

Tipo de artículo: Artículo original
Temática: Informática jurídica
Recibido: 30/10/2015 | Aceptado: 13/01/2016

Guía para el aseguramiento de la confiabilidad en aplicaciones web

Guide to ensuring reliability in web application

Yoslenys Roque Hernández^{1*}, Bárbara Almarales Lara¹, Yordi Chaveco Bustamante¹

¹ Centro de Gobierno Electrónico. Facultad 3. Universidad de las Ciencias Informáticas, Carretera a San Antonio de los Baños, km 2 ½, La Lisa, Torrens, La Habana. Cuba. {yrhdez, balmarales, ybustamante}@uci.cu

* Autor para correspondencia: yrhdez@uci.cu

Resumen

El propósito del presente trabajo es presentar una guía para el aseguramiento de la confiabilidad en el desarrollo de aplicaciones web en el Centro de Gobierno Electrónico perteneciente a la Universidad de las Ciencias Informáticas. Su realización estuvo motivada por los riesgos en los que pueden incidir los productos en el centro al no efectuarse un proceso de aseguramiento de la característica de calidad: confiabilidad. Para la confección de esta guía se analizaron de manera exhaustiva algunos de los conceptos, modelos y estándares de calidad, su aseguramiento y los aspectos fundamentales relacionados a la confiabilidad. Esto permitió la utilización de un conjunto de aspectos que se utilizaron para la elaboración de la guía, la cual define un conjunto de actividades y elementos a tener en cuenta en cada una de las etapas del ciclo de vida del software. Esta estrategia se ha validado positivamente con especialistas en el tema que han dado un criterio acertado sobre la misma, lo que posibilita que pueda servir de base para que se haga extensiva su aplicación y/o adaptación a todos los proyectos de la Universidad y otras empresas.

Palabras clave: Aseguramiento, Calidad, Confiabilidad, Guía.

Abstract

The purpose of this study is to present a guide for ensuring reliability in the development of web applications at the Center for Electronic Government belonging to the University of Informatics Sciences. It's realization was motivated by the risks that may affect the products in the center by failing to make assurance process quality characteristic: reliability. For the preparation of this guide were analyzed some of the concepts, models and quality standards, your assurance and fundamental aspects related to reliability. This allowed the use of a number of aspects that were used for the preparation of the guide, which defines a set of activities and elements to be considered in each of the stages of the software life cycle. This strategy has been positively validated with experts in the field that have sound

judgment on the same, making it possible to serve as a basis to be extended the application and / or adaptation to all University projects and others companies.

Keywords: Assurance, Guide, Quality, Reliability.

Introducción

Durante todo el decurso de la humanidad se han identificado disímiles necesidades a las que el hombre se ha propuesto darle respuestas con el mejor grado de solidez posible, para así evitar volver a incurrir en lo que ya supuso estuvo solucionado, esto es a lo que se llama un trabajo “hecho con calidad”. En la actualidad, la informática se ha convertido en la ciencia clave para el progreso en casi todos los sectores económicos de un país y la producción de software ha ocupado un lugar significativo en el ámbito económico mundial.

En todo este vertiginoso desarrollo hacia la informatización, el uso de las tecnologías web y las aplicaciones de este tipo traen consigo soluciones novedosas, que hacen que la calidad tenga una presencia vital debido al papel fundamental que implica la competitividad de las empresas, pues los productos y servicios brindados deben adaptarse a las necesidades, gustos y exigencias de los potenciales clientes para mantenerse en el mercado. El propósito de las actividades de garantía de la calidad es tener un proceso de desarrollo sin errores, por lo que cada aplicación hecha debe llevar intrínseca propiedades o cualidades como la seguridad, la disponibilidad, el rendimiento óptimo, la total integridad de los datos, la facilidad de mantenimiento y la confiabilidad, definiéndose este último como la capacidad que tiene un producto de realizar su función de la manera prevista sin incidentes por el período de tiempo especificado en el diseño del mismo (QUIÑONES, 2006). La confiabilidad no se mide con el mismo rigor en todos los productos de software, aquellos que van a ser explotados por un largo período de tiempo necesitan ser más confiables, seguros y de fácil mantenimiento para poder disminuir los costos a la hora de entrar en etapa de explotación.

