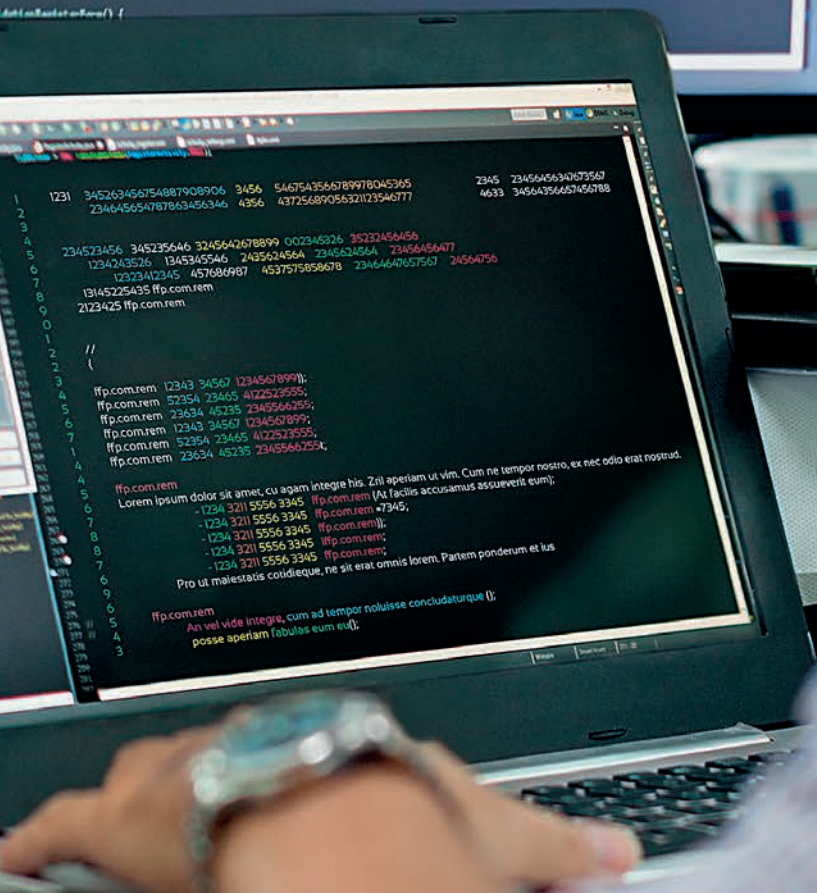




La industria del software tiene el desafío de

construir una nueva generación de sistemas

capaces de proveer servicios en la nube, movilidad y servicios físico-cibernéticos en dispositivos inteligentes (Smart).



Utilización de la técnica QFD en una arquitectura de procesos de software

Using the QFD technique in a software process architecture

RESUMEN

Varios modelos del Despliegue de la Función de Calidad (QFD) han sido propuestos para la industria del software como técnica de gestión de requerimientos. Sin embargo, dichos modelos describen las actividades y productos de trabajo de la técnica sin explicar cómo utilizarla en procesos ya definidos para un ambiente de producción de software específico. El objetivo de este trabajo es presentar un método para la inserción de la técnica QFD en una arquitectura de procesos de software basada en el Modelo Integrado de Madurez de Capacidades para Desarrollo (CMMI-DEV). El resultado de la aplicación del método fue la actualización del objeto proceso de gestión de requerimientos según los elementos procedimentales de la técnica QFD.

ABSTRACT

Different Quality Function Deployment (QFD) models have been proposed as a requirements management technique for software industry. Although these models describe the activities and work products of the technique, they don't detail how to apply QFD in a defined process for a given software production environment. The aim of this paper is to present a method for integrating QFD in a software process architecture based on the Capability Maturity Model Integration for Development (CMMI-DEV). Applying this method, the requirements management of process objects was updated according to the procedural elements of QFD.

INTRODUCCIÓN

La industria del software tiene el desafío de construir una nueva generación de sistemas capaces de proveer servicios en la nube, movilidad y servicios físico-cibernéticos en dispositivos inteligentes (Smart). Estos sistemas operan en ambientes complejos de negocio en los cuales la información y la infraestructura de diferentes organizaciones están interconectadas a gran escala. En este contexto, requerimientos de calidad como la resiliencia, usabilidad, privacidad y seguridad son mandatorios. Por ello, no basta con que la gestión de la calidad del software se ocupe solo de atender los requerimientos funcionales y disminuir la cantidad de errores, fallas y defectos en operación. La misión del gestor de la calidad del software es garantizar que los sistemas entreguen el valor esperado al negocio [1], por lo que es necesario integrar la gestión de la calidad con la gestión de producto y la gestión de la tecnología de información (TI) [2].

La agilidad y calidad requeridas en el desarrollo de esos sistemas modernos de software hacen que las empresas de la industria del rubro sean entendidas como sistemas vivos y adaptativos, que deben ser capaces de responder a cambios como la adaptación de la estructura y operación de los procesos para integrar ambientes de producción, a la adopción de buenas prácticas de los modelos de calidad o la automatización de los procesos [3].

Para mejorar los procesos del software como estrategia de gestión de la calidad, son utilizados modelos de procesos tales como ISO 15504 [4], ISO 29110 [5], el Modelo de Procesos para la Industria del Software (MoProSoft) de la Universidad Nacional Autónoma de México (UNAM) [6], el Modelo de Mejoría de Pro-



Palabras Clave

Arquitectura de procesos de software, QFD, SQFD, mejora de procesos de software.

Key words

Software Process Architecture, QFD, SQFD, Software Process Improvement.

ceso del Software Brasileiro (MPS.br) de la Softex [7] y el CMMI (Capability Maturity Model Integrator) del Instituto de Ingeniería de Software (SEI, Software Engineering Institute) [8].

