

ANÁLISIS DE LOS ARTEFACTOS DEL PROCESO ÁGIL ICONIX PARA EVALUAR LA CALIDAD DEL PRODUCTO SOFTWARE, 2017

Efraín E. Porras Flores

Unidad de Investigación e Innovación de Ingeniería de Minas, Geología y Civil

Programa de Ingeniería de Sistemas-Área Ingeniería de Software

E-mail: efrain.porras@unsch.edu.pe

RESUMEN

El desarrollo de software con agilidad tiene una gran comunidad de usuarios en el mundo, porque el desarrollo a pequeña escala, no ha sido resuelto por la ingeniería de software, si se exige calidad del producto software, disminuir costos y tiempos de entrega. El objetivo del estudio es analizar las fases del proceso ágil Iconix, para tener una matriz que permita evaluar la calidad del producto software, mediante la NTP ISO/IEC 9126; se ha considerado el análisis de las fases de: análisis de requisitos, diseño preliminar, diseño e implementación. El estudio fue realizado en el Perú, porque existe necesidad de software para Pymes y Mipés que son aproximadamente 2 millones que requieren software. Se recolectó datos en un solo momento, mediante encuestas a diez expertos en proceso ágil, el estudio es retrospectivo, analítico porque tiene las variables: artefacto del proceso ágil Iconix, calidad del producto software. El análisis, síntesis ha permitido formular las matrices de relación entre los artefactos del proceso ágil Iconix y la calidad del producto software, según el modelo de calidad para características y sub características de acuerdo a los estándares; NTP ISO/IEC 9126-1, 2 y 3, NTP-ISO/IEC 14598-1 y 2, adaptado por INDECOPI. Según la hipótesis general, el proceso ágil Iconix está relacionado con la evolución de la calidad del producto software. De acuerdo a las hipótesis específicas las fases de: análisis de requisitos, diseño preliminar, diseño e implementación y la evaluación de calidad del producto software, demuestran que están relacionados.

Palabras clave: Artefacto del proceso ágil Iconix, Calidad de producto software, Proceso ágil Iconix, ISO/IEC 9126.

ANALYSIS OF THE ARTIFACTS OF THE ÁGIL ICONIX PROCESS TO EVALUATE THE QUALITY OF THE SOFTWARE PRODUCT, 2017

ABSTRACT

Software development with agility has a large community of users in the world, because the development on a small scale, has not been solved by software engineering, if software product quality is demanded, decrease costs and delivery times. The objective of the study is to analyze the phases of the agile Iconix process, to have a matrix that allows to evaluate the quality of the software product, through the NTP ISO / IEC 9126; the analysis has been considered the phases of: Requirements analysis, preliminary design, design and implementation. The study was conducted in Peru, because there is a need for software for Pymes and Mipés that are approximately 2 million that require software. Data was collected in a single moment, by means of surveys to ten experts in an agile process, the study is retrospective, analytical because it has the variables: artifact of the agile Iconix process, quality of the software product. The analysis, synthesis has allowed to formulate the matrices of relationship between the devices of the agile Iconix process and the quality of the software product, according to the model of quality for characteristics and sub characteristics according to the standards; NTP ISO / IEC 9126-1, 2 and 3, NTP-ISO / IEC 14598-1 and 2, adapted by INDECOPI. According to the general hypothesis, the agile process Iconix is related to the evolution of the quality of the software product. According to the specific hypotheses, the phases of: requirements analysis, preliminary design, design and implementation and the quality evaluation of the software product, show that they are related.

Keywords: Iconix agile process artifact, Software product quality, Iconix agile process, ISO / IEC 9126.

INTRODUCCIÓN

Los procesos de desarrollo de software ágil surgen como respuesta a uno de los problemas de la ingeniería de software que se ha discutido por muchos años, sobre cómo deben realizarse las tareas del desarrollo de software con el fin de agilizar las entregas, reducir costos y obtener mejores soluciones (Aguilar, 2015).

Según Rosenberg y Scott (1999), Rosenberg y Scott (2001), Rosenberg et al. (2005) y Rosenberg y Stephens (2007), el proceso ágil Iconix tiene las fases de: análisis de requisitos, diseño preliminar, diseño e implementación.

