risti
Revista Ibérica de Sistemas e Tecnologias de Informação
Revista Ibérica de Sistemas y Tecnologías de Información

Recebido/Submission: 03/27/2016
Aceitação/Acceptance: 04/15/2016

Una aproximación basada en metamodelado del área de proceso de Validación del CMMI: Un caso de estudio

José de J. Jiménez Puello[1], Tomás San Feliu[2], Jose A. Calvo-Manzano[2]

jjimpue@hotmail.com, {tomas.sanfeliu, joseantonio.calvomanzano}@upm.es

[1] Universidad de Panamá, Facultad de Informática, Electrónica y Comunicación, Departamento de Informática, Campus Octavio Méndez Pereira, Panamá, Panamá

[2] Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros Informáticos, Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software, Boadilla del Monte, 28660, Madrid, España

DOI: 10.17013/risti.17.26–40

Resumen: En el presente artículo se establece una aproximación para la mejora de procesos a través del área de proceso de validación del CMMI, mediante un enfoque basado en metamodelado. Se consideró el área de proceso de Validación, específicamente la meta SG1 preparar la validación. Mediante el metamodelo, se definen una taxonomía de proyectos, la caracterización de pruebas, plantillas de pruebas para el producto a validar, para el entorno de validación, para los procedimientos y criterios de prueba, y un plan de pruebas. La aproximación fue sometida a evaluación, por medio de un caso de estudio. El caso de estudio se llevó a cabo en la Dirección de Informática, de una institución de educación superior pública. La aproximación demostró su validez, ya que, los probadores consideran que les aporta las pruebas específicas para el desarrollo del proceso de validación y les permite preparar la validación para un proyecto determinado.

Palabras-clave: CMMI; mejora de procesos; metamodelo; modelo de madurez; validación de software.

An approach based on metamodelling for the Validation CMMI process area: A case study

Abstract: In this article an approach to process improvement is established through the validation process area of CMMI, by focusing on making a metamodel. Validation process area was considered specifically, SG1 prepare for validation. By the metamodel, a taxonomy of projects, the testing characterization, the testing templates for the product to validate, for the validation environment, for the testing procedures and criteria, and a test plan are defined. The focus was subjected to evaluation by a case study. The case study was conducted in the Department of Computer Science, a public higher education institution. The approach proved its validity, since testers feel that it gives them specific tests for the development of the validation process and allows them to prepare validation for a particular project.

Keywords*:* CMMI; process improvement; metamodel; maturity model; software validation.

Artigos RISTI N17.indd 26 5/12/2016 1:22:04 PM

1. Introducción

En la actualidad la calidad es considerada como el elemento de mayor importancia y que presenta un impacto directo en cuanto al éxito de un producto software (Faizan, Ulhaq & Khan, 2015). La baja calidad en el software puede ser el origen de algunos problemas como pueden ser la falta de fiabilidad y un mayor mantenimiento del producto (Elhag, Elshaikh, Mohamed & Babar, 2013). El mercado valora, cada vez más, la calidad del software, por consiguiente, las compañías demandan la reducción de errores y sancionan los retrasos en entregas y las cancelaciones de proyectos (Piattini, García, García Rodriguez & Pino, 2014). A pesar de la importancia de las pruebas de software, muchas empresas tienen dificultad para generar el proceso de prueba (Camargo, Ferrari & Fabbri, 2014).

En la práctica, la definición de los mecanismos que pueden adoptarse para mejorar y controlar las fases del proceso de pruebas y el orden en que deben aplicarse, en general, es un tarea difícil (Anderson, 2004). Un gran número organizaciones reconocen que las iniciativas de mejora de los procesos pueden resolver estos problemas (Camargo, Ferrari & Fabbri, 2014, Muñoz, Gasca & Valtierra, 2014). Aunque el uso de buenas prácticas permite reducir el coste de pruebas fallidas o poco fiables, ya que el personal de prueba garantiza que las pruebas funcionen de forma correcta (Kerry & Delgado, 2009).

Es primordial que toda organización que se dedique al desarrollo de TI establezca una serie de mecanismos de control que permitan determinar que su producto cumpla con normas, procesos y estándares de calidad, que garanticen que su producto esté libre de errores, ante el mercado competitivo. Uno de los procesos que aseguran esta competitividad es la validación de software. La validación no es más que la prueba de software en sí (Monteiro, Machado & Kazman, 2009; Oh, Choi, Han & Wong, 2008; Kurokawa & Shinagaw, 2008, Méndez, Pérez, Domínguez & Mendoza, 2010).

Según (Ballou, 2008) todavía las organizaciones tienen que tratar con los defectos de software, un año después de ser entregado. El 55% del software sigue dando problemas dos años después de su implementación, donde un 55 por ciento señala que el software presenta problemas en el transcurso de los meses iniciales después de realizada la prueba. Debido a ello muchos proyectos presentan fallos debido a que las funcionalidades del sistema no satisfacen de forma esperada las necesidades de los usuarios en el lugar de trabajo (Cooter, 2008; King, 2008; Davis & Venkatesh, 2004). Lo anterior trae como consecuencia que los proyectos de desarrollo de software fallen debido a pruebas inadecuadas, a pruebas no planificadas o pobres, que indican que el proyecto no está metodológicamente comprobado, lo que ocasiona fallos, probablemente debido a la inadecuada capacitación de los usuarios que desconocen el propósito de las pruebas (Quillen, 2011; Jäntti, 2008).

El problema en la preparación de la validación, debido a la ausencia de planificación y estrategias para abordar las pruebas y la escasa o nula formación por parte de los probadores, incide en un proceso de prueba desorganizado donde los métodos, técnicas, pruebas, entorno de validación, procedimientos y criterios de pruebas de validación no corresponden seguramente al proyecto a validar. Tal como señalan (Iyengar & Karamouzis, 2007), la falta de pruebas y estándares de calidad, como la ausencia de

Artigos RISTI N17.indd 27 5/12/2016 1:22:04 PM

consistencia, son precursores de que sucedan interrupciones en las organizaciones debido a fallos en el software que pueden ser costosos. No hay una definición clara de los tipos de pruebas que deben aplicarse a un tipo de proyecto específico que cada organización aplica (Causevic, Sundmark & Punnekkat, 2010).