Los modelos de procesos de software son conjuntos de buenas prácticas que indican “qué” hacer y dejan a las empresas definir “cómo” hacer [9]. En la práctica, esos modelos son adaptados a ambientes de producción de software específicos según los requerimientos del negocio de cada empresa [9]. Como resultado de la adaptación de uno [10] o varios de dichos modelos simultáneamente [11], se definen los procesos del ambiente de producción de software, conceptualmente representados por una Arquitectura de Procesos de Software (APS). Es a través de la APS que los procesos, sus interfaces y relaciones son proyectados, visualizados y mantenidos [12].

Precisar la participación de cada equipo de un mismo proyecto de software, además de la interacción entre ellos, es un factor crítico de éxito en la mejora de procesos de software [13]. Por ello, la APS debe describir tanto la estructura como la operación de los procesos. Especificar dicha operación comprende la selección e inserción de los métodos, técnicas y herramientas de ingeniería de software que van a componer una APS. También algunas prácticas originarias de la manufactura, por ejemplo, Kanban [14], Kaizen [15] y QFD [16], han sido adaptadas y utilizadas para mejorar los procesos de software.

Concretamente, el QFD (Quality Function Deployment) ha sido aplicado en la industria del software como técnica para la gestión de requerimientos [16] [17], garantía de la calidad del producto, análisis de costo [18] e incluso como herramienta para la adaptación del modelo de calidad CMMI dentro del mejoramiento de procesos de un ambiente de producción del software [9] [19]. En estas aplicaciones se presentaron las actividades y productos generados por el QFD, pero no se detalla cómo el QFD puede ser usado por los procesos de la organización.

Dado que la APS es un área de investigación en evolución y experimentación, y para facilitar el uso de técnicas en ambientes de producción de software, este artículo presenta un método para insertar la técnica QFD en una APS con el objetivo de mejorar la calidad de los procesos de software.

Este trabajo utiliza el Design Science Research (DSR) [20] como método de investigación. Para eso, se entendió el conocimiento más moderno en APS por medio de una investigación científica inspirada en su aplicación práctica. El contexto para el DSR (escenario de aplicación) es un ambiente de producción de software operado por los estudiantes del quinto año del curso de Ingeniería de Computación de la Escuela Politécnica da Universidad de São Paulo (POLI-USP), Brasil. Este ambiente posee una APS basada en el segundo nivel de madurez del modelo CMMI-DEV. Para aumentar la calidad de los procesos se propuso insertar el QFD en la APS. El artefacto para el DSR fue el método de inserción de técnicas en la APS y el tratamiento para el DSR fue la aplicación del método para insertar el QFD en la APS del ambiente de producción de la POLI-USP.

En la sección Fundamentos se explican los conceptos de arquitectura de procesos, objeto proceso y QFD en la industria del software. La sección Metodología expone el método de inserción de técnicas en APS, y la sección Aplicación del Método muestra los resultados de dicha aplicación en el contexto ya introducido. Finalmente, la última sección presenta las conclusiones.

FUNDAMENTOS

El modelo de Fábrica Integrada del Software (FIS) [21] presenta un enfoque de arquitectura corporativa para empresas de esta industria. El modelo FIS es representado por los niveles de jerarquía estratégica, gerencial y operacional; las unidades de negocio de la empresa, entre ellas el ambiente de producción de software; y, las vistas arquitecturales de la empresa definidas según los cinco puntos de vista del Modelo de Referencia para Procesamiento Abierto Distribuido (RM-ODP, del inglés Reference Model of Open Distributed Processing) [22]. Cada punto de vista tiene un dominio arquitectural determinado por una especificación que describe una proyección de la empresa de software (entendida como un sistema).

La Figura 1 muestra el modelo de FIS. El conjunto de todas las vistas, niveles de jerarquía y unidades de negocio conforman la arquitectura corporativa de la empresa de software.

La especificación obtenida a partir del punto de vista empresa del RM-ODP describe el propósito, las responsabilidades y actividades de la empresa. En el contexto de la arquitectura corporativa, dicha especificación debe proveer un modelo de la arquitectura del negocio [23], de la cual es parte la arquitectura de procesos (AP). Una AP debe contener los objetivos globales, principios generales, modelos y directrices de los procesos [12]. El alcance de este trabajo, destacado en el gris más oscuro en la Figura 1, son los elementos de la AP corporativa que representan conceptualmente los procesos del ambiente de producción del software, o sea, la APS.

La APS es un modelo conceptual para la definición y mantenimiento de los procesos asociados al ciclo de vida del software, de la relación entre esos procesos y de la relación con procesos de otras unidades de negocio de la organización. La APS debe estar alineada con los objetivos de negocio, con la estrategia y con la arquitectura corporativa de la empresa [11].

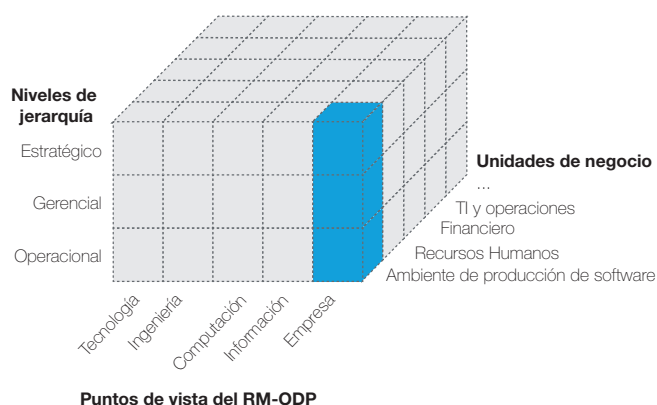


Figura 1: Visión corporativa de una empresa de software a través del modelo de Fábrica de Software Integrada.
Fuente: Adaptado de Borsoi, B., & Becerra, J. "A Method to Define an Object Oriented Software Process Architecture". 19th ASWEC, Perth, WA: IEEE. 2008.