Los aspectos de calidad interna y externa de software son tratados de acuerdo a las normas técnicas peruanas; NTP

ISO/IEC 9126-1:2004, NTP ISO/IEC 9126-2:2004, NTP ISO/IEC 9126-3:2005, NTP ISO/IEC 14598-1:2005 y NTP ISO 14598-3:2005, para evaluar la calidad del producto software. Por tanto, los resultados sobre la calidad de producto software, está referido a las normas citadas.

El estudio tiene como finalidad brindar un método de alineamiento entre; el proceso ágil Iconix y la calidad del producto software, para los desarrolladores de software del Perú, donde existe aproximadamente 2 millones Pymes y Mipés que requieren software sobre demanda.

Los procesos de desarrollo de software ágil no han sido resueltos totalmente por la ingeniería de software, particularmente si está relacionado a la calidad del producto software. Esta realidad está vigente en el Perú y el mundo, se

ha planteado realizar el análisis de los artefactos del proceso ágil Iconix y su relación con la calidad del producto software.

Los objetivos específicos son: a) Analizar la fase de análisis de requisitos a fin de evaluar la calidad del producto software. b) Analizar la fase de diseño preliminar a fin de evaluar la calidad del producto software. c) Analizar la fase de diseño a fin de evaluar la calidad del producto software. d) Analizar la fase de implementación a fin de evaluar la calidad del producto software.

MATERIAL Y MÉTODOS

Diseño De Investigación

“El diseño señala al investigador lo que debe hacer para alcanzar sus objetivos de estudio, contestar las interrogantes que se ha planteado y analizar la certeza de la hipótesis formulada en un contexto en particular” (Hernández et al., 2010). El estudio es un diseño no experimental transversal y

retrospectivo. Donde se pretende demostrar que el análisis de los artefactos del proceso ágil Iconix y la evaluación de la calidad del producto software están relacionados en función al modelo de calidad de la norma NTP ISO/IEC 9126 y sus extensiones, recolectando información mediante encuesta a diez expertos del proceso ágil Iconix del Perú en el año 2017.

Técnicas Para Recolectar Información

Se ha utilizado la técnica de encuesta a expertos del proceso software ágil Iconix, que permita evaluar la calidad del producto software durante el desarrollo.

Instrumentos Para Recolectar Información

Se ha diseñado el instrumento cuestionario, para la encuesta a diez expertos del proceso software ágil Iconix, que permita evaluar la calidad del producto software durante el desarrollo, utilizando una escala actitudinal a los encuestados. A continuación presentamos el cuestionario.

Cuestionario 1. Encuesta para calificar si el análisis de los artefactos del proceso ágil Iconix, permite evaluar la calidad del producto software de acuerdo al modelo de calidad NTP ISO/IEC 9126.						
Señor encuestado, sírvase calificar según la escala siguiente: (5) Totalmente de acuerdo (4) Parcialmente de acuerdo (3) Ni de acuerdo ni en desacuerdo (2) Parcialmente en desacuerdo (1) Totalmente en desacuerdo						
N° It.	Pregunta	5	4	3	2	1
¿Cómo califica el análisis de requisitos que permite evaluar la calidad del producto software?. Para:						
1	Los requisitos					
2	Objetos del mundo real y modelo de dominio inicial					
3	Los casos de uso					
4	Prototipo de interfaz gráfica					
5	Casos de uso empaquetados					
6	Primer borrador de casos de uso					
¿Cómo califica el análisis de diseño preliminar que permite evaluar la calidad del producto software?. Para:						
7	Primer borrador para cada caso de uso reescrito					
8	Primer corte de objetos que completan escenarios para cada caso de uso					
9	Modelo de dominio actualizado					
¿Cómo califica el análisis de diseño que permite evaluar la calidad del producto software?. Para:						
10	Modelo de dominio actualizado para cada caso de uso dividido					
11	Diagrama de secuencia para cada caso de uso					
12	Diagrama de clases					
13	Diagrama de clases de un caso de uso actualizado					
14	Controladores para pruebas unitarias					
¿Cómo califica el análisis de implementación que permite evaluar la calidad del producto software?. Para:						
15	Base de datos física					
16	Código para clases entidad					
17	Código para las interfaces					
18	Pruebas unitarias para cada controlador					
19	Código fuente para cada controlador					
20	Pruebas unitarias para cada controlador					
21	Pruebas de aceptación para cada caso de uso					
Señor encuestado, sírvase calificar, de acuerdo a la norma NTP ISO/IEC 9126 y sus extensiones, la capacidad del producto software, según la escala siguiente: (5) Totalmente de acuerdo (4) Parcialmente de acuerdo (3) Ni de acuerdo ni en desacuerdo (2) Parcialmente en desacuerdo (1) Totalmente en desacuerdo						