A pesar que la industria y la academia han desarrollado modelos y estándares orientados a la mejora de procesos, mediante la aplicación de buenas prácticas, que incluyen el proceso de validación como son: el modelo de capacidad y madurez integrado para desarrollo versión 1.3 (CMMI), ISO 12207, ISO 9001, IEEE-1012 verificación y validación y el modelo de mejora de procesos del software brasileño (MPS.BR), éstos solo indican "el Qué" sin indicar "el Cómo" (Humphrey *et al.*, 2007, Faroog & Dumke, 2007). Uno de los problemas de la validación es que los modelos establecidos para este proceso son difíciles de utilizar, porque solo indican "el qué hacer" (García Guzmán, de Amescua Seco & Velasco de Diego, 2006).

Además, uno de los mayores problemas en la prueba de software, radica en determinar si de acuerdo al tipo de software, se han realizado las pruebas adecuadas (Desai & Shah, 2011; Nayyar & Rizwan, 2009; Rajamani, 2006). Esto último es determinante en un proyecto porque los problemas que afectan el proceso de validación, inciden en que los proyectos de software fracasen y presente errores, fallos o defectos.

Estos problemas deben abordarse desde la perspectiva de la mejora de procesos a través de un metamodelo, que integre mejores prácticas de prueba para la validación de software. Por consiguiente, se ha desarrollado un metamodelo de validación de software que facilite a las organizaciones establecer la preparación e identificación de las pruebas para realizar la validación de un proyecto.

2. Trabajos relacionados

Un metamodelo es presentado por (Fleurey, Steel & Baudry, 2004), para técnicas de validación de software en el entorno de la arquitectura dirigida por modelos. Los autores sugieren que a través de un modelo de pruebas se puede establecer técnicas de pruebas funcionales y estructurales, incluso otros tipos prueba como la de mutación para mejorar la calidad de la prueba. Pero resaltan que queda mucho por hacer hasta que no se normalice el lenguaje para el modelo de transformaciones. Por su parte (Fleurey, Steel, Baudry & Le Traon) establecen un metamodelo basado en generación de datos de pruebas denominados (modelos de pruebas) para modelos de transformación en el contexto de la arquitectura dirigida por modelos, como mecanismo de comprobar las transformaciones que tiene que realizar los modelos generados. Para ello, implementa un algoritmo que crea modelos de pruebas de un metamodelo. (Hsue, Shenm, Yang & Yang 2008) presentan un metamodelo que aplica un modelo de simulación para comprobar la definición de procesos implementados en SPEM, mediante las prácticas de validación del CMMI. En (Sadilek & Weißleder, 2008) proponen un metamodelo para pruebas de metamodelos, que determine la existencia de errores en los metamodelos. Para ello, se enfoca en un metamodelo de prueba automatizado basado en una especificación de prueba. (Hernández, King & Clarke, 2008), establecen un metamodelo para soportar

Artigos RISTI N17.indd 28 5/12/2016 1:22:04 PM

pruebas de regresión para aplicaciones Web, debido al rápido cambio en la tecnología, lo que genera que las aplicaciones que migran de forma continua no puedan validarse debido al cambio de las plataforma y de las pruebas. (Lévesque, 2009) proponen un metamodelo para pruebas de unidad para lenguaje de programación orientado a objeto, que facilita escribir pruebas de unidad dentro de las clases para que sean probadas. (Marin, Vos, Giachetti, Baars & Tonella, 2011), proponen una metodología de prueba para aplicaciones Web, mediante la integración de búsqueda de pruebas, a través de un metamodelo de prueba que implementa la metodología.

Los metamodelos descritos no resuelven la problemática de preparar la validación y tampoco identifican las pruebas a la que debe someterse un proyecto específico. Además aplican el concepto de metamodelo definido para la arquitectura dirigida por modelos, que como señala (Muñoz, 2003) un metamodelo es un soporte para crear y desarrollar modelos a partir de un modelo.

Por el contrario, el metamodelo a plantear, debe resolver la problemática de preparar la validación. El metamodelo consiste en una solución innovadora referente a los modelos y estándares existentes, ya que, proporciona "el cómo hacer" identificando y aplicando las prácticas de validación para la mejora del proceso de validación. El metamodelo diseñado no aplica el concepto de metamodelo de la arquitectura dirigida por modelos que trata sobre la transformación de modelos.

3. Metamodelo para validación de software

El metamodelo que se presenta toma como referente el concepto de que a partir de un modelo se podrán definir otros modelos. Es decir, el metamodelo permite que se puedan establecer modelos para preparar la validación acordes a diferentes proyectos, tomando como referente la estructura base del metamodelo que a su vez es el modelo. El metamodelo desarrollado está basado en el modelo CMMI para desarrollo v.1.3, que al ser un metamodelo puede derivar otros modelos. El CMMI está integrado por áreas de procesos, el área de proceso seleccionada para la definición del metamodelo es el área de procesos de validación, y su meta específica preparar la validación que incluye las prácticas específicas (SP 1.1. Seleccionar los productos a validar, SP 1.2. Establecer el entorno de validación y SP 1.3 Establecer los procedimientos y criterios de validación). Estas prácticas específicas permiten organizar y establecer el proceso de preparación de la validación de software para luego ejecutar la validación. El metamodelo basado en estas prácticas específicas, representa un marco para establecer los aspectos que debe tomar en cuenta una organización de desarrollo de software al momento de validar el software desarrollado.

3.1. Diseño del Metamodelo

El metamodelo está organizado a partir de una taxonomía de proyectos de software, un método para caracterización de pruebas, plantillas de prueba y el plan de pruebas como resultado de la aplicación de las prácticas específicas definidas para preparar la validación (Fig. 1).

