

Tipo de artículo : Artículo original
Temática : Ingeniería y gestión de software
Recibido : 11/10/2020 | Aceptado : 27/12/2020

Gestión de la mantenibilidad desde etapas tempranas en el desarrollo de software

Management of maintainability from early stages in software development

Lisandra Tamayo Espinosa ^{1*} <https://orcid.org/0000-0003-1067-0983>

Nemury Silega Martínez¹ <https://orcid.org/0000-0002-8436-5650>

¹Universidad de las Ciencias Informáticas. Carretera a San Antonio de los Baños, Km 2 ½, Torrens, Boyeros, La Habana, Cuba. E-mail: {ltespinosa, nsilega}@uci.cu

*Autor para la correspondencia. (ltespinosa@uci.cu)

RESUMEN

Las empresas gastan entre sesenta y setenta por ciento de los recursos para corregir, adoptar y mantener el software existente. La inadecuada atención a la mantenibilidad, desde etapas tempranas, podría provocar que la mayor parte del tiempo que dedican los programadores al desarrollo, sea invertido en el mantenimiento. Un nivel adecuado de mantenibilidad aporta beneficios como: simplificación de la complejidad, reducción de los tiempos y los costos de mantenimiento. La mantenibilidad tiene un impacto significativo en la calidad general de una aplicación y en el éxito económico del producto de software. A partir de la importancia de esta característica en la gestión de software, se presenta un proceso que incluye las actividades a realizar en las disciplinas ingenieriles de desarrollo, así como los roles y artefactos de entradas y salida. Se define una lista de requisitos de mantenibilidad, para facilitar su identificación en los

proyectos de software, asociados a los factores que influyen en las sub-características mantenibilidad, planteadas en la Norma Cubana ISO/IEC 25010.

Palabras clave: software; mantenibilidad; sub-características; proceso; factores.

ABSTRACT

Companies spend between sixty and seventy percent of their resources to correct, adopt and maintain existing software. Inadequate attention to maintainability, from early stages, could cause most of the time programmers spend on development to be spent on maintenance. An adequate level of maintainability brings benefits such as: simplification of complexity, reduction of maintenance time and costs. Maintainability has a significant impact on the overall quality of an application and on the economic success of the software product. Based on the importance of this feature in software management, a process is presented that includes the activities to be performed in the engineering disciplines of development, as well as the roles and artifacts of inputs and outputs. It is defined a list of maintainability requirements, to facilitate their identification in software projects, associated to the factors that influence the maintainability sub-characteristics, raised in the Cuban Norm ISO/IEC 25010.

Keywords: software; maintainability; sub-characteristic; process; factors.

Introducción

La calidad de software en una empresa constituye un elemento diferenciador de competitividad y de imagen frente a sus clientes (Piattini, y otros, 2008; Irrazábal, 2015). Tanto la calidad del producto como la del proceso, son los aspectos más importantes actualmente en el desarrollo de software (Ebert, 2009; Irrazábal,

2015; Mendoza, 2018). Por este motivo los proyectos centran esfuerzos en gestionar la calidad del software usando modelos, normas, métricas, herramientas y estándares de referencia. La Norma Cubana ISO/IEC 25010: 2016 propone el modelo de calidad del producto con ocho características: adecuación funcional, eficiencia de desempeño, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad y portabilidad (NC ISO/ IEC 25010, 2016). La mantenibilidad se define como: la facilidad con la que un sistema o componente de software puede ser transformado para corregir fallos, obtener un mejor rendimiento o adaptarse a un entorno cambiante (IEEE, 1990; Arya, 2014; Soni, 2015; Abdullah, 2015; Dohmen, y otros, 2016; Pardo, 2018).