N° It.	Pregunta	5	4	3	2	1
1	En relación a la funcionalidad, el producto software tiene la capacidad de: ¿Presentar la funcionalidad que el usuario ha especificado?					
2	En relación a la funcionalidad, el producto software tiene la capacidad de: ¿Presentar resultados acordados con un grado necesario de precisión?					
3	En relación a la fiabilidad, el producto software tiene la capacidad de: ¿Evitar fallos como resultado de defectos en el software?					
4	En relación a la fiabilidad, el producto software tiene la capacidad de: ¿Mantener un nivel especificado de funcionamiento en caso de defectos del software o de incumplimiento de su interfaz especificada?					
5	En relación a la usabilidad, el producto software tiene la capacidad de: ¿Permitir al usuario entender si el software es aplicable, y cómo puede ser utilizado para las tareas y las condiciones particulares de aplicación?					
6	En relación a la usabilidad, el producto software tiene la capacidad de: ¿Permitir al usuario aprender su aplicación?					
7	En relación a la usabilidad, el producto software tiene la capacidad de: ¿Permitir al usuario operarlo y controlarlo?					
8	En relación a la eficiencia, el producto software tiene la capacidad de: ¿Utilizar apropiadas cantidades y tipos de recursos cuando este funciona bajo las condiciones establecidas?					
9	En relación a la eficiencia, el producto software tiene la capacidad de: ¿Proveer tiempos apropiados de respuesta y procesamiento, y ratios de rendimiento cuando realiza su función bajo las condiciones establecidas?					
10	En relación a la facilidad de mantenimiento, el producto software tiene la capacidad de: ¿Ser diagnosticado por deficiencias o causas de fallos en el software o la identificación de las partes a ser modificadas?					
11	En relación a la facilidad de mantenimiento, el producto software tiene la capacidad de: ¿Permitir que una determinada modificación sea implementada?					
12	En relación a la portabilidad, el producto software tiene la capacidad de: ¿Ser adaptado a diferentes entornos definidos sin aplicar acciones o medios diferentes de los previstos para el propósito del software considerado?					
13	En relación a la portabilidad, el producto software tiene la capacidad de: ¿Ser instalado en un entorno definido?					

RESULTADOS Y DISCUSIÓN

Artefactos del proceso ágil Iconix

Luego de revisar el marco teórico, se ha identificado las tareas, artefactos, técnicas y responsables del proceso ágil Iconix según sus fases.

Tarea	Artefacto	Técnica	Responsables
Identificar requisitos	a. Requisitos funcionales y no funcionales	a. Entrevistas	Usuario
	b. Casos de prueba	b. Definir lo que el sistema debe hacer c. Escribir al menos un caso de prueba para cada requisito	Cliente Analista

Identificar objetos del mundo real y dibujar modelo de dominio	Modelo de dominio	<ul style="list-style-type: none"> a. Identificar clases clave del negocio b. Identificar sustantivos y depurar c. Identificar objetos en requisitos funcionales y asignar al modelo de dominio d. Utilizar agregación y generalización 	Analista
Realizar prototipo de interfaz gráfica	Prototipo GUI	<ul style="list-style-type: none"> a. Utilizar historia de eventos del usuario b. Utilizar los requisitos funcionales c. Diseñar interfaz gráfica básica 	Programador Analista
Descubrir casos de uso	Lista de casos de uso	<ul style="list-style-type: none"> a. Utilizar requisitos funcionales b. Entrevistas 	Usuario Cliente Analista
Dibujar y empaquetar casos de uso	<ul style="list-style-type: none"> a. Diagrama de casos de uso b. Paquete de casos de uso 	<ul style="list-style-type: none"> a. Identificar roles y responsabilidades de actores b. Asociar actores con casos de uso c. Relacionar casos de uso d. Agrupar lógicamente casos de uso 	
Asignar requisitos funcionales a los casos de uso	Relación entre requisitos funcionales y casos de uso	<ul style="list-style-type: none"> a. Asignar requisitos funcionales a los casos de uso 	Analista
Escribir el primer borrador de casos de uso	Primer borrador de casos de uso	<ul style="list-style-type: none"> a. Utilizar glosario de objetos del modelo de dominio b. Utilizar la regla de dos párrafos c. Escribir el caso de uso como flujos de evento/respuesta d. Escribir el caso de uso con estructura sustantivo-verbo-sustantivo e. Escribir caso de uso en voz activa f. Referenciar por su nombre las pantallas 	

Tabla 1. Análisis de requisitos.