En la Universidad de las Ciencias Informáticas (UCI), el Centro de Gobierno Electrónico (CEGEL) de la Facultad 3, está conformado por un grupo de proyectos que, en sentido general, tienen como misión satisfacer las necesidades de instituciones gubernamentales mediante el desarrollo de productos, servicios y soluciones integrales de alta confiabilidad, calidad, competitividad, fidelidad y eficiencia. En una entrevista aplicada a diferentes miembros de los equipos de desarrollo de dichos proyectos, dígase arquitectos, líderes de proyecto y especialistas de calidad, se pudo evidenciar que no se aplican o no está correctamente determinado el empleo de estándares de calidad, listas de chequeo o pruebas de software enfocadas específicamente a la confiabilidad. A pesar de haberse realizado en el centro

varios estudios referentes al tema, los mismos no se encuentran bien documentados y no se siguen durante todo el ciclo de desarrollo de las aplicaciones web, por lo que no hay actividades que respondan íntegramente al aseguramiento de la confiabilidad. En el Centro Nacional de Calidad de Software (CALISOFT), específicamente en el Departamento de pruebas de software (DPSW) se evalúa la confiabilidad del software mediante una lista de chequeo, pero es aplicada una vez terminado el software, por lo que en las etapas tempranas del ciclo de vida del producto no se responde por el cuidado de esta característica (CALISOFT, 2012). Esto constituye un riesgo que a lo largo del tiempo pudiera ocasionar pérdida de información, revelación de datos confidenciales y corrupción de programas o datos. Además, provoca que los sistemas sean no fiables, inseguros o rechazados a menudo por sus usuarios, fallos en el hardware o software que implican altos costos por daños ocasionados y por concepto de reparación. También existen otros problemas como la denegación de servicio al entrar en un estado en que sus servicios normales no se encuentran disponibles y otros inconvenientes que pueden producir, de acuerdo a las características e importancia del sistema, pérdidas económicas, daños físicos e insatisfacción del cliente.

A raíz de la problemática antes descrita, esta investigación se plantea como objetivo general proponer una guía para el aseguramiento de la confiabilidad en el proceso de desarrollo de aplicaciones web en CEGEL, que sirva de base para otros proyectos productivos.

Materiales y métodos

Calidad de Software

Según Pressman la calidad de software es la: “Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente (PRESSMAN, 2006).” Esta definición brinda una visión clara, profunda y la define de la manera más específica garantizando la calidad del producto y de la documentación.

Aseguramiento de la calidad de software orientado a la confiabilidad del producto

El proceso de aseguramiento de la calidad del software es el destinado a cumplir con las especificaciones tanto del cliente como de calidad. La presente investigación se centra en el aseguramiento de calidad, pero orientado a la confiabilidad y para ello se define como aseguramiento de la confiabilidad como *un diseño planificado de acciones o actividades a realizar a lo largo del ciclo de vida de software que se requieren para garantizar la confiabilidad en el producto final* (MCGRAW-HILL, 2007).

El proceso de aseguramiento es de suma importancia dentro del desarrollo del software, ya que brinda un margen para la obtención de un producto que cumpla con lo que realmente el cliente quiere y, es muy conveniente, que los ingenieros encuentren y corrijan inmediatamente los defectos introducidos para asegurar la confiabilidad del software.

Confiabilidad

La confiabilidad es uno de los principales elementos que se deben tener en cuenta cuando se construye un software pues, permite predecir las posibilidades de fallo del producto y, por tanto, conocer con antelación las deficiencias y errores permisibles que este pueda presentar (ISO STANDARDS DEFINITION, 2008).