Artigos RISTI N17.indd 29 5/12/2016 1:22:04 PM

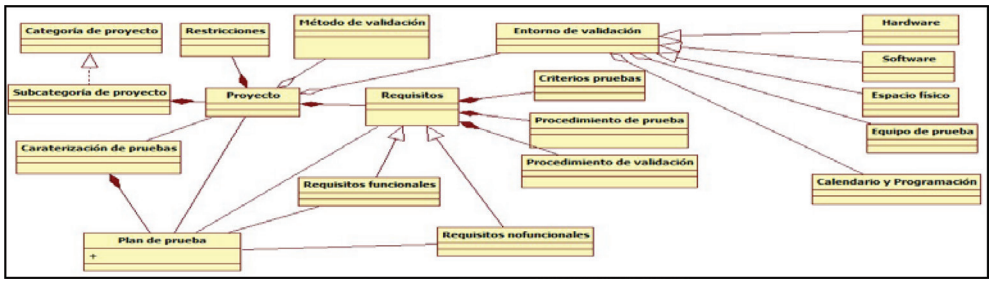

Figura 1 – Metamodelo para validación.

La taxonomía (Fig. 2) define una serie de categorías, y subcategorías de proyectos, que establece los proyectos específicos en base a su tecnología para desarrollo de software (Calvo-Manzano, Cuevas, San Feliu & Jiménez, 2015). Otro elemento del metamodelo, es la caracterización de pruebas que permite identificar cuáles son las pruebas de software que deben ser aplicadas a un proyecto específico para validarlo Calvo-Manzano, Cuevas, San Feliu & Jiménez, 2015).

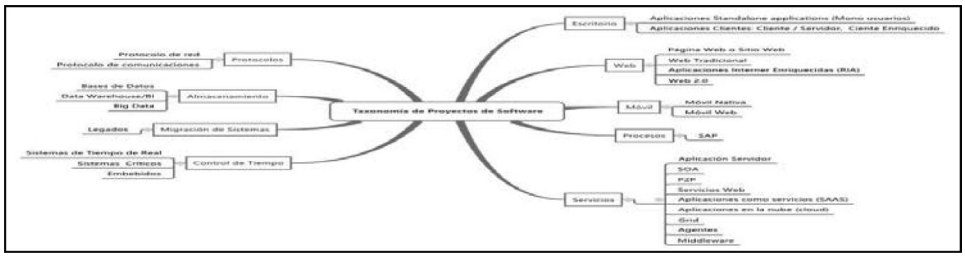

Figura 2 – Taxonomía

La caracterización establece un método (Fig. 3.) para identificar las pruebas de acuerdo a un proyecto determinado.

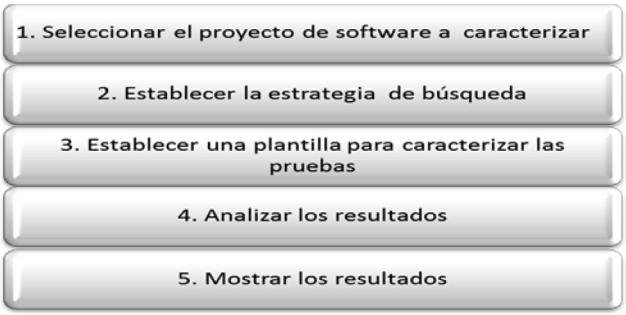

Figura 3 – Método para caracterización

Artigos RISTI N17.indd 30 5/12/2016 1:22:05 PM

El metamodelo además, establece una serie de plantillas de prueba basadas en las prácticas SP 1.1, SP 1.2 y SP 1.3. La plantilla que corresponde a la práctica SP 1.1 establecer el producto, está conformada por el producto a validar (proyecto software), los requisitos, restricciones de validación del producto y el método de validación.

El producto se selecciona a partir de la taxonomía de proyectos. Para cada producto se debe seleccionar la categoría y subcategoría del proyecto a la que pertenece el producto. Además, se identifica el alcance del proyecto y si el producto es un proyecto o componente de proyecto, las restricciones que presenta el producto y se establecen los requisitos funcionales y no funcionales que deben comprobarse en el producto (Fig. 4).

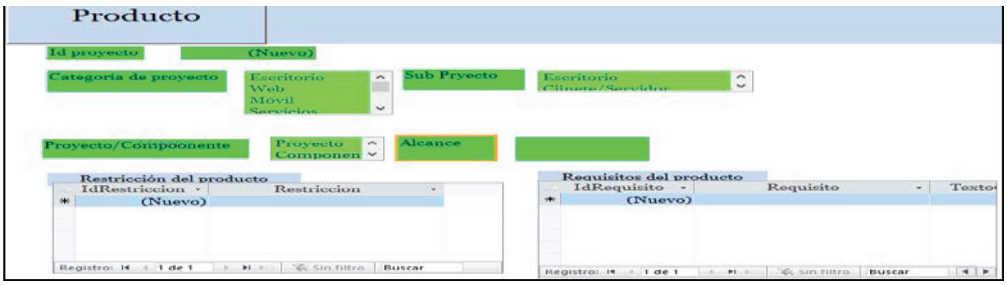

Figura 4 – Establecer el producto

La plantilla de la práctica SP 1.2 establecer el entorno de validación, implica los recursos de software, hardware, datos de pruebas, comunicación de datos, equipo de pruebas, útiles de oficina, espacio físico, calendario y programación de las pruebas. El entorno de validación establece los elementos que deben considerarse para probar los requisitos del producto según el proyecto o componente de proyecto (Fig. 5).