Tarea	Artefacto	Técnica	Responsables
Rescribir el primer borrador para cada caso de uso	Caso de uso desambiguado	<ul style="list-style-type: none"> a. Rescribir el caso de uso durante el análisis de robustez 	
Identificar el primer corte de objetos que completan escenarios para cada caso de uso	Diagrama de robustez	<ul style="list-style-type: none"> a. Copiar la descripción del caso de uso en el diagrama de robustez. b. Usar las clases del modelo de dominio c. Crear un objeto interfaz por cada GUI y nombrarlo d. Transformar verbos del caso de uso en controladores e. Relacionar un caso de uso al diagrama de robustez cuando es invocado f. Utilizar las reglas para construir el diagrama de robustez 	Analista
Actualizar el modelo de dominio	Modelo de dominio actualizado	<ul style="list-style-type: none"> a. Actualizar el modelo de dominio con nuevas clases y atributos durante el análisis de robustez 	
Actualiza el diagrama de clases de análisis	Modelo de dominio actualizado	<ul style="list-style-type: none"> a. Actualizar el diagrama de clases de análisis al finalizar el análisis de robustez b. Asignar atributos a las clases entidad 	

Tabla 2. Diseño preliminar.

Tarea	Artefacto	Técnica	Responsables
Dividir modelo de dominio actualizado para cada caso de uso	Parte de modelo de dominio actualizado	a. Coincidir las clases entidad del diagrama de robustez con parte del modelo de dominio actualizado y dibujarlo	Diseñador
Dibujar un diagrama de secuencia para cada caso de uso	Diagrama de secuencia	a. Copiar la descripción del caso de uso b. Copiar objetos entidad, interfaz y actores del diagrama de robustez c. Verificar que un mensaje del diagrama de secuencia es verbo en el caso de uso d. Hacer refactoring al diagrama de secuencia antes de codificar	Programador Diseñador
Actualizar el diagrama de clases de un caso de uso	Diagrama de clase	a. Asignar operaciones a las clases a partir de mensajes del diagrama de secuencia b. Establecer multiplicidad en las clases c. Depurar las clases, operaciones y atributos del diagrama de clases	
Extraer controladores para pruebas unitarias	Lista de controladores	a. Identificar controladores para la lógica del negocio desde un diagrama de robustez	

Tabla 3. Diseño.

Tarea	Artefacto	Técnica	Responsables
Implementar la base de datos física	Base de datos física	a. Escribir el script usando el modelo de dominio actualizado b. Ejecutar el script usando un DBMS	Programador
Implementar código para clases entidad	Código fuente	a. Escribir o generar código fuente con una herramienta usando el modelo de dominio actualizado	Programador
Implementar código para las interfaces	Código fuente	a. Generar código fuente usando una herramienta	Programador
Crear pruebas unitarias para cada controlador	Prueba unitaria	a. Escribir código fuente para una prueba unitaria usando una herramienta	Programador
Implementar código fuente para cada controlador	Código fuente	a. Escribir el código fuente siguiendo el flujo normal del diagrama de secuencia usando una herramienta b. Actualizar el diagrama de secuencia con la codificación	Programador
Ejecutar pruebas unitarias para cada controlador	Reporte de pruebas unitarias	a. Ejecutar el módulo de cada prueba unitaria b. Modificar código fuente si la prueba unitaria muestra resultado incorrecto	Programador
Ejecutar pruebas de aceptación para cada caso de uso	Reporte de pruebas de aceptación	a. Utilizar los casos de prueba de aceptación b. Ejecutar el módulo de un caso de uso c. Modificar código fuente si la prueba de aceptación muestra resultado incorrecto	Programador Usuario Cliente

Tabla 4. Implementación**Evaluación de la calidad del producto software**

La evaluación de los requerimientos de calidad del producto software, está de acuerdo al modelo de calidad adaptado por INDECOPI, Norma Técnica Peruana NTP ISO/IEC 9126. Se

ha determinado los pesos de las características según las necesidades del usuario, que es el experto de desarrollo de software, se muestra en la tabla 5.