Las aplicaciones web no están exentas a la necesidad de implementar esta característica de calidad durante todo su proceso de desarrollo para ponerla en práctica durante su explotación. La necesidad de crear productos confiables, seguros, con alto grado de disponibilidad y capaces de recuperarse ante fallos constituye hoy una meta para cualquier industria de software. Es por ello que con el objetivo de definir qué aspectos influyen en la confiabilidad de un producto, se lleva a cabo un estudio sobre un conjunto de modelos y estándares de calidad mediante el cual se analiza el concepto de Confiabilidad brindado por cada uno; así como los diferentes elementos o sub-categorías asociadas a esta. En la Tabla 1. Definiciones de confiabilidad y sub-características asociadas a este atributo de calidad según los modelos y estándares estudiados, se muestran los resultados obtenidos:

Tabla 1. Definiciones de confiabilidad y sub-características asociadas a este atributo de calidad según los modelos y estándares estudiados.

Estándar o modelo	Definición de confiabilidad	Sub-categorías asociadas
Modelo de McCall	“Grado en que un programa se puede esperar para llevar a cabo su función con la precisión requerida”. (MCCALL, RICHARDS y WALTERS, 1977)	Disponibilidad Precisión Robustez Tolerancia
Modelo de Boehm	“Se puede esperar el software para que lleve a cabo las funciones previstas de forma satisfactoria”. (BOEHM, BROWN y LIPOW, 1976)	Precisión Compleitud Consistencia Auto-contención Robustez/Integridad
NC-ISO/IEC 9126-1: 2005	“Capacidad del producto de software para mantener un nivel de ejecución especificado cuando se usa bajo las condiciones especificadas”. (JTC1/SC7, ISO/IEC y CD 25010, 2007)	Madurez Tolerancia al Defecto Recuperabilidad Fiabilidad Conformidad
Modelo QUINT2	“Conjunto de atributos que refieren la capacidad del software para mantener su nivel de funcionamiento bajo condiciones indicadas por un período de tiempo indicado”.	Madurez Tolerancia a fallos Recuperabilidad Disponibilidad

	(SALANOVA, 2006)	Degradabilidad
Modelo de Calidad Web (WQM)	“Capacidad del portal para llevar a cabo su funcionalidad de forma precisa”. (CALERO, 2008)	Frecuencia y severidad de las fallas Capacidad de recuperación ante fallas
Portal Quality Model (PQM)	“Capacidad del portal para llevar a cabo su funcionalidad de forma precisa”. (CALERO, CACHERO, CÓRDOBA y MORAGA, 2008)	Tolerancia a fallos Utilización de recursos Disponibilidad Calidad en las búsquedas

Tras culminar el estudio, los autores de la presente investigación definen a la confiabilidad como el “conjunto de atributos que refieren la capacidad del software para mantener su nivel de funcionamiento bajo condiciones indicadas por un período de tiempo indicado y con la precisión requerida”. En cuanto a las sub-categorías asociadas a esta se precisan las siguientes: Disponibilidad, Recuperabilidad, Precisión, Tolerancia a fallos, Mantenibilidad, Madurez, Consistencia, Completitud y Robustez/Integridad.

Ingeniería de la Confiabilidad

La Ingeniería de la Confiabilidad surge en 1970, es una práctica que permite planear y guiar el proceso de prueba del software de manera cuantitativa y está dedicada al estudio de los aspectos físicos y aleatorios del fenómeno falla. Se caracteriza por tener dos elementos fundamentales:

El uso esperado relativo de las funcionalidades del sistema: su objetivo es centrarse en caracterizar de manera cuantitativa el uso esperado del sistema mediante la definición del llamado perfil de operaciones que no es más que la caracterización cuantitativa del uso que se espera de las funcionalidades principales del sistema. Dicha caracterización permite optimizar el uso de los recursos en las funciones que tengan mayor impacto y mayor uso esperado dentro del sistema (LLANES, 2007).

Los requerimientos de calidad definidos por el cliente, que incluyen la confiabilidad, la fecha de liberación y costo del ciclo de vida del proyecto: se refiere al enfoque al cliente mediante el establecimiento de objetivos cuantitativos asociados a la calidad del producto (representados con base a las fallas del producto). La satisfacción de dichos objetivos permite establecer un balance entre los costos del producto, así como la satisfacción de las necesidades del cliente (KATY ESPRONCEDA JOR, 2010).