Para definir una APS deben construirse sus abstracciones. Para ello, el ambiente de producción de la POLI-USP, que es el escenario de aplicación en este estudio, se procedió de la siguiente forma [10] [12] [24]:

- 1) En la *abstracción conceptual*, la estructura de un elemento proceso se define como un conjunto de actividades (variantes e invariantes), precondiciones (artefactos e información para ejecución del proceso), poscondiciones (resultados obtenidos), e interfaces de entrada y salida.
- 2) La *abstracción de referencia* está compuesta por las buenas prácticas establecidas en el modelo CMMI para el nivel 2 de madurez.
- 3) La *abstracción operacional* resulta de la instanciación de la abstracción de referencia considerando los requerimientos del negocio de la POLI-USP.
- 4) La *abstracción de proyecto* se deriva de la instanciación de la abstracción operacional teniendo en cuenta las características específicas del proyecto definido para cada curso.
- 5) La *abstracción de la base de conocimiento* es producto de la colección del histórico de uso de la APS en cada instancia del curso.

La Figura 2 es un modelo conceptual que representa la APS como el componente de la AP corporativa que define y gestiona los procesos del ambiente de producción de software. Es decir, los procesos asociados al ciclo de vida del producto software; la APS como parte de la vista arquitectural empresa del RM-ODP; los procesos categorizados en los niveles de jerarquía corporativos, conforme el modelo de FIS; los procesos definidos por la adaptación de normas y modelos de procesos de software según los requerimientos de negocio de cada empresa; y, la operación de los procesos especificada al nivel de proyecto por el uso de conceptos, metodologías, técnicas y herramientas de la ingeniería del software.

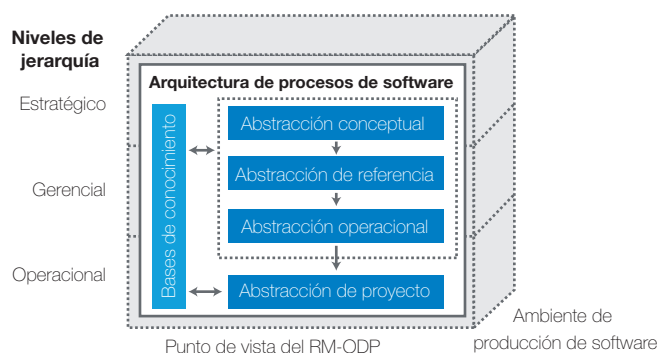


Figura 2 - Abstracciones de una Arquitectura de Procesos de Software. Fuente: Adaptado de Pesantes, M. H., Lemus, C., Mitre, H., & Mejía, J. "Software Process Architecture: Roadmap". *Ninth Electronics, Robotics and Automotive Mechanics Conference* (págs. 111 - 116). Cuernavaca: IEEE, 2012.

a) Objeto proceso

El elemento arquitectural de la APS es el objeto proceso (OP), o sea los procesos de la APS son representados como OP en término de su composición, comportamiento y relación de acuerdo con un dominio y un contexto específicos [24] [25].

Un OP, esquematizado en la Figura 3, es una entidad discreta que encapsula los atributos y actividades de un proceso y los ofrece a través de interfaces [21]. Los atributos del proceso son los papeles (asociados a las funciones de negocio del modelo

organizacional) y los artefactos (entradas y productos de trabajo de las actividades). El comportamiento del OP es definido en la especificación de las actividades y por la relación entre los OP. El OP opera a través de la acción de las actividades y el resultado de su comportamiento puede ser observado en la actualización de los artefactos.

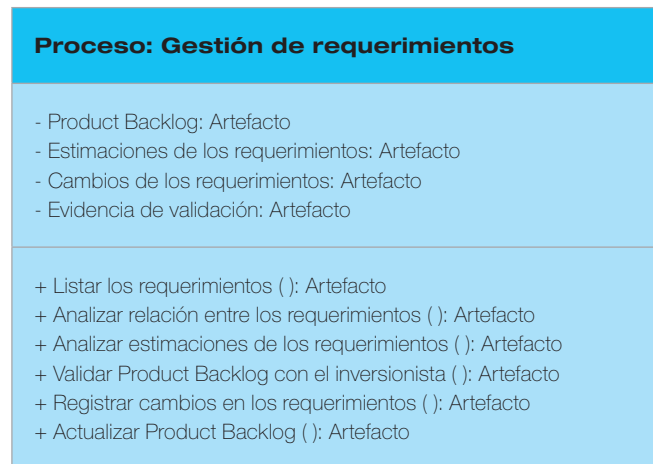


Figura 3 – Diagrama del objeto proceso gestión de requerimientos mediante el UML. Fuente: Elaboración propia.

La Figura 3 muestra el objeto proceso gestión de requerimientos, que es de interés en este trabajo, pues es el elemento arquitectural en que el QFD fue insertado.

b) Despliegue de la Función de Calidad (QFD) en el desarrollo de software

El QFD fue concebido en la manufactura con el objetivo de traducir los requerimientos de los clientes en requerimientos técnicos apropiados para cada fase del desarrollo de un producto [26]. Actualmente, es también usado en todo el mundo en la mejora y desarrollo de servicios [9].

Entre los beneficios registrados por el uso del QFD se destacan la reducción en el número de alteraciones en un proyecto; el aumento de la calidad apreciada por los clientes [27] [28]; y una mejor comunicación entre equipos multidisciplinares [29].