Característica	Peso Característica
Funcionalidad	25%
Fiabilidad	15%
Usabilidad	25%
Eficiencia	15%
Facilidad de Mantenimiento	10%
Portabilidad	10%

Tabla 5. Pesos de características de las necesidades de usuarios.

La tabla 6, muestra las características en relación al peso de cada sub característica, para evaluar la calidad del producto

software, que logra las necesidades del usuario experto en desarrollo de software con Iconix.

Calidad Externa e Interna		
Característica	Sub Característica	Peso Sub Característica
Funcionalidad	Aplicabilidad	50.00%
	Precisión	50.00%
Fiabilidad	Madurez	55.00%
	Tolerancia a fallas	45.00%
Usabilidad	Entendibilidad	25.00%
	Facilidad de aprendizaje	40.00%
	Operabilidad	35.00%
Eficiencia	Comportamiento en el tiempo	45.00%
	Utilización de recursos	55.00%
Facilidad de Mantenimiento	Cambiabilidad	100.00%
Portabilidad	Adaptabilidad	55.00%
	Instalabilidad	45.00%

Tabla 6. Pesos de sub características de las necesidades de usuarios.

Característica	Sub Característica	Métricas	Nivel Requerido
Funcionalidad	Aplicabilidad	Métrica interna, Métrica externa	72.00%
	Precisión	Métrica interna, Métrica externa	68.00%
Fiabilidad	Madurez	Métrica interna, Métrica externa	70.00%
	Tolerancia a fallas	Métrica interna, Métrica externa	55.00%
Usabilidad	Entendibilidad	Métrica interna, Métrica externa	73.00%
	Facilidad de aprendizaje	Métrica interna, Métrica externa	75.00%
	Operabilidad	Métrica interna, Métrica externa	68.00%
Eficiencia	Utilización de recursos	Métrica interna	65.00%
	Comportamiento en el tiempo	Métrica interna	55.00%
Facilidad de Mantenimiento	Cambiabilidad	Métrica interna	60.00%
Portabilidad	Adaptabilidad	Métrica interna	82.00%
	Instalabilidad	Métrica interna	65.00%

Tabla 7. Categorías de medición de calidad interna y externa.

La tabla 8, presenta el plan de evaluación para la calidad del producto software, plan desarrollado en relación a las tablas 1 a 4, 6 y 7. La tabla 8 permite el análisis de los artefactos del

proceso Iconix para evaluar la calidad del producto software durante el desarrollo.

Sub Característica	Artefactos a ser Evaluados	Métricas Internas Usadas	Métricas Externas Usadas
Aplicabilidad	1. Catálogo de requisitos. 2. Descripción de casos de uso. 3. Interfaz gráfica de usuario. 4. Código fuente.	1. Adecuación funcional. 2. Integridad de implementación funcional. 3. Cobertura de la implementación funcional.	1. Adecuación funcional 2. Integridad de implementación funcional 3. Cobertura de implementación funcional
Precisión	1. Catálogo de requisitos. 2. Código fuente. 3. Pruebas unitarias. 4. Reporte de pruebas de aceptación.	1. Exactitud de cálculos.	1. Exactitud de cálculos.
Madurez	1. Reporte de pruebas unitarias. 2. Reporte de pruebas de integración. 3. Casos de pruebas de aceptación. 4. Reporte de pruebas de aceptación.	1. Eliminación de fallas.	1. Densidad de fallas contra los casos de prueba. 2. Madurez de la prueba
Tolerancia a fallas	1. Catálogo de requisitos. 2. Reporte de pruebas unitarias. 3. Casos de prueba de aceptación. 4. Reporte de pruebas de aceptación.	1. Prevención de operación incorrecta.	1. Prevención de operación incorrecta.
Entendibilidad	1. Catálogo de requisitos. 2. Descripción de casos de uso. 3. Interfaz gráfica de usuario.	1. Función de comprensión	1. Comprensión de entradas y salidas
Facilidad de aprendizaje	1. Catálogo de requisitos. 2. Lista de casos de uso. 3. Descripción de casos de uso.	1. Integridad de la documentación del usuario y/o facilidad de ayuda	1. Facilidad de aprender a realizar una tarea en uso
Operabilidad	1. Descripción de casos de uso. 2. Código fuente. 3. Interfaz gráfica de usuario.	1. Claridad de mensajes. 2. Claridad de la interfaz.	1. Entendibilidad del mensaje en uso
Utilización de recursos	1. Código fuente. 2. Estándares de codificación.	1. Utilización de memoria.	
Comportamiento en el tiempo	1. Casos de prueba. 2. Reporte de pruebas de aceptación. 3. Reporte de pruebas de integración.		1. Rendimiento
Cambiabilidad	1. Catálogo de requisitos. 2. Código fuente.	1. Registro de cambios	
Adaptabilidad	1. Catálogo de requisitos. 2. Arquitectura técnica (Diagrama de despliegue, diagrama de componentes).	1. Adaptabilidad al entorno organizacional (adaptabilidad a la organización y a la infraestructura de la misma)	
Instalabilidad	1. Procedimiento de instalación.		1. Facilidad de reinstalación