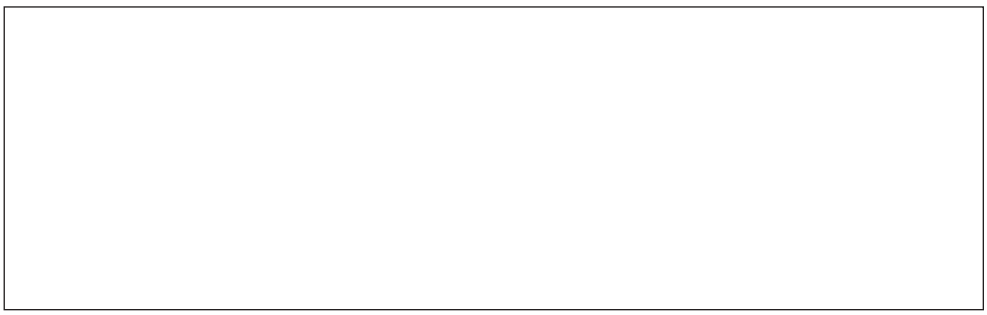

Figura 5 – Entorno de validación

Como parte del entorno de validación, está el equipo de prueba. Este se refiere al personal asignado para llevar a cabo las diferentes pruebas de cada requisito del producto. Se determina la responsabilidad dentro del equipo de prueba de cada persona y a que entorno de prueba está asignado (Fig. 6).

Artigos RISTI N17.indd 31 5/12/2016 1:22:05 PM

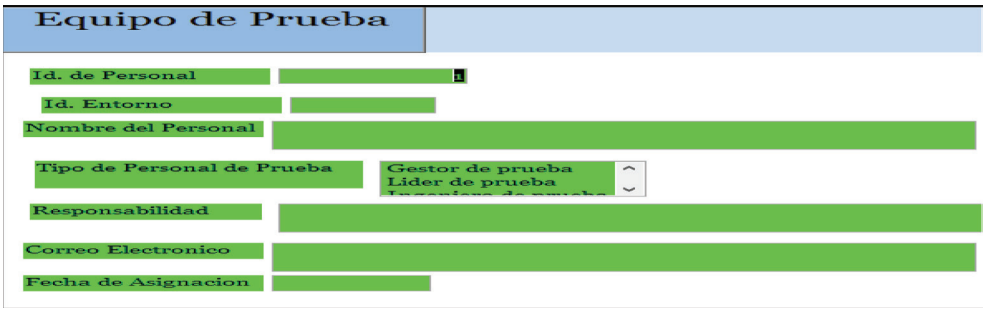

Figura 6 – Equipo de prueba

Por su parte la plantilla de la práctica SP 1.3, incluye: el procedimiento y criterios de validación. El procedimiento de prueba describe el caso de uso y el caso de prueba que debe ser utilizado para probar el requisito del producto. Por medio del procedimiento de prueba el probador comprueba si el requisito cumple o no con el resultado esperado (Fig.7).

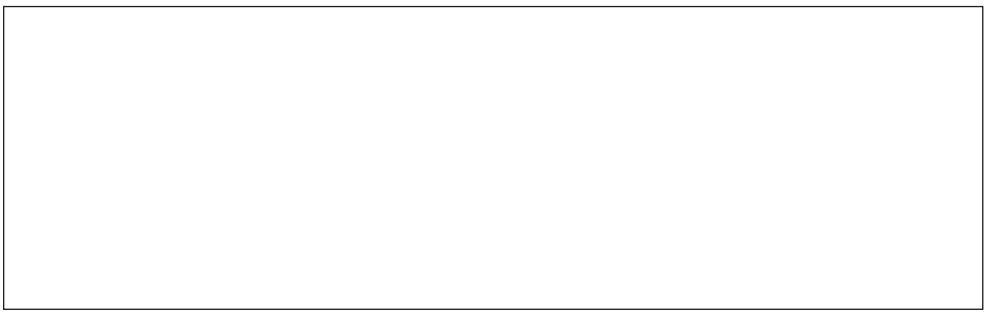

Figura 7 – Procedimiento de pruebas

Los criterios de prueba de validación, permiten determinar si la prueba del requisito pasó o falló, según los datos de entrada y salida que se obtiene con la prueba. Permite describir el fallo y si la prueba fue suspendida o reanudada. Además incluye la fecha a que corresponde dicho evento. El procedimiento está estrechamente relacionado con el procedimiento de prueba. Lo que ocurra con el caso de prueba repercute en los criterios de prueba de validación (Fig. 8).

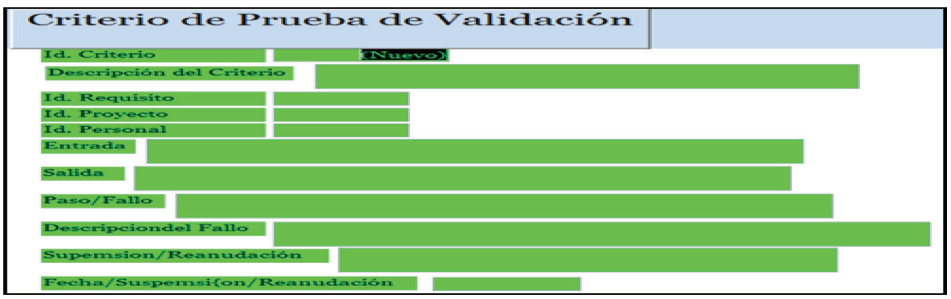

Figura 8 – Equipo de prueba

Artigos RISTI N17.indd 32 5/12/2016 1:22:06 PM

Además, el metamodelo incluye un plan de prueba que establece las pruebas a que debe someterse un proyecto como resultado de la caracterización de pruebas (Fig. 9.)