La mantenibilidad es el grado de efectividad, eficacia o eficiencia con la que un producto puede ser modificado (NC ISO/ IEC 25010, 2016; Calabrese, 2018; Pardo, 2018). Se considera un atributo de calidad particularmente interesante, ya que las actividades de mantenimiento representan el mayor costo en el desarrollo del software (Sommerville, 2001; Erazo, y otros, 2015; NC ISO/ IEC 25010, 2016). Realizar más esfuerzo durante el ciclo de vida para hacer que el software sea mantenible, puede reducir significativamente el total de los costos (Erazo, y otros, 2016). Por tanto, se puede afirmar que el aumento de la mantenibilidad es un requisito económico importante para las empresas con grandes sistemas de información (English, y otros, 2016). La falta de mantenibilidad del producto, está identificada por la escasa atención a los principales factores: complejidad, malas prácticas en la codificación, estructura no estandarizada y documentación no disponible (I. Gonzalez, y otros, 2019).

Para obtener evidencias empíricas sobre el impacto de la mantenibilidad, se realizó una encuesta en dos entidades desarrolladoras de software. Los resultados obtenidos mostraron que el 37% reconoce que en los proyectos en los que ha trabajado, no se definen los requisitos de mantenibilidad. El 60% afirma que la mantenibilidad debe gestionarse desde etapas tempranas en el ciclo de vida del software, para evitar consecuencias negativas. Un 50% de los encuestados asegura que si no se gestiona aumenta la dificultad para encontrar los fallos en el software, mientras que el 43% reconoce costos elevados de mantención y software poco reusable. Los problemas observados durante la investigación se deben en gran medida a la falta de buenas prácticas, por ejemplo: el 70% de los encuestados expresó la necesidad de capacitar a los equipos de desarrollo en temas de mantenibilidad, un 60% planteó que se debe incluir esta característica

durante todo el ciclo de desarrollo y para realizar las tareas, disponer de la persona que creó originalmente el software.

Los problemas constatados a partir de la revisión de la literatura, así como evidencias empíricas, llevaron a elaborar un proceso para la gestión de la mantenibilidad desde etapas tempranas en el ciclo de vida de desarrollo de software. Describiendo las actividades que se deben realizar en las disciplinas ingenieriles: Modelado de negocio, Requisitos, Análisis y diseño, Implementación y Pruebas. Estas actividades están asociadas a los factores que influyen en cada una de las sub-características de mantenibilidad planteadas por la Norma Cubana ISO/IEC 25010: 2016 (NC ISO/ IEC 25010, 2016).

Métodos o Metodología Computacional

Luego de realizada la encuesta para la evaluación del comportamiento de la mantenibilidad, se continuó la investigación con una revisión de la literatura. La bibliografía analizada abordaba los temas referentes a la gestión de la mantenibilidad, con el objetivo de identificar cómo diferentes investigadores trabajan la temática. El análisis bibliográfico confirmó que varios autores abordaban el aseguramiento y control de la mantenibilidad, lo que se pudo constatar por la abundante bibliografía encontrada, evidenciando la relevancia de la temática tratada en el presente artículo. Se identificaron 46 publicaciones cuyos temas especificaban la mantenibilidad, de ellas se consideraron relevantes para la presente investigación 15. De los trabajos seleccionados siete son publicaciones en revistas referenciadas, dos son libros pertenecientes a la literatura clásica relacionada con la materia y seis son tesis.

De la revisión bibliográfica resaltaron cinco artículos de los autores Irrazábal, Erazo y Rodríguez, que presentaban diferentes propuestas para evaluar, certificar y gestionar la mantenibilidad. En estas publicaciones se examinaban las sub-características de mantenibilidad planteadas por la ISO/IEC 25010: 2011, los factores y las buenas prácticas para el proceso de desarrollo de software, teniendo en cuenta el aumento de la mantenibilidad. Después de identificados los factores que influyen en las sub-características, se procedió a clasificarlos por disciplinas ingenieriles, teniendo en cuenta las propuestas por la metodología

de desarrollo AUP-UCI (Informáticas, 2015). Esta metodología es la encargada de estandarizar las actividades de desarrollo en la Universidad de las Ciencias Informáticas.

Resultados y discusión

Reconociendo el impacto que tienen los factores de mantenibilidad sobre la calidad del producto de software. Se realizó una clasificación de estos, teniendo como referencia los modelos presentados por los autores (Rodríguez, 2015; Erazo, 2015; Irrazábal, 2015; Erazo, 2016). Primeramente, los factores se relacionaron de acuerdo a su influencia en las sub-características de mantenibilidad planteadas