Tomando en cuenta los conceptos planteados se puede llegar a la conclusión que, a través del estudio de la confiabilidad de un producto de software, es posible reconocer si el producto conserva buen desempeño ante las diversas circunstancias que se pueden presentar, verificando siempre que el producto reaccione positivamente ante las posibles fallas que pueda presentarse por defectos en el programa o causas externas, protegiendo siempre la integridad de la información. Por su parte los modelos y estándares de calidad que referencian a los productos permiten mayor organización y clasificación de las características de calidad.

Aplicación web

Una aplicación web se puede definir como el “software cliente/servidor que interactúa con usuarios y sistemas utilizando HTTP (Hypertext Transfer Protocol, Protocolo de Transferencia de Hipertexto)” (BENCHIMOL, 2011). Esta interacción permite realizar consultas a bases de datos, así como brindar servicios de manera efectiva.

Ciclo de vida del proceso de desarrollo de software basado en el Programa de Mejora

Actualmente la universidad está acometiendo un Programa de Mejora particular, basado en CMMI, que contempla la Administración de la Configuración, la Medición y Análisis, el Monitoreo y Control de Proyecto, la Planeación del Proyecto, el Aseguramiento de la Calidad del proceso y el producto, la Administración de Requisitos y la Administración de Acuerdo con Proveedores como las áreas de procesos del Nivel 2 de CMMI (AXENTIA, 2006).

El Programa de Mejora en cuestión define un ciclo de vida para el desarrollo de los productos. Debido a que la presente investigación está encaminada a proponer una guía para el aseguramiento de la confiabilidad en el proceso de desarrollo de aplicaciones web de CEGEL y que esta sirva de base para otros proyectos productivos de la universidad, se hace necesario realizar un estudio de dicho ciclo de vida. En la Tabla 2 se muestran cada una de las etapas que lo conforman.

Tabla 2. Ciclo de vida del software propuesto por el Programa de Mejora en la UCI.

Estudio Preliminar
Modelación del Negocio
Requisitos
Análisis y Diseño
Implementación
Pruebas Internas
Pruebas de Liberación
Despliegue
Soporte

A partir del estudio realizado y de las características del ciclo de vida, se definen en cada una de las etapas elementos en los que puede influir la confiabilidad del sistema. En Figura 1 se detallan los aspectos que se tienen en cuenta en la guía propuesta.

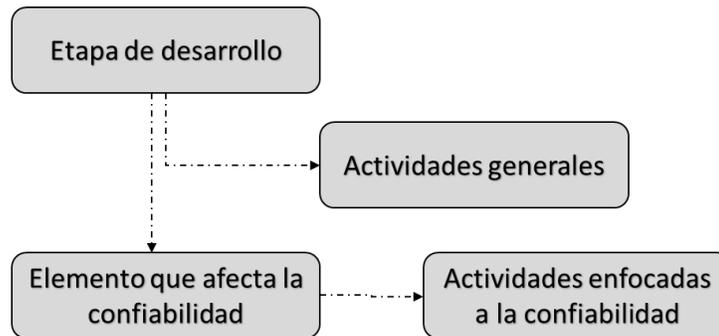


Figura 1. Elementos que contiene la guía.

Resultados y discusión

Guía de actividades a realizar por etapas

Cualquier proyecto de desarrollo de software sigue una metodología determinada, ya sea ágil o robusta. Es necesario aclarar que los proyectos en los que se va a implantar la guía propuesta son aquellos que desarrollan soluciones empleando el Programa de Mejora como apoyo para alcanzar la madurez de los procesos. En este punto se debe precisar que la guía debe ser aplicada por etapas de desarrollo, para las cuales se identifica el elemento que afecta la confiabilidad y se definen, además, las actividades a realizar para garantizarla; dichas actividades no aparecen especificadas por las metodologías y se deben efectuar de conjunto con las acciones propuestas por la que rige el proyecto.