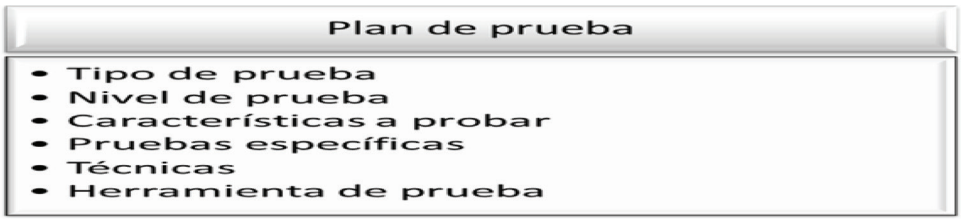

Figura 9 – Plan de prueba

4. Evaluación del metamodelo

En la evaluación del metamodelo se aplicó un caso de estudio acorde con (Wohlin, Höst & Henningsson, 2003), que plantea que una estrategia para caso de estudio, es comparar los resultados de una nueva propuesta y una línea base. Para establecer la línea base se aplicó la técnica de la encuesta a través de un cuestionario. Además, se definieron dos variables como indicadores para determinar el estado actual del proceso de validación, según las prácticas SP1, SP2 y SP3 en la organización de estudio. El cuestionario consta de once ítems (Tabla 1) que se definió para dar soporte a las variables. Se estableció una escala de tipo Likert (1= nunca, 2= pocas veces, 3= muchas veces, 4= casi siempre, 5= siempre) para determinar la frecuencia en la que se considera se ubica cada respuesta afirmativa. Las variables definidas son dos: Estado del proceso de validación y el estado de las pruebas en la organización.

Nº	Preguntas	Si	No	1	2	3	4	5
1	Aplican pruebas específicas al proyecto.							
2	Disponen de una metodología, modelo, guía, marco o referencia para el desarrollo de las pruebas.							
3	Definen un plan de pruebas.							
4	El plan de prueba está basado en los requisitos del usuario y del proyecto.							
5	Antes de validar define que pruebas se deben aplicar al proyecto.							
6	La pruebas se basan en algún estándar de calidad de software.							
7	Aplican técnicas de pruebas.							
8	Utilizan herramientas de pruebas.							
9	Las pruebas tiene definidos los procedimientos y criterios deben ser probados.							
10	Establece un entorno de validación.							
11	Disponen de un equipo de prueba con conocimiento en el desarrollo de pruebas.							

Tabla 1 – Cuestionario para determinar la línea base.

Artigos RISTI N17.indd 33 5/12/2016 1:22:06 PM

Las respuestas obtenidas en el cuestionario proporcionan información sobre el estado actual del proceso de validación en la unidad de estudio.

4.1. Caso de estudio

El caso de estudio se llevó a cabo en la Dirección de Informática, de una institución de educación superior pública. La dirección de informática tiene a su cargo el desarrollo de software académico, administrativo y financiero de la institución. Se pudo observar la ausencia de una estructura definida para el desarrollo de pruebas de validación, documentación y resultados, por lo que acogió con entusiasmo e interés aplicar el metamodelo en los proyectos de software que desarrolla, para la mejora de sus procesos de validación. Solo dispone de tres analistas programadores encargados del desarrollo de aplicaciones a nivel Web y cliente servidor. Los analistas programadores son los que realizan las pruebas de validación a cada proyecto que desarrollan. Los analistas programadores no tienen formación en validación y pruebas de software. El proyecto seleccionado, en este caso, es un proyecto Web tradicional que forma parte de una subcategoría dentro de la taxonomía de proyectos establecida como elemento del metamodelo.

Línea base

La población encuestada fue de tres analistas programadores, por consiguiente se trabajó con toda la población (N=3). Los resultados del estado del proceso de validación (variable 1), indican carencia de documentación del proceso de validación. El 100% señala que no se aplica metodología alguna, modelo, guía o marco de referencia para el desarrollo de las pruebas de validación de software. Un plan de prueba no se define ni se aplica, probablemente debido a que el personal no tiene conocimiento en pruebas de software (Fig. 10). Sin embargo, el 66% considera que establecen pruebas antes de validar. Esta confusión obedece a que aplican alguna prueba de entrada de datos para comprobar el resultado de salida, que obviamente, no implica que estén desarrollado un proceso organizado de validación del software.

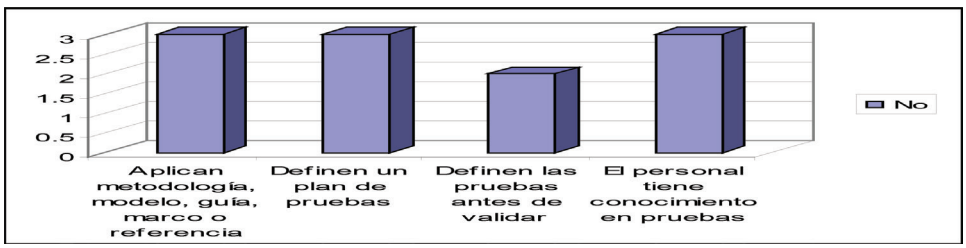

Figura 10 – Estado de la organización del proceso de validación

En cuanto a la variable 2 (pruebas en la organización) (Fig. 11), el 100% señala que no aplican pruebas según un estándar de calidad, no utilizan técnicas de prueba y menos herramientas de prueba y no establecen un entorno de validación. Sin embargo, en la (Fig. 12), el 100% indica que aplican pruebas específicas al proyecto, muchas veces,

Artigos RISTI N17.indd 34 5/12/2016 1:22:07 PM

pero no siempre. Un 66% aplica un plan de prueba según los requisitos del usuario y proyecto. Esto contradice el estado de la organización en cuanto al proceso de validación, ya que, indican que no aplican un plan de prueba. Lo que significa que hacen pruebas de validación y consideran que esto es un plan de pruebas, porque cubren las entradas de datos del proyecto y de esta forma comprueban la funcionalidad, pero esto no significa que aplican un plan de prueba específico al proyecto, que sería lo indicado.