Tabla 8. Plan de evaluación para la calidad del producto software.

Para la hipótesis específica a) de investigación

Hipótesis nula (H_0). Si realizamos el análisis de la fase de análisis de requisitos entonces es posible evaluar la calidad del producto software.

Hipótesis alterna (H_a). Si realizamos el análisis de la fase de análisis de requisitos entonces no es posible evaluar la calidad del producto software.

Se encontró que el valor calculado con la prueba T Student es de 1.7625, con un T crítico de 1.7959, error de 5%, 11 Gl; que significa que existe relación entre el análisis de requisitos y la evaluación de la calidad del producto software desarrollado mediante las tablas del 1, 7 y 8.

Para la hipótesis específica b) de investigación

Hipótesis nula (H_0). Si realizamos el análisis de la fase de diseño preliminar entonces es posible evaluar la calidad del producto software.

Hipótesis alterna (H_a). Si realizamos el análisis de la fase de diseño preliminar entonces no es posible evaluar la calidad del producto software.

Se encontró que el valor calculado con la prueba T Student es de 2.8721, con un T crítico de 2.923, error de 5%, 1 Gl; que significa que existe relación entre el diseño preliminar y la evaluación de la calidad del producto software desarrollado mediante las tablas del 2, 7 y 8.

Para la hipótesis específica c) de investigación

Hipótesis nula (H_0). Si realizamos el análisis de la fase de diseño entonces es posible evaluar la calidad del producto software.

Hipótesis alterna (H_a). Si realizamos el análisis de la fase de diseño entonces no es posible evaluar la calidad del producto software.

Se encontró que el valor calculado con la prueba T Student es de 2.1275, con un T crítico de 2.1318, error de 5%, 4 Gl; que significa que existe relación entre el diseño y la evaluación de la calidad del producto software desarrollado mediante las tablas del 3, 7 y 8.

Para la hipótesis específica d) de investigación

Hipótesis nula (H_0). Si realizamos el análisis de la fase de implementación entonces es posible evaluar la calidad del producto software.

Hipótesis alterna (H_a). Si realizamos el análisis de la fase de implementación entonces no es posible evaluar la calidad del producto software.

Se encontró que el valor calculado con la prueba T Student es de 1.7628, con un T crítico de 1.7709, error de 5%, 13 Gl; que significa que existe relación entre la implementación y la evaluación de la calidad del producto software desarrollado mediante las tablas del 4, 7 y 8.

Para la hipótesis general de investigación

Hipótesis nula (H_0). Si realizamos el análisis de los artefactos del proceso ágil Iconix entonces será posible evaluar la calidad del producto software.

Se encontró que el valor calculado con la prueba T Student

de 1.685, con un T crítico de 1.6924, error de 5%, 32 Gl; que significa que existe relación entre el análisis de los artefactos del proceso ágil Iconix y la evaluación de la calidad del producto software desarrollado mediante las tablas del 7 y 8. Demostrando que la afirmación; “Si realizamos el análisis de los artefactos del proceso ágil Iconix entonces será posible evaluar la calidad del producto software”, es verdadera.

DISCUSIÓN

El modelo de calidad del software proporciona una definición formal de conceptos acordados que aseguren la interpretación correcta del conocimiento (Martin, 2004). El estudio usa la NTP-ISO/IEC 9126-1:2004 modelo de calidad, NTP-ISO/IEC-TR 9126-2:2004 métricas externas, NTP-ISO/IEC-TR 9126-3:2005 métricas internas, NTP-ISO/IEC 14598-1:2005 evaluación del producto software; asegurando el análisis de los artefactos del proceso ágil Iconix y la evaluación de calidad del producto software, desarrollado mediante Iconix, demostrado según el contraste de la hipótesis general.