Primeramente, se hace necesaria la definición de un conjunto de actividades que deben realizarse de forma transversal durante todo el ciclo de desarrollo que son de suma importancia para lograr efectividad en la guía que se propone a continuación:

1. Identificar previamente los riesgos que afectan la confiabilidad durante el ciclo de vida del proyecto y reflejarlos en un documento donde queden especificadas las acciones con respecto a ellos, ya sea mitigarlos, evitarlos o transferirlos.
2. Conformar el conjunto de tareas y responsabilidades a ejecutar en función de la capacitación del personal para garantizar la confiabilidad durante las etapas que define el Programa de Mejora.

3. Realizar revisiones de los productos del desarrollo, mediante listas de chequeo y revisiones técnicas formales después de que se termine cada etapa del ciclo de vida del Programa de Mejora, en pos de encontrar alguna anomalía y establecerlas en el registro de no conformidades.
4. Prevenir y eliminar errores en la documentación en cuanto a formato estándar (cumplimiento con el formato definido), redacción (ortografía, gramática, semántica, idioma) y aspectos técnicos (realización correcta de diagramas, gráficos y descripciones).

Teniendo definidos estos elementos, se muestra en la Tabla 3 cada una de las actividades propuestas por las etapas de desarrollo que conforman el ciclo de vida planteado por el Programa de Mejora de la universidad.

Tabla 3. Guía de actividades a realizar por etapas.

Etapa de desarrollo	Elemento que afecta la confiabilidad	Actividades para garantizar la confiabilidad
Estudio Preliminar	Alcance del sistema	Definir el alcance del sistema a implementar.
		Definir los posibles usuarios y clientes que va a tener el sistema y trabajar la confiabilidad desde ese punto de vista.
		Identificar el entorno de la organización donde se va a realizar el proyecto.
Modelación del Negocio	Reglas del Negocio	Identificar y definir las reglas del negocio sin ambigüedades, definiendo los flujos de validación y comportamiento del sistema.
		Definir reglas del negocio relacionadas con: Entidades. Actividades. Posibles usuarios que contribuyan a mejorar la fiabilidad.
		Definir los diferentes tipos de usuarios que utilizarán el sistema.
Requisitos	Requisitos funcionales	Entender las necesidades del cliente y transformarlas en requisitos entendibles por el equipo de desarrollo sin ambigüedades.
		Definir las entradas y salidas del sistema (datos, tipo de datos y restricciones de los mismos).
		Definir el nivel de fiabilidad que va a tener el sistema.
		Realizar el perfil de operaciones.
	Requisitos no funcionales	Definir los requisitos no funcionales que garanticen la disponibilidad y la confiabilidad.

		Elegir el hardware en correspondencia con la tecnología de software definida. (RAM, Microprocesador, Disco Duro, Tarjeta Gráfica, Conectividad, Periféricos)
		Definir de forma medible el rendimiento de la aplicación, el cual estará dado en el tiempo de respuesta de las funcionalidades arquitectónicamente significativas.
		Definir la cantidad de transacciones que va a realizar el sistema en un tiempo determinado.
		Definir la cantidad de transacciones en paralelo que soportará el sistema.
Análisis y Diseño	Casos de uso/Historias de usuario	Describir los casos de uso/ historias de usuario de forma clara, precisa y abarcar todos los flujos, tanto el flujo básico como todos los alternativos.
	Estilos arquitectónicos	Diseñar los mecanismos para garantizar el cumplimiento de los requisitos identificados en la etapa de requisitos a través de componentes del diseño.
		Definir el patrón de organización general.
		Identificar las interacciones que se establecen entre los componentes.
		Seleccionar correctamente los estilos y patrones arquitectónicos.
	Patrones de diseño	Utilizar patrones en torno a la confiabilidad que permitan lograr bajo acoplamiento, alta cohesión y modularidad.
Modelos de evaluación de la arquitectura	Utilizar modelos de evaluación de arquitectura.	
Implementación	Buenas Prácticas de Programación	Utilizar entornos de desarrollo adecuados para gestionar componentes.
		Utilizar buenas prácticas de programación como son los estándares de codificación.
		Realizar una correcta implementación en la aplicación de las funciones descritas en las especificaciones, teniendo en cuenta la complejidad de los algoritmos.