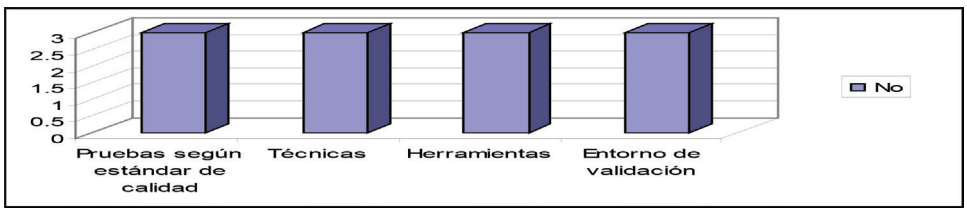

Figura 11 – Pruebas en la organización

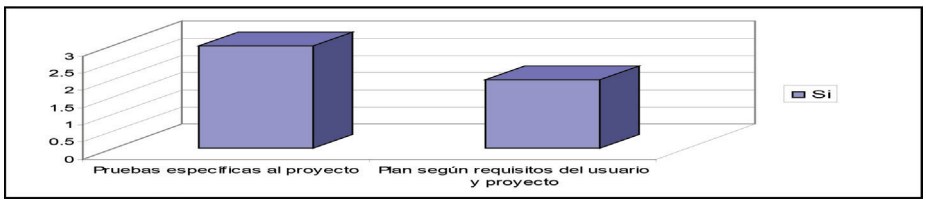

Figura 12 – Pruebas en la organización

Resultados de la aplicación del metamodelo

La validación del proyecto se realizó según lo especificado por el metamodelo, a través del plan de prueba y las plantillas de prueba aplicadas al proyecto Web. La prueba aplicada fue la funcional y se realizó de forma manual. La técnica utilizada fue la de partición de equivalencia. Previo a la aplicación de la misma se explicó a los tres analistas programadores en qué consistía la prueba y la técnica.

Los resultados de la validación indican que del total de requisitos probados con el metamodelo, un porcentaje significativo (36%) presentó algún tipo de error o fallos. Posterior a la validación del proyecto y obtenidos los resultados, se procedió a aplicar a los tres analistas programadores, una encuesta a través de un cuestionario para determinar la valoración sobre el metamodelo. Los resultados obtenidos a partir del cuestionario se muestran en la Tabla 2.

Artigos RISTI N17.indd 35 5/12/2016 1:22:07 PM

Nº	Preguntas	Si	No	1	2	3	4	5
1	El metamodelo facilitó el desarrollo de las pruebas.							
2	Las pruebas organizadas en el plan son las apropiadas al proyecto.							
3	El metamodelo le proporcionó técnicas, pruebas, herramientas y criterios de pruebas antes no considerados.							
4	Se detectó ausencia de requisitos, errores, fallos o defectos con el plan de pruebas aplicado.							
5	Con las pruebas aplicadas se dispone de un producto que satisface al usuario.							
6	Utilizaría el metamodelo para validar otros proyectos.							
7	El plan de prueba del metamodelo permite contar con un proyecto fiable, confiable y seguro.							
8	El aplicar el plan de prueba garantizó la calidad del producto.							
9	La aplicación del metamodelo fue satisfactoria en cuanto a los resultados obtenidos a partir de las plantillas..							
10	El metamodelo permitió el aprendizaje durante el desarrollo de las pruebas.							

Tabla 2 – Cuestionario para evaluar el metamodelo

Los resultados obtenidos permiten establecer y aplicar las pruebas específicas de un proyecto para comprobar que los requisitos del usuario, han sido satisfechos en cuanto a su funcionalidad, tal como se había planteado acerca del metamodelo y su relación con las prácticas de validación. Estos resultados se han dividido y agrupado en dos partes. La primera parte abarca las preguntas sobre la aplicación del plan de prueba y la segunda es sobre la aplicación del metamodelo.

En la (Fig. 13) se observa que el plan de prueba para proyecto Web, proporcionado por el metamodelo es apoyado en un 100% por los probadores. Un alto porcentaje (66%) considera que muchas veces el plan le proporciona las pruebas acordes al proyecto, y un 34% considera que siempre. Iguales resultados se obtuvieron, al considerar si el plan les permitió detectar ausencia de requisitos, fallos, errores o defectos. Por otro lado, el 66 % considera que el plan de pruebas permite satisfacer los requisitos del usuario casi siempre y el 34% que siempre. Con respecto a la calidad del producto, el 66% considera que casi siempre con el plan se garantiza la calidad del producto y un 34% que muchas veces.

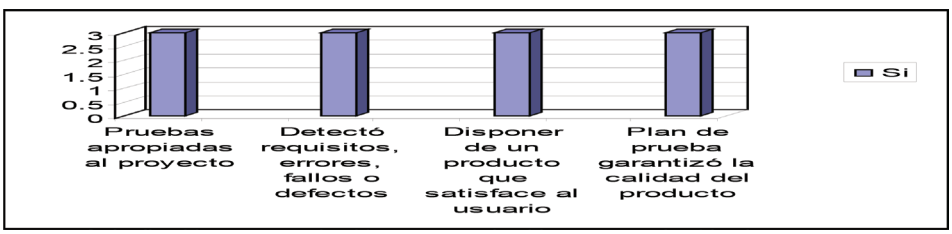

Figura 13 – Aplicación del plan de prueba

Artigos RISTI N17.indd 36 5/12/2016 1:22:08 PM

Aplicación del metamodelo

La aplicación del metamodelo resultó muy positiva y de mucha ayuda. Como se observa en la Fig. 14 y Fig. 15, los probadores en un 100% aprueban el metamodelo por los aspectos siguientes: facilita las prueba, proporciona técnicas de pruebas, herramientas y criterios de pruebas antes no considerados, lo utilizarían para validar otros proyectos, permite contar con un proyecto fiable, confiable y seguro, satisfacción en cuanto a los resultados obtenidos y permitió el aprendizaje durante el desarrollo de las pruebas. Al igual que en el caso anterior un alto porcentaje (66%) considera que el metamodelo facilita las pruebas para un proyecto. Con relación a si proporcionó técnicas, pruebas, herramientas y criterios de pruebas, muchas veces obtuvo el 66%. El 66% de los participantes utilizaría el metamodelo para validar otros proyectos. Este mismo porcentaje señala que "casi siempre" el metamodelo permite contar con un proyecto fiable, confiable y seguro, de satisfacción en cuanto a los resultados obtenidos y que permitió el aprendizaje durante el desarrollo de las pruebas.