El proceso Iconix presenta un modelo estático que es refinado durante las iteraciones del modelo dinámico (Pantoja, 2009). Realizamos la presentación de los artefactos que produce el proceso ágil Iconix, descrito en las tablas 1 a 2, se observa que en las fases de análisis de requisitos y diseño preliminar existen iteraciones que permiten evaluar la calidad del producto software, demostrado según el contraste de las hipótesis específicas a y b.

Iconix es un proceso ágil puro, práctico y simple, que realiza análisis y diseño, con la característica “rastreadibilidad de los requisitos”, compuesto por los artefactos: modelo de dominio, modelo de caso de uso, diagrama de robustez, diagrama de secuencia y diagrama de clase (Pantoja, 2009). Los resultados mostrados según la prueba de las hipótesis específicas a, b y c, para el análisis de los artefactos del proceso ágil Iconix que permite evaluar la calidad del producto software, demuestran que existe relación.

La gestión de calidad del software, es probar el software para encontrar errores, y evaluar la calidad del software usando métricas (Scalone, 2003). Se presenta las métricas en la tabla 8, que se realiza mediante las pruebas luego de la implementación y tiene relación con los artefactos del proceso ágil Iconix que permite evaluar la calidad de producto software, según el modelo de calidad. Se demuestra que existe relación de acuerdo a la hipótesis específica d.

Conclusiones

- El análisis de los artefactos del proceso software ágil Iconix permite evaluar la calidad del producto software durante el desarrollo, según el contraste de la hipótesis general.
- Los artefactos del proceso ágil Iconix mostrados en las tablas 2, 7 y 8 del diseño preliminar permiten evaluar la calidad del producto software durante el desarrollo.
- Los artefactos del proceso ágil Iconix presentados en las tablas 3, 7 y 8 de la fase de diseño permiten evaluar la calidad del producto software durante el desarrollo.
- Los artefactos del proceso ágil Iconix de la fase de implementación permiten evaluar la calidad del producto software presentados en la tabla 4, según las NTP-ISO/IEC 9126-1:2004 modelo de calidad, NTP-ISO/IEC-TR 9126-2:2004 métricas externas, NTP-

ISO/IEC-TR 9126-3:2005 métricas internas, NTP-ISO/IEC 14598-1:2005.

Recomendaciones

- Se debe realizar un estudio de nivel aplicativo, para demostrar la relación practica entre el proceso ágil Iconix y la calidad del producto software.
- Se debe investigar, si los procesos agiles y formales, implícitamente generan calidad del producto software durante el desarrollo aplicando técnicas de ingeniería de software para procesos agiles.
- Se debe adaptar los procesos agiles para desarrollo de software, mediante las técnicas formales de la ingeniería de software, con la finalidad de mejorar el grado de calidad del producto software.

AGRADECIMIENTOS

Mi agradecimiento a los expertos en desarrollo de software ágil, específicamente Iconix, quienes son ex alumnos actualmente Ingenieros de Sistemas egresados de la UNSCH, por su aporte en el análisis y opinión durante las encuestas.

REFERENCIAS BIBLIOGRÁFICAS

Aguilar, E. (2015) Integración del diseño centrado en usuario con metodologías ágiles en el desarrollo de un catálogo de plantas. Un estudio de investigación - acción. Perú, Facultad de Ciencias e Ingeniería. Pontificia Universidad Católica del Perú.

Comisión de Reglamentos Técnicos y Comerciales-INDECOPI. (2004) Norma Técnica Peruana NTP-ISO/IEC 9126-1, Ingeniera de software. Calidad el Producto. Parte 1; Modelo de Calidad. Primera Edición. Perú, Lima.

Díaz, J. (2001) Lineamientos para la Determinación del Perfil de Calidad de un Producto de Software. Tesis de Maestría. México, Programas de Posgrado en Electrónica, Computación, Información y Comunicaciones. Instituto Tecnológico y de Estudios Superiores de Monterrey.

Fowler, M. y J. Highsmith. (2001) the agil manifestó. Software Development.

Jacobson, I.; Booch, G. y J. Rumbaugh, (2000) El Proceso Unificado de Desarrollo de Software. Primera Edición. Madrid, Pearson Educación S.A.

Letelier, P. y M. Penadés, (2006) Metodologías Ágiles para el Desarrollo de Software: eXtreme Programming (XP), Investigación. España, Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia.