	Pautas definidas por la arquitectura	Tener en cuenta las pautas definidas por la arquitectura para el manejo de los errores y el control del flujo.
	Tratamiento de errores y respuesta del sistema	Tratamiento de errores y respuesta del sistema a los mismos.
		Realizar las validaciones necesarias para evitar la ocurrencia de errores.
		Usar procedimientos de control de cambios.
		Utilizar un mecanismo de captura de excepciones siguiendo las pautas definidas por la arquitectura.
Pruebas Internas	Mecanismos para realizar pruebas	Definir los mecanismos que permitan probar el comportamiento del sistema ante situaciones anormales.
		Discutir y documentar con el equipo de desarrollo los resultados de las pruebas realizadas.
	Pruebas de Confiabilidad	Realizar las pruebas de Confiabilidad (Pruebas de Estrés, Pruebas Reales, Pruebas de Destrucción Aleatorias, Pruebas de Integración, Pruebas Estructurales, Pruebas de Recuperación y Tolerancia a fallos).
		Diseñar casos de prueba para probar el perfil de operaciones (ver Anexo 1).
		Pruebas de aceptación por parte del cliente.
		Comprobar que el sistema cumple con las reglas del negocio y que las funcionalidades responden a los requisitos sin producir fallas.
Despliegue	Escenarios de despliegue	Disponer de un Plan para el despliegue detallado que deberá incluir factores críticos en el tiempo de recuperación y especificar configuraciones que contribuyan a la confiabilidad en los servidores de base de datos y web.
		Comprobar que el escenario de despliegue cumple con los requisitos no funcionales.
		Realizar pruebas pilotos.
Soporte	Elementos definidos en la implementación	Tener en cuenta los elementos definidos en la etapa de implementación.

Para la validación de la guía para el aseguramiento de la confiabilidad se hizo la selección de los especialistas que realizaron la valoración de la misma, se tuvo en cuenta una serie de características que posibilitaron obtener información de la trayectoria sobre los posibles especialistas. Basándose principalmente en la participación de estos

en eventos científicos, la labor que desempeñan actualmente, los años de experiencia en el tema arquitectura, calidad de software, desarrollo web y los conocimientos teóricos que tienen del tema tratado.

Basándose en las informaciones antes expuestas, la lista a consultar quedó conformada por 5 especialistas, que de una forma u otra están vinculados al área de arquitectura y calidad de software y cuentan con una ardua experiencia en el tema. A continuación, se muestra la Tabla 4 con el listado de especialistas.

Tabla 4. Listado de especialistas.

Listado de especialistas		Cargos
1	Doris Maza Oval	Especialista en calidad.
2	Yosviel Domínguez González	Arquitecto principal de SITPC
3	Yosvany Gómez Perdomo	Arquitecto de bases de datos de SITPC
4	Bárbara Almarales Lara	Analista principal de Portales
5	Adilaraima Martínez Barrio	Líder del proyecto SITPC

De acuerdo al cuestionario aplicado se evidencia que en una escala del 1 al 5, el promedio de conocimiento teórico que poseen los especialistas es de 4. En cuanto a la experiencia en la actividad práctica el promedio es de 4,8. Estos datos demuestran un nivel de competencia adecuado para la validación de esta propuesta, lo que da alto valor a sus criterios con respecto a cada una de las preguntas realizadas en el cuestionario.

En el cuestionario se formulan 8 preguntas destinadas a valorar los elementos definidos para asegurar la confiabilidad, durante las etapas del ciclo de desarrollo de software basado en el programa de mejora que lleva a cabo la universidad. Dichas respuestas se presentan a continuación para que sean comprendidas de la mejor manera.

Tabla 5. Respuestas a las preguntas aplicadas.

