference on Methods and Associated Tools for the Information Systems Life Cycle*, (New York, NY, USA), pp. 169–194, Elsevier Science Inc., 1994.
- [13] M. Cervera, M. Albert, V. Torres, and V. Pelechano, “Turning method engineering support into reality,” in *Engineering Methods in the Service-Oriented Context* (J. Ralyté, I. Mirbel, and R. Deneckère, eds.), vol. 351 of *IFIP Advances in Information and Communication Technology*, pp. 138–152, Springer Berlin Heidelberg, 2011.
- [14] OMG, *Software & Systems Process Engineering Meta-Model Specification (SPEM)*, (v2.0). OMG Object Managemet Group, 2008.
- [15] S. Bezerra, A. Lins De Vasconcelos, and R. Cavalcante, “Mapeamento dos conceitos do guia do cmmi em notações do spem no contexto da definição do processo de software,” vol. v. 5, pp. 83–92, INFOCOMP (UFLA), 2006.
- [16] C. Portela, A. Vasconcelos, A. Silva, E. S. Ariane Sinimbú, M. Ronny, W. Lira, and S. Oliveira, “A comparative analysis between bpmn and spem modeling standards in the software processes context,” in *The 5th International Conference on Computer Science and Software Engineering (CSSE 2014), January 12-14, 2014, Shenzhen, China*, vol. 5, pp. 330–339, Journal of Software Engineering and Applications, Mayo 2012.
- [17] V. Esterkin and C. Pons, “Análisis y evaluación del mdd (model driven software development) desde la perspectiva del nivel 2 del cmmi-dev 1.3,” in *Workshop Ingeniería de software (WIS)* (C. Commons, ed.), Centro de Altos Estudios en Tecnología Informática (CAETI), 2012.
- [18] OMG, *Reusable Asset Specification (RAS)*. *OMG Available Specification version 2.2*. OMG Object Managemet Group, 2005.
- [19] OMG, *OMG Object Constraint Language (OCL) (v2.3.1)*. OMG Object Managemet Group, 2012.
- [20] M. Saeki, “Configuration management in a method engineering context,” in *Advanced Information Systems Engineering* (E. Dubois and K. Pohl, eds.), vol. 4001 of *Lecture Notes in Computer Science*, pp. 384–398, Springer Berlin Heidelberg, 2006.