Martin, M. (2004) *Sistema de Catalogación de Métricas e Indicadores con Potencia de Web Semántica*. Tesis de Magíster. Argentina, Facultad de Informática. Universidad Nacional de La Plata.

NTP-ISO/IEC 9126-1 (2004) Ingeniería de Software. Calidad del Producto. Parte 1: Modelo de Calidad. Primera Edición. Lima. INDECOPI.

NTP-ISO/IEC 9126-2 (2004) Ingeniería de Software. Calidad del Producto. Parte 2: Métricas Externas. Primera Edición. Lima. INDECOPI.

NTP-ISO/IEC 9126-3 (2005) Ingeniería de Software. Calidad del Producto. Parte 3: Métricas Internas. Primera Edición. Lima. INDECOPI.

NTP-ISO/IEC 14598-1 (2005) Tecnología de la Información. Evaluación del Producto. Parte 1: Visión General. Primera Edición. Lima. INDECOPI.

NTP-ISO 14598-3 (2005) Ingeniería de Software. Evaluación del Producto. Parte 3: Proceso para Desarrolladores. Primera Edición. Lima. INDECOPI.

Morán, E. (2008) Enfoque de la calidad del software enfoque ISO 12207 e ISO 9126”, M&T Consulting.

Porras, E. (2010) Comparación de dos procesos de desarrollo de software usando los métodos Iconix y XP, caso: comercialización de la tara en la región Ayacucho. Tesis de Magíster. Perú, Maestría en Ciencias con Mención en Ingeniería de Sistemas. Universidad Nacional de Ingeniería.

Rafael, M. (2004) Processos de Desenvolvimento de Software: RUP e ICONIX. Tesis Bachiller. Brasil, Universidade Estadual de Londrina.

Salvador, C. (2013) Una Revisión Sistemática de Usabilidad en Metodologías Agiles. Tesis de Magíster. Perú, Maestría en Informática Mención Ingeniería de Software. Pontificia Universidad Católica del Perú.

Samamé, J. (2013) Aplicación de una metodología ágil en el desarrollo de un sistema de información. Tesis de Magíster. Perú, Maestría en Informática Mención Ingeniería de Software. Pontificia Universidad Católica del Perú.

Rosenberg, D.; Stephens, M. y M. Collins-Cope, (2005) Agile Development with ICONIX Process: People, Process, and Pragmatism. Primera Edición. EUA, Apress.

Rosenberg, D. y K. Scott (1999) Use Case Driven Object Modeling with UML: A Practical Approach. Primera Edición. EUA. Addison-Wesley.

Rosenberg, D. y K. Scott (2001) Applying Use Case Driven Object Modeling with UML: Un Anotated e- Commerce Sample. Primera Edición. United States. Addison-Wesley.

Rosenberg, D. y M. Stephens (2007) Use Case Driven Object Modeling with UML: Theory and Practice. Primera Edición, United States. Apress.

Scalone, F. (2006) Estudio Comparativo de Los Modelos y Estándares de Calidad del Software. Tesis de Magíster. Argentina, Maestría en Ingeniería en Calidad. Universidad Tecnológica Nacional.

Pantoja, J. (2005) “Construindo Softwares com Qualidade e Rapidez Usando ICONIX”. [En línea]. Brasil, disponible en: <http://www.guj.com.br/article.show.logic?id=172>. [Acceso 15 setiembre 2017].

Patricia, C. (s.f). La Metodología ICONIX. México.

Porras, E. (2011). La Metodología Ágil y Formal ICONIX para el Desarrollo de Software: Teoría y Práctica. Ayacucho, Perú: Ami Ayacucho.

Rafael, M. (2004) Processos de Desenvolvimento de Software: RUP e ICONIX. Tesis Bachiller. Brasil. Universidade Estadual de Londrina.

Sommerville, I. y Alfonso, M. (2003). Ingeniería del software. (7° Ed.) Madrid, España: Pearson Educación.

Tamayo, M. y Tamayo, A. (1997). El Proceso de la Investigación Científica. (3a ed.). México: LIMUSA.

Teniente, E., Ramon, A, Mayol, E. y Gómez, C. (2003) Diseño de sistemas software en UML. Barcelona: Ediciones UPC.

The Agile Methodology. [En línea]. EUA. Disponible en: <<http://agilemethodology.org>>. [Acceso 25 marzo 2017].