Preguntas	Muy Adecuado	Bastante Adecuado	Adecuado	Poco Adecuado	No Adecuado
Pregunta # 1	1	1	3		
Pregunta # 2	3	2			
Pregunta # 3	4	1			
Pregunta # 4		4	1		
Pregunta # 5	3	1	1		
Pregunta # 6	4	1			
Pregunta # 7		3	2		
Pregunta # 8	1	2	2		

Conclusiones

Con el desarrollo de este trabajo se dio cumplimiento al objetivo general concebido, arrojando a las siguientes conclusiones:

- ✓ Se comprobó a través de una entrevista que los proyectos de CEGEL pertenecientes a la UCI no utilizan ningún modelo para asegurar la confiabilidad en las aplicaciones web que desarrollan.
- ✓ Se escogió un modelo de calidad que abarca las subcaracterísticas de confiabilidad definidas en la investigación, que sirvió como guía para el desarrollo de la propuesta de solución.
- ✓ Se determinó en cada una de las etapas del ciclo de vida del software elementos que afectan la confiabilidad del sistema.
- ✓ Se definió la guía para el aseguramiento de la confiabilidad en los proyectos de CEGEL, con el propósito de determinar la aplicabilidad de la misma y verificar el cumplimiento de los objetivos para la cual fue creada, la misma fue validada satisfactoriamente por especialistas en el tema con un criterio valorado de un 100% entre Muy adecuado y Adecuado.

Referencias

- AXENTIA. (2006). Una Introducción a CMMI. Tratto il giorno abril 18, 2012 da <http://es.scribd.com/doc/69546265/WP03-Una-Introduccion-a-CMMI>
- BENCHIMOL, Daniel. Hacking desde Cero: Manuales Users (Spanish Edition). Creative Andina Corp., Argentina, 2011. ISBN: 987177303X, 9789871773039.
- BOEHM, B. W., BROWN, J. R., LIPOW, M. *Quantitative evaluation of software quality*. In Proceedings of the 2nd international conference on Software engineering, IEEE Computer Society, Los Alamitos (CA), USA, 1976; 592- 605.
- CALERO, Coral (2007). Grupo ALARCOS. *Modelos de calidad software*. [En línea] http://www.eici.ucm.cl/Academicos/R_Villarrol/descargas/calidad.produccion/WQM_Calero.pdf
- CALERO, Coral; CACHERO, Cristina; CÓRDOBA, Julio; MORAGA, M^a Ángeles. “PQM vs. BPQM: studying the tailoring of a general quality model to a specific domain”. *Advances in Conceptual Modeling–Foundations and Applications*. Springer Berlin Heidelberg, 2007. p. 192-201. ISBN 978-3-540-76291-1, 978-3-540-76292-8.

CALISOFT 2012. (s.d.). Tratto il giorno abril 1, 2012 da <http://calisoft.uci.cu/index.php/proceso-de-mejora>

ISO Standards Definition. (s.d.). Tratto il giorno marzo 10, 2012 da http://www.ehow.com/facts_5762097_iso-standards-definition.html

JTC1/SC7, ISO/IEC. 2007. CD 25010, *Software engineering-Software product Quality Requirements and Evaluation (SQuaRE) Quality model*. Quebec: s.n., 2007.

KATY ESPRONCEDA JOR, Y. L. (2010). Diseño de un procedimiento para evaluar la confiabilidad de los productos con tecnología Postgres, de DATEC.

LLANES, K. R. (2007). Propuesta para lograr fiabilidad en las aplicaciones bioinformáticas.

MCCALL, J. A.; RICHARDS, P. K.; WALTERS, G. F. *Factors in Software Quality: Concepts and definitions of software quality*, Volumes I. US Rome Air Development Center Reports, US Department of Commerce, USA, 1977. 168 pp. F030602-76-C-0417.

MCGRAW-HILL. (2007). *Ingeniería del Software, Un enfoque Práctico*. Sexta Edición, pág 436.

PRESSMAN, Roger S. *Ingeniería del Software. Un enfoque práctico*, Sexta Edición. McGraw-Hill, México, 2006. 958 pp. ISBN 970-10-5473-3.