El enfoque tradicional del QFD en la manufactura es un proceso de cuatro fases: Planeación del producto, diseño del producto, planeación del proceso y control del proceso [30]. Para cada fase se crea una matriz. La que corresponde a la primera de estas fases, llamada casa de la calidad (CDLC) QFD, es la principal. La Figura 4 presenta los elementos de dicha matriz:

- El "qué" lista de forma estructurada los requerimientos del cliente.
- El "peso" cuantifica la prioridad de los requerimientos del cliente.
- El "cómo" enumera los requerimientos técnicos del producto.
- Los "objetivos" definen la forma de medir y priorizar los requerimientos técnicos.
- La "correlación" sirve para identificar si los requerimientos técnicos se soportan o impiden entre ellos.
- La "interrelación" define la relación entre los requerimientos del cliente y los requerimientos técnicos.

Como muestra la Figura 4, el “cómo” y los “objetivos” de la primera matriz se tornan respectivamente el “qué” y el “peso” de la segunda matriz, y así sucesivamente [31].

La diferencia de un producto manufacturado con relación a un producto de software es que este último no es identificado por sus características físicas sino por su comportamiento, es decir, no es valorado por lo que es sino por lo que hace y cómo lo hace.

En el software es necesario distinguir entre requerimientos del negocio del cliente (quién usa y para qué), requerimientos funcionales (funciones del producto) y requerimientos no funcionales (características de calidad). Ello implica que cuando se aplica al desarrollo de productos de software, el QFD tradicional de la manufactura necesita ser adaptado [32] [33].

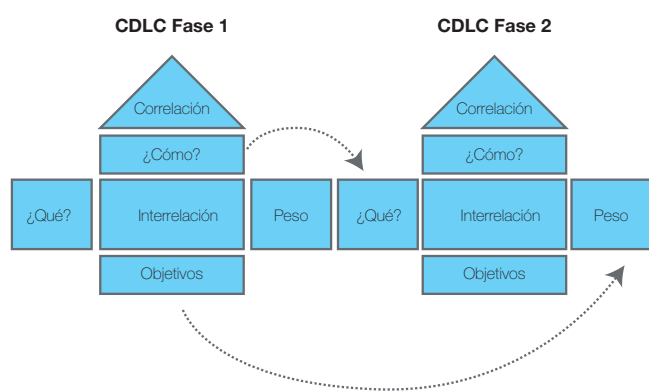


Figura 4 - Casa de la calidad del QFD y desdoblamiento entre las fases. Fuente: Adaptado de Herzwurm, G., Schockert, S., & Pietsch, W. "QFD for Customer-Focused Requirements Engineering". *International Requirements Engineering Conference* (págs. 330 - 338). IEEE. 2003.

Con base en los modelos de Zultner [34], Shindo [35] y Ohmris [36], Herzwurm propuso el modelo PriFo QFD para software [16], también conocido como Ingeniería de Requerimientos Conjunta por sugerir reuniones interdisciplinarias para construir las matrices del QFD. En el modelo PriFo, representado en la Figura 5, la matriz clásica del QFD de la fase de planeación del producto es dividida en dos:

- 1) CDLC clásica: en la que el “cómo” clásico de la manufactura es substituido por los requerimientos no funcionales e;
- 2) CDLC del software: en la cual el “cómo” clásico de la manufactura es reemplazado por los requerimientos funcionales.

En este modelo, la matriz de la fase de diseño del producto es construida colocando los requerimientos funcionales en el “qué” y los requerimientos no funcionales en el “cómo”. En la “interrelación” entre requerimientos funcionales y no funcionales se registran puntos que justifican las decisiones arquitecturales. El modelo PriFo no considera las dos últimas fases del QFD: planeación del proceso y control del proceso.

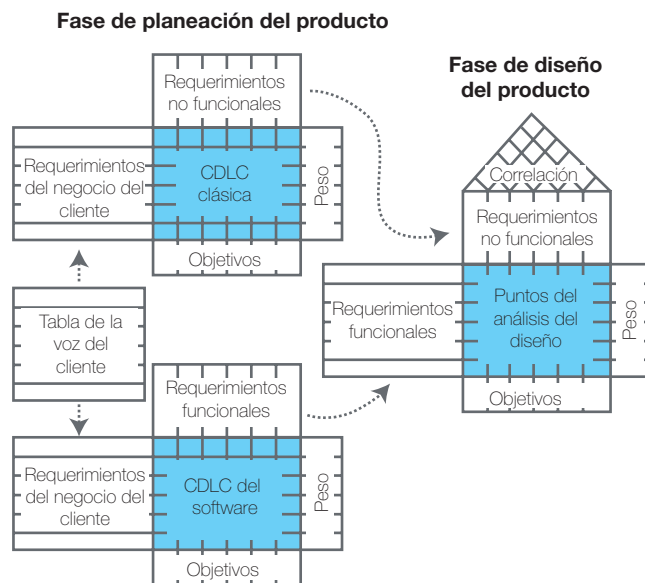


Figura 5 - Modelo PriFo QFD para software adaptado al contexto de este trabajo. Fuente: Adaptado de Herzwurm, G., Schockert, S., & Pietsch, W. "QFD for Customer-Focused Requirements Engineering". *International Requirements Engineering Conference* (págs. 330 - 338). IEEE. 2003.

METODOLOGÍA

En la construcción de una APS, modelos de procesos abstractos son transformados en modelos de procesos ejecutables. Esta transformación es conocida como instanciación o “tailoring” de procesos. La instanciación sucede cuando es preciso especializar los procesos para tener en cuenta requerimientos específicos de un nuevo contexto que necesita ser descrito [10].

Dias propuso un método de instanciación de APS, el cual está compuesto de tres fases: especificación de los procesos, modelado de los procesos y proyecto de los procesos. El autor utilizó en su método el objeto proceso [21] como elemento arquitectural.

Es importante resaltar que, cuando se utiliza el método Dias en la instanciación de una APS del nivel de abstracción de referencia para el nivel de abstracción operacional, se considera una empresa específica como contexto y los requerimientos de los procesos son los requerimientos de negocio de la empresa. En otro contexto, el empleo de la instanciación en una APS del nivel de abstracción operacional para el nivel de abstracción de proyecto supone un proyecto específico como contexto y los requerimientos de los procesos son los requerimientos de aquel proyecto.

MÉTODO DE INSERCIÓN DE TÉCNICAS EN ARQUITECTURAS DE PROCESOS DEL SOFTWARE

La inserción de una técnica en una APS es entendida como una forma de instanciación de la misma para una nueva versión de la APS. El método para insertar la técnica QFD en una APS usa una adaptación del método del Dias¹ (Figura 6).

¹ El método adaptado de Dias, utilizado para la inserción de la técnica QFD en el contexto de la APS, genera una nueva versión de la APS.

Dicha adaptación se hizo del siguiente modo:

- Considera la propia APS como contexto de la instanciación y la utilización de la técnica QFD como requisito del proceso.
- La APS es instanciada sin una transformación entre niveles de abstracción. No es necesaria la actividad "organizar modelos en niveles de abstracción" en la fase de proyecto de los procesos del método Dias.
- En vez de identificar los requerimientos de los procesos, son identificados los elementos procedimentales de la técnica QFD, o sea las tareas y productos de trabajo que deben ser insertados en el contexto de la APS.
- La APS a adaptar se relaciona con los requerimientos de los procesos sino con los elementos procedimentales de la técnica.
- No son consideradas las actividades de validación del método de Dias pues el único requisito del proceso es la utilización de la técnica QFD.

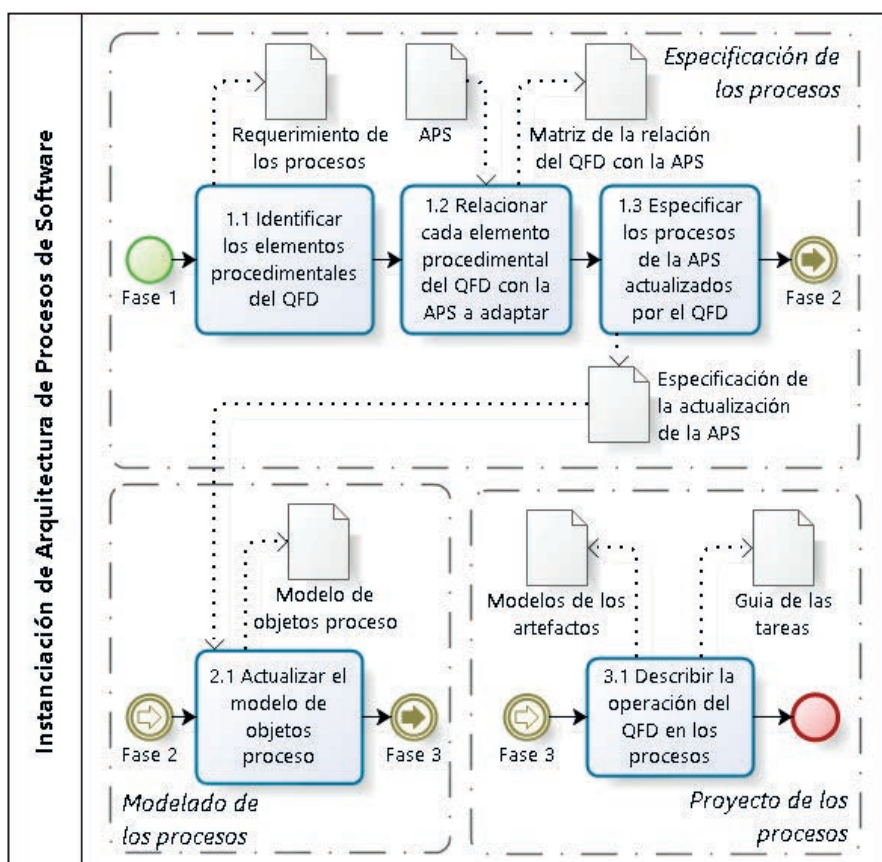


Figura 6. Método de instanciación de APS adaptado para inserción del QFD
 Fuente: Adaptado por Dias, L. D. "Método de Instanciación de uma Arquitetura de Processos Aplicado em Fábrica de Software". Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção de título de Mestre em Engenharia. São Paulo, 2010.

APLICACIÓN DEL MÉTODO

a) Escenario de aplicación

El escenario de aplicación del método es el ambiente de producción de software del Laboratorio de Tecnología de Software (LTS) de la Escuela Politécnica de la Universidad de São Paulo (POLI-USP). Este ambiente es operado por los estudiantes del quinto año del curso de Ingeniería de Computación para el desarrollo del prototipo de un producto innovador. La Figura 7 muestra la APS del LTS en el nivel de abstracción operacional.

En dicha figura se utilizó el diagrama de objetos UML para representar gráficamente los OP y la relación entre ellos. A pesar de haber quedado ocultos, cada OP posee sus atributos y actividades de la misma forma presentada en la Figura 3. Como la APS del LTS fue definida sobre la base del CMMI DEV nivel 2 de madurez, en la instanciación de los procesos se decidió que los nombres de los OP correspondieran a las áreas de proceso del CMMI DEV, con excepción del área de procesos de garantía de la calidad que fue integrada en el OP de monitoreo y control del proyecto.

b) Resultados de la aplicación del método

Los cuadros 1 y 2 describen parte del resultado de la fase Especificación de los procesos del método de instanciación de APS adaptada para la inserción del QFD considerando solo el OP gestión de requerimientos. El resultado total incluye la repetición de ambos cuadros para todos los OP de la APS del LTS relacionados con el QFD.

Cada columna corresponde al resultado de una actividad de la primera fase del método, del siguiente modo:

- Las columnas Artefactos del QFD del Cuadro 1 y Tareas del QFD del Cuadro 2 fueron rellenas en la actividad 1.1 Identificar los elementos procedimentales del QFD del método de la Figura 6.
- Las columnas Artefactos del OP gestión de requerimientos del Cuadro 1 y Actividades del OP gestión de requerimientos del Cuadro 2 fueron rellenas en la actividad 1.2 Relacionar los elementos procedimentales del QFD con la APS a adaptar del método de la Figura 6.
- La columna Acción a ser tomada del Cuadros 1 y la columna Acción a ser tomada del Cuadro 2 fueron rellenas en la actividad 1.3 Especificar los procesos de la APS actualizados por el QFD del método de la Figura 6. Las posibles acciones a ser tomadas (en ambos cuadros) son: Expandir significa que el artefacto (Cuadro 1) o la actividad (Cuadro 2) del OP gestión de requerimientos precisa ser expandido para incluir los elementos procedimentales del QFD que están relacionados; Mantener significa que el artefacto (Cuadro 1) o la actividad (Cuadro 2) del OP gestión de requerimientos ya incluye un elemento procedimental equivalente al del QFD y entonces el OP no será alterado; Adicionar indica que no hay artefactos (Cuadro 1) o actividades (Cuadro 2) en el OP gestión de requerimientos que se relacionen con ningún elemento procedimental del QFD y entonces será adicionado al OP.

Artefactos del QFD	Artefactos del OP gestión de requerimientos	Acción a ser tomada
Lista de requerimientos de negocio en el lenguaje del cliente	Lista de requerimientos de negocio	Adicionar
Peso de los requerimientos de negocio según el cliente	Product Backlog	Mantener
Lista de requerimientos funcionales	Product Backlog	Mantener
Casa de la calidad del software (matriz de relación de los requerimientos de negocio en el lenguaje del cliente con requerimientos funcionales)	Casa de la calidad del software	Adicionar
Casa de la calidad del QFD	Casa de la calidad QFD	Adicionar

Cuadro 1. Matriz de relación de los artefactos del QFD con el objeto proceso gestión de requerimientos
Fuente: Elaboración propia.

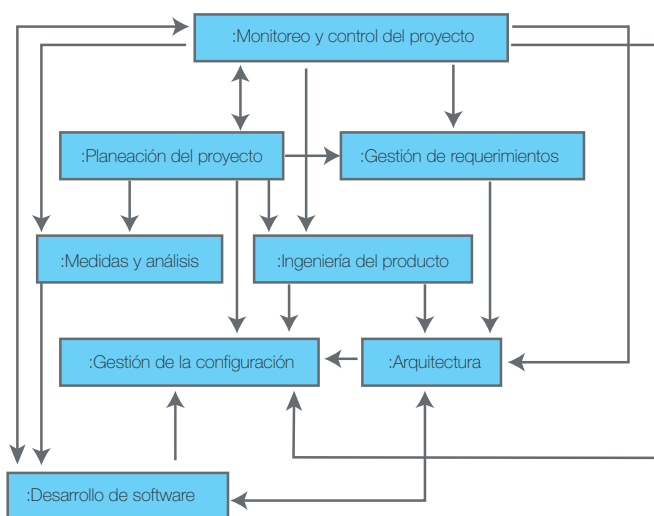


Figura 7. Arquitectura de objetos proceso de software del laboratorio de tecnología de software LTS
Fuente: Elaboración propia.

Tareas del QFD	Actividades del OP Gestión de requerimientos	Acción a ser tomada
Listar los requerimientos de negocio en el lenguaje del cliente	Listar los requerimientos	Expandir
Asociar para cada requerimiento del cliente un peso que indique su prioridad	Listar los requerimientos	Expandir
Priorizar los requerimientos del cliente	Listar los requerimientos	Expandir
Listar los requerimientos funcionales	Listar los requerimientos	Expandir
Definir una métrica para evaluar cada requerimiento funcional y establecer un valor como objetivo la ser alcanzado	Analizar las estimaciones de los requerimientos	Expandir
Agrupar los requerimientos funcionales según los componentes del sistema que los implementan	Analizar la interrelación de los requerimientos	Expandir

Relacionar los requerimientos del producto en el lenguaje del cliente y los requerimientos funcionales. Para cada relación es definido un valor que indica el nivel de la relación	Analizar la relación de los requerimientos	Expandir
Priorizar los requerimientos funcionales según el peso de los requerimientos del cliente y el objetivo de los requerimientos funcionales	Analizar la relación de los requerimientos	Expandir
Listar los requerimientos no funcionales	Listar los requerimientos	Expandir
Definir una métrica para evaluar cada requisito no funcional y fijar un valor como objetivo a ser alcanzado	Analizar las estimaciones de los requerimientos	Expandir
Correlacionar los requerimientos no funcionales y determinar si entre ellos se soportan o anulan	Analizar la relación de los requerimientos	Expandir
Relacionar los requerimientos del producto en el lenguaje del cliente con los requerimientos funcionales	Analizar la relación de los requerimientos	Expandir

Cuadro 2. Matriz de relación de las tareas del QFD con el objeto proceso gestión de requerimientos
Fuente: Elaboración propia.

La Figura 8 presenta el resultado de la fase Modelado de los procesos del método de instanciación de APS adaptado para la inserción del QFD. En el OP de gestión de requerimientos actualizado se destacan dos artefactos del QFD que fueron adicionados como nuevos atributos del OP.