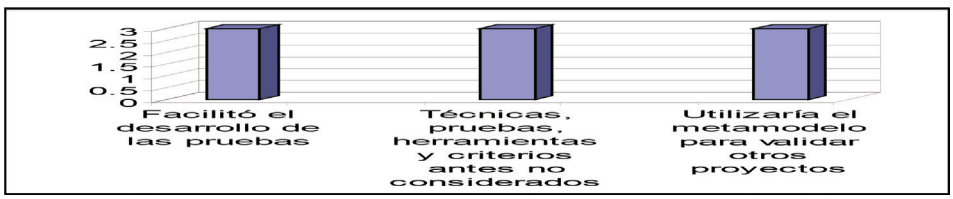

Figura 14 – Aplicación del metamodelo.

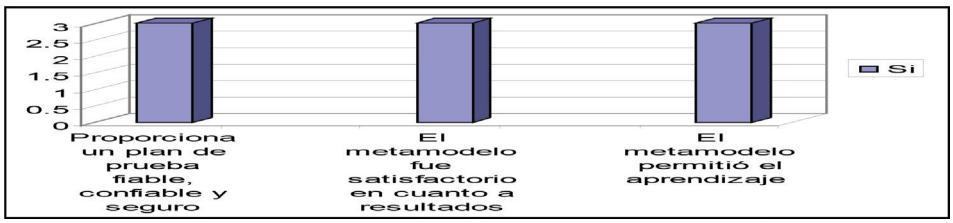

Figura 15 – Aplicación del metamodelo.

5. Conclusión

Se implementó una aproximación basada en metamodelado del área de proceso de Validación del CMMI DEV 1.3, específicamente la meta SG1 preparar la validación. Esta meta incluye las prácticas específicas SP 1.1, SP1. 2 y SP 1. 3. El metamodelo se diseñó en base a la meta preparar la validación, y está conformado por una taxonomía de proyectos de software, un método para la caracterización de pruebas de software, plantillas de pruebas y el plan de pruebas que establece las pruebas para un proyecto específico.

Se aplicó la técnica de estudio de casos para evaluar el metamodelo, mediante un caso práctico, en donde se determinó de forma inicial, que la organización donde se realizó la experimentación del metamodelo, carece de la estructura para preparar la validación y

Artigos RISTI N17.indd 37 5/12/2016 1:22:08 PM

por lo tanto, de un plan de prueba que establezca las pruebas para un proyecto software. Además, se constató la ausencia en la cultura de validación formal de sus proyectos y de la documentación del proceso de validación. El personal en general carece de formación y de conocimiento en pruebas o desconoce de las mismas. Inclusive la organización adolece de una política y estrategia para validación de sus proyectos.

El metamodelo proporciona una guía sistemática y organizada, para preparar la validación, a pesar, de la falta de formación y de conocimiento en pruebas de validación por parte de los probadores. Es importante resaltar que la organización donde se aplicó el estudio es pequeña y por lo tanto, los hallazgos y resultados obtenidos se basan en el contexto de la misma. Los resultados obtenido con el estudio de casos, confirman que los probadores comprobaron que el metamodelo les aporta las pruebas específicas y requeridas para el desarrollo de los procesos de validación. Además, les permite preparar la validación con los requisitos que corresponden a un proyecto determinado, y así ofrecer un software que cumpla con la calidad y con los requisitos del cliente.

Agradecimientos

Este trabajo ha sido patrocinado por everis Aeroespacial y Defensa, y la Universidad Politécnica de Madrid, a través de la Cátedra de Mejora de Procesos de Software en el Espacio Iberoamericano.

Referencias

Anderson, J. (2004) TPI a model for test process improvement. Seminar. University of Helsinki, Helsinki – Finland.

Ballou, M. (2008). Improving Software Quality to Drive Business Agility Sponsored by: Coverity Inc. Retrieved from http://www.coverity.com/library/pdf/IDC_Improving_Software_Quality_June_2008.pdf

Calvo-Manzano, J., Gonzalo Cuevas, A., San Feliu Gilabert, T. & Jiménez Puello, J. (2015). A Taxonomy for Software Testing Projects. Sistemas e Tecnologias de Informação, Atas da 10ª Conferência Ibérica de Sistemas e Tecnologias de Informação. Águeda, Portugal. AISTI, Universidade de Aveiro Vol. I Artigos Tomo 1, pp 289–294. Editores Álvaro Rocha, Arnaldo Martins, Gonçalo Paiva Dias, Luís Paulo Reis, Manuel Pérez Cota.

Calvo-Manzano, J., Gonzalo Cuevas, A., San Feliu Gilabert, T. & Jiménez Puello, J. (2015). Characterization of tests focused on validation of software Project. 22nd EuroAsiaSPI Conference. EuroSPI, Turkish Standards Institute, Ankara, Turkey.

Camargo, K.G., Ferrari, F.C. & Fabbri, S. CPF. (2015). Characterising the state of the practice in software testing through a TMMi-based process. Journal of Software Engineering Research and Development, 3:7.

Causevic, A., Sundmark, D. & Punnekkat, S. (2010). An Industrial Survey on Contemporary Aspects of Software Testing. Software Testing Verification and Validation *(ICST)*. Third International Conference, pp. 393–401. doi: 10.1109/ICST.2010.52

Artigos RISTI N17.indd 38 5/12/2016 1:22:08 PM

Cooter, M. (2008). Poor software testing hits companies in the pocket Techworld. Retrieved from http://www.infoworld.com/d/developer-world/poor-software-testing-hits-companies-in-pocket-305

Davis, F. & Venkatesh, V. (2004). Toward Preprototype User Acceptance Testing of New Information Systems: Implications for Software Project Management. IEEE Transactions on Engineering Management, vol. 51, No. 1.