QUIÑONES, E. (2006, 10 20). Modelos de calidad de software y software libre. Tratto il giorno 2 20, 2012.

SALANOVA, Pilar Esparza. “*Modelos de Calidad Web. Clasificación de Métricas*”. Director: Dr. Jesús M^a Minguet Melián. Proyecto de Fin de Carrera de Ingeniero Informático. Escuela Técnica Superior de Ingeniería Informática. España, 2006.

Anexos

Anexo 1: Pasos para realizar un perfil de operación.

1- Desarrollar perfil de operaciones

Tabla 4. Perfil de operaciones.

No. de la funcionalidad	Nombre de la Funcionalidad	Probabilidad
1	[Nombre del Caso de Uso o requisito funcional, historia de usuario u otro artefacto que defina la Metodología utilizada ordenado ascendentemente en	$\frac{\text{No. de la Funcionalidad}}{\text{Total de las funcionalidades}}$

	dependencia de la importancia de la funcionalidad para la arquitectura] Un caso de uso es arquitectónicamente significativo si: <ul style="list-style-type: none"> • El caso de uso resuelve un objetivo del negocio. • Las clases o componentes involucrados en la implementación del caso de uso son claves en la arquitectura. • El caso de uso es importante para el cliente (LLANES, 2007). 	
--	--	--

2- **Definir el Nivel de Falla:** se define lo que se considera como “falla” para el producto en desarrollo, así como los medios para identificarla. Esta definición es crítica para el proceso y debe ser constante durante todo el ciclo de vida.

Tabla 5. Nivel de Falla.

Severidad de la Falla	Falla	Impacto al Sistema	Cuantificación de la intensidad de la falla
Alta Media Baja	Descripción de la Falla.	Consecuencias para el sistema si ocurre la falla.	Cantidad de transacciones, páginas impresas, llamadas a funciones, accesos.

Anexo 2: Entrevista aplicada a los proyectos del Centro para verificar cómo aseguran la confiabilidad en sus productos.

1. ¿Cómo evalúan la confiabilidad del software? _____
2. ¿Utilizan algún modelo u estándar para asegurar la confiabilidad en sus productos? _____
3. ¿Cuáles son las principales fallas que ha presentado el software? _____

Anexo 3: Cuestionario de preguntas para validar la guía propuesta.

En la etapa de estudio preliminar: ¿Considera que el alcance del sistema es el elemento fundamental en la etapa estudio preliminar que influye en la confiabilidad del sistema?

En la etapa de Modelación del Negocio: ¿Considera usted que las reglas del negocio son el elemento fundamental que influye en la confiabilidad del sistema en la etapa de Modelación del negocio?

En la etapa de Requisitos: ¿Considera usted que la definición de los requisitos funcionales y no funcionales en la etapa de requisitos influye en la confiabilidad del sistema?

En la etapa de Análisis y Diseño: ¿Considera que la descripción correcta de los casos, así como seleccionar los estilos y patrones arquitectónicos son elementos primordiales que garantizan el aseguramiento de la confiabilidad durante la etapa de Análisis y Diseño?

En la etapa de Implementación: ¿Considera usted que tener en cuenta las pautas definidas y el manejo de los errores, las buenas prácticas de programación y el tratamiento de errores y respuesta del sistema a los mismos son elementos para garantizar la confiabilidad en la etapa de implementación?

En la etapa de Pruebas: ¿Considera usted que en la etapa de pruebas el comportamiento del sistema y las pruebas de confiabilidad (Pruebas de estrés, pruebas reales, pruebas de destrucción aleatorias, pruebas de integración, pruebas estructurales, pruebas de recuperación y tolerancia a fallos) son el elemento que influye en la confiabilidad del sistema?

En la etapa de Despliegue: ¿Considera usted que los escenarios de despliegue durante la etapa de Despliegue es un elemento para garantizar la confiabilidad?

En la etapa de Soporte: ¿Considera usted que las tener en cuenta los elementos definidos en la etapa de implementación son el elemento que garantiza la confiabilidad del sistema en la etapa de soporte?