Gestión de requerimientos
<ul style="list-style-type: none"> - Product Backlog: Artefacto - Estimaciones de los requerimientos: Artefacto - Cambios de los requerimientos: Artefacto - Evidencia de validación: Artefacto - Casa de la calidad del software: Artefacto - Casa de la calidad del QFD: Artefacto
<ul style="list-style-type: none"> + Listar los requerimientos (): Artefacto + Analizar relación entre los requerimientos (): Artefacto + Analizar estimaciones de los requerimientos (): Artefacto + Validar Product Backlog con el inversionista (): Artefacto + Registrar cambios en los requerimientos (): Artefacto + Actualizar Product Backlog (): Artefacto

Figura 8. Objeto proceso gestión de requerimientos utilizando el QFD como técnica
Fuente: Elaboración propia.

Finalmente, en la fase de Proyecto de los procesos del método de instanciación de APS adaptado para la inserción del QFD, fueron elaboradas las instrucciones de operación de los OP

impactados por la inserción del QFD. Esas instrucciones sirven como guía para la operación de OP, es decir, para gestionar los requerimientos del proyecto.

CONCLUSIONES

De forma general, la adaptación del método de instanciación de APS como estrategia para la inserción del QFD es aplicable a cualquier otra técnica. Sin embargo, es indispensable tener definida una APS.

De la misma forma que un arquitecto de software utiliza los modelos arquitecturales para decidir en cuál modulo, componente y objeto debe implementar cada método o regla de negocio, el arquitecto de los procesos emplea la APS para identificar en cuál proceso, actividad y artefacto se precisan tomar acciones para insertar alguna técnica.

Una técnica puede ser insertada en una APS en el nivel de abstracción operacional o de proyecto. En el primer caso, el uso de la técnica es formalizado para todos los proyectos y en el segundo, solo para uno.

REFERENCIAS

- [1] Suma, V., Shubhamangala, R., & Rao, L. (2013). Customization of quality models in software projects to enhance the business value. *IEEE 3rd. International Advance Computing Conference (IACC)* (pp. 1479-1485). Ghaziabad, India.
- [2] Breu, R., Kuntzmann-Combelles, A., & Felderer, M. (2014). New perspectives on software quality. *IEEE Software*,31(1), 32-38.
- [3] Kandjani, H., Bernus, P., & Nielsen, S. (2013). Enterprise Architecture Cybernetics and the Edge of Chaos: Sustaining Enterprise as Complex System in Complex Business Environments. *46th Hawaii International Conference on System Sciences* (pp. 3858-3867). Maui, Hawaii.
- [4] International Organization for Standardization, & International Electrotechnical Commission (2004). *ISO/IEC 15504-4 - Information technology — Process assessment — Part 4: Guidance on use for process improvement and process capability determination*.
- [5] International Organization for Standardization, & International Electrotechnical Commission. (2011). *ISO/IEC 29110-4-1:2011 Software engineering -- Lifecycle profiles for Very Small Entities (VSEs) -- Part 4-1: Profile specifications: Generic profile group*.
- [6] Oktaba, H., Alquicira, C., Su, A., Martinez, A., Quintanilla, G., Ruvalcaba, M., López, F., Rivera, M., Orozco, M., Fernandez, Y., & Flores, M. (2005). *MoPROSoft: Modelo de Proceso para la Industria del Software (vol. 1.3)*.
- [7] Softex. (2012). *MPS.BR - Melhoria de Processo do Software Brasileiro: Guia Geral MPS de Software*. Recuperado de http://www.softex.br/wp-content/uploads/2013/07/MPS.BR_Guia_Geral_Software_2012-c-ISBN-1.pdf

- [8]** Software Engineering Institute. (2010). *CMMI® for Development, Version 1.3, Improving processes for developing better products and services*. Hascom AFB: Carnegie Mellon University.
- [9]** Sun, Y., & Liu, X. (2010). Business-oriented software process improvement based on CMMI using QFD. *Information and Software Technology*, 52 (1), 79-91.
- [10]** Dias, L. D. (2010). Método de Instanciação de uma Arquitetura de Processos Aplicado em Fábrica de Software (Tesis de maestría). Universidade de São Paulo, São Paulo, Brasil.
- [11]** Pesantes, M. H. (2015). *A Method to Design a Domain-Specific Software Process Architecture in a Multimodel Improvement Environment*. (Tesis doctoral). Centro de investigación de matemáticas, Zacatecas, Mexico.
- [12]** Pesantes, M. H., Lemus, C., Mitre, H., & Mejia, J. (2012). Software Process Architecture: Roadmap. *Ninth Electronics, Robotics and Automotive Mechanics Conference* (pp. 111 - 116). Cuernavaca, Mexico: IEEE.
- [13]** Niazi, M. (2015). A comparative study of software process improvement implementation success factors. *Journal of Software: Evolution and Process*, 27 (9), 700-722.
- [14]** Muhammad Ovais, A., Markkula, J., & Oivo, M. (2013). Kanban in software development: A systematic literature review. *39th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)* (pp. 9 - 16). Santander, España: IEEE.
- [15]** Estacio, B., Prikladnicki, R., Mora, M., & Notari, G. (2014). Software Kaizen: Using Agile to Form High-Performance Software Development Teams. *Agile Conference (AGILE)* (pp. 1-10). Kissimmee, Estados Unidos: IEEE.
- [16]** Herzwurm, G., Schockert, S., & Pietsch, W. (2003). QFD for Customer-Focused. Requirements Engineering. *International Requirements Engineering Conference* (pp. 330 - 338). Washington, USA: IEEE.
- [17]** Ioannou, G., Pramataris, K. C., & Prastacos, G. P. (2004). A Quality Function Deployment Approach to Web Site Development: Applications for Electronic Retailing. *Les Cahiers du Management Technologique*, 13 (3), 1-18.
- [18]** Kashyap, D., & Misra, A. K. (2013). Software Cost Estimation Using Particle Swarm Optimization in The Light of Quality Function Deployment Technique. *International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-8). Coimbatore, India.
- [19]** Cao, Y. (2013). Software Process Improvement Framework Based on CMMI Continuous Model Using QFD. *International Journal of Computer Science Issues*, 10 (3), 281-286.
- [20]** Wieringa, R. J. (2014). *Design Science Methodology for Information Systems and Software Engineering*. New York: Springer.
- [21]** Borsoi, B., & Becerra, J. (2008). A Method to Define an Object Oriented Software Process Architecture. *19th Australian Software Engineering Conference (ASWEC)* (pp. 650-655). Perth, Australia.
- [22]** International Organization for Standardization, & International Electrotechnical Commission. (1996). *ISO/IEC 10746-2:1996 . Information technology -- Open Distributed Processing -- Reference Model: Foundations*
- [23]** Kutvonen, L. (2008). Using the ODP Reference Model for Enterprise Architecture. *Eleventh International IEEE EDOC Conference Workshop (EDOCW'07)* (pp. 231-238). Munich, Alemania.
- [24]** Borsoi, B. T. (2008). *Arquitetura de Processo Aplicada na Integração de Fábricas de Software* (Tesis doctoral). Universidad de São Paulo, Sao Paulo, Brasil. Recuperado de <http://www.teses.usp.br/teses/disponiveis/3/3141/tde-25092008-093143/pt-br.php>
- [25]** Dominguez, L., Restrepo, J., Marquez, D., & Risco, J. (2012). Fundamentos de uma fábrica de software orientada a objetos processos. *Augusto Guzzo Revista Acadêmica*, 1(9), 53-61.
- [26]** Mehrjerdi, Y. Z. (2010). Applications and extensions of quality function deployment. *Assembly Automation*, 30 (4), 388 - 403.
- [27]** Carnevalli, J. A., Sassi, A. C., & Cauchick M. P. (2004). Aplicação do QFD no desenvolvimento de produtos: levantamento sobre seu uso e perspectivas para pesquisas futuras. *Gestão & produção*, 11(1), 33-49.
- [28]** de Borba Prá, F., & Cauchick, P. A. (2013). Evolução na aplicação do QFD: análise de publicações qualificadas em periódicos. *Exacta*, 11(1), 89-100.
- [29]** Loos, M. J., & Cauchick, P. A. (2014). Utilização do QFD no desenvolvimento de novos produtos: uma análise das publicações em periódicos nacionais. *Produto & Produção*, 15(3), 17-31.
- [30]** Bhatia, S. M., & Sharma, A. K. (2015). Research Methodology for the Development of Software Tool to Improve the Quality of Software Product using Fuzzy Quality Function Deployment. *International Journal of Software Computing and Testing*, 1(1), 9-13.
- [31]** Nicchiotti, G. (2014). Health monitoring requirements elicitation via House of Quality. *Aerospace Conference* (pp. 1 - 15). Big Sky, Montana.
- [32]** Hari, A., Kasser, J. E., & Weiss, M. P. (2007). How Lessons Learned from Using QFD Led to the Evolution of a Process for Creating Quality Requirements for Complex Systems. *Systems Engineering*, 10(1),45-63.
- [33]** Kuo, T. C. (2013). Mass customization and personalization software development: a case study eco-design product service system. *Journal of Intelligent Manufacturing*, 24(5), 1019-1031.

- [34] Zultner, R. E. (1994). Software quality function deployment - the North American experience. *Software Quality Concern for People. Proceedings of the Fourth European Conference on Software Quality* (pp. 143-158). Zurich, Suiza.
- [35] Shindo, H. (1999). Application of QFD to Software and QFD Software Tools. *Fifth International Symposium on Quality Function Deployment and the First Brazilian Conference on Management of Product Development*. Belo Horizonte, Brasil.
- [36] Ohmori, A. (1994). Software Quality Deployment Approach: Framework Design, Methodology and Example. *Software Quality Journal*, 3(4), 209-240.

ACERCA DE LOS AUTORES

Juan Felipe Restrepo Naranjo

Es ingeniero electrónico de la Universidad de Antioquia, Colombia. Actualmente, cursa la maestría en Ingeniería de Computación adjunto al Grupo de Arquitectura y Fábrica de Software de la USP, Brasil. Tiene experiencia en desarrollo de software, definición de ambientes de producción de software, BPM y arquitectura corporativa.

 felipe.restrepo@usp.br

Jorge Luis Risco Becerra

Es profesor doctor del Departamento de Ingeniería de Computación y Sistemas Digitales de la Escuela Politécnica de la Universidad de São Paulo, Brasil. En la actualidad realiza investigaciones en producción de software centrada en arquitectura, arquitectura corporativa y ambientes inteligentes.

 jorge.becerra@usp.br

Ana Claudia Rossi

Es científica de la computación y tiene maestría en Ingeniería de Computación por la USP, Brasil. Es profesora en la Universidad Presbiteriana Mackenzie e investigadora en el Grupo de Arquitectura y Fábrica de Software. Tiene experiencia en arquitectura de software, desarrollo de software y en definición de ambientes de producción de software.

 ana.rossi@mackenzie.br

Fabio Silva Lopes

Es graduado en Tecnología de Procesamiento de Datos y tiene maestría en Salud Pública y doctorado en Ciencias por la USP, Brasil. Es profesor asociado en la Universidad Presbiteriana Mackenzie. Posee experiencia en arquitecturas de datos para inteligencia de negocios.

 flopes@mackenzie.br