Desai, A. & Shah, S. (2011). Knowledge management and software testing. Proceedings of the International Conference & Workshop on Emerging Trends in Technology (ICWET'). ACM, New York, pp. 767–770.

Elhag, A.A.M.; Elshaikh, M.A.; Mohamed, R. & Babar, M.I. (2013). Problems and future trends of software process improvement in some Sudanese software organizations. Computing, Electrical and Electronics Engineering (ICCEEE), International Conference, pp. 263–268. doi: 10.1109/ICCEEE.2013.6633945

Faizan, M., Ulhaq, S. & Khan, M.N.A.(2014). Defect Prevention and Process Improvement Methodology for Outsourced Software Projects. Middle-East Journal of Scientific Research 19 (5): 674–682. doi: 10.5829/idosi.mejsr.2014.19.5.13669

Faroog, A. & Dumke, R. (2007). Research Directions in Verification & Validation Process Improvement. ACM SIGSOFT Software Engineering Notes, vol. 32 (4), pp. 1–3.

García Guzmán, J., de Amescua Seco, A. & Velasco de Diego, M. (2006). Top 10 De Problemas Relativos a la Mejora del Proceso de Verificación y Validación en Organizaciones Intensivas en Software. Taller sobre Pruebas en Ingeniería del Software (PRIS).

Hernandez, Y., King, M., Pava, J. & Clarke, P. (2008). A Meta-Model to Support Regression Testing of Web Applications. Retrieved from http://users.cis.fiu.edu/~clarkep/research/areas/REUPubs/HernandezKingPavaClarke.pdf

Humphrey, W. S., Konrad, M. D., Over, J. W. & Peterson,W. C. (2007). Future directions in process improvement. Crosstalk, The Journal of Defense Software Engineering, vol. 20, No. 2.

Iyengar, P. & Karamouzis, F. (2007). Offshore Application Testing Drives Greater Business Value. Gartner Research, pp. 1–3.

Jäntti, M. (2008). Difficulties in Managing Software Problems and Defects. Joka Kuopio. Kuopion y Liopiston Julkaisuja H. Informaatioteknologia jakauppatieteet 11 Kuopio University Publications H. Business And Information Technology 11 Doctoral dissertation. Retrieved from http://wanda.uef.fi/uku-vaitokset/vaitokset/2008/isbn978-951-781-990-9.pdf

Kerry, E. & Delgado, S. (2009). Applying software engineering practices to produce reliable, high-quality and accurate automated test systems. AUTOTESTCON, IEEE, pp.69–71. doi: 10.1109/AUTEST.2009.5314054

King, L. (2008). Computerworld UK. Businesses fear lost revenues after poor software testing. Retrieved from http://www.infoworld.com/d/developer-world/businesses-fear-lost-revenues-after-poor-software-testing-134

Artigos RISTI N17.indd 39 5/12/2016 1:22:08 PM

Kurokawa, T. & Shinagaw M. (2008). Technical Trends and Challenges of Software Testing. Science & Technology Trends Quarterly Review, 29, pp. 34–45.

Lévesque, M. (2009). A Metamodel of Unit Testing for Object-Oriented Programming Languages. Retrieved from http://arxiv.org/pdf/0912.3583.pdf

Marin, B., Vos, T. Giachetti, G., Baars, A. & Tonella, P. (2011). Towards testing future Web applications, *Research Challenges in Information Science,* pp. 1–12.

Méndez, E., Pérez, M. A., Domínguez, K. & Mendoza, L. E. (2010). SAM: Sistema Automatizado del Método MECAP para Especificar Casos de Prueba. RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação, (6), pp. 45–58.

Monteiro, P., Machado, R. & Kazman, R. (2009). Inception of Software Validation and Verification Practices within CMMI Level 2. ICSEA, pp. 536–541.

Muñoz, M. (2003). SCMM: Metamodelo para la Creación de Aplicaciones en Redes de Siguiente Generación. Tesis Doctoral. Universidad Carlos III De Madrid, Departamento De Ingeniería Telemática, Doctorado en Tecnologías de las Comunicaciones, Leganés.

Muñoz, M., Gasca, G. & Valtierra, C. (2014). Caracterizando las Necesidades de las Pymes para Implementar Mejoras de Procesos Software: Una Comparativa entre la Teoría y la Realidad. RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação, no.spe1, pp. 1–15.

Nayyar, M. & Rizwan, J. (2009). Improvement of Key Problems of Software Testing in Quality Assurance. Sci.Int.(Lahore). 21(1), pp. 25–28. Retrieved from http://arxiv.org/ftp/arxiv/papers/1202/1202.2506.pdf

Oh, H., Choi, B., Han, H. & Wong, W. (2008). Optimizing Test Process Action Plans by Blending Testing Maturity Model and Design of Experiments. QSIC. The Eighth International Conference on Quality Software, pp. 57–66.

Piattini, M., García, F., García Rodriguez, I. & Pino, F. (2014). Calidad de Sistemas de Información. Segunda edición, Madrid: Rama, 2014.

Quillen, R. (2011).Why Some Software Development Projects Fail and What You Can Do About It. Quillen Infrastructure Technologies. Retrieved from http://www.quillenit.com.au/docs/Why_Software_Development_Projects_Fail.pdf

Rajamani, S. (2006). Automatic Property Checking for Software: Past, Present and Future, *Software Engineering and Formal Methods, p*p. 18–20.

Sadilek, D. & Weißleder, S. (2008). Testing Metamodels. Model Driven Architecture Foundations and Applications Lecture Notes in Computer Science, vol. 5095, pp 294–309.

Wohlin, C., Höst, M. & Henningsson, K. (2003). Empirical Research Methods in Software Engineering. ESERNET, pp. 7–23.

Artigos RISTI N17.indd 40 5/12/2016 1:22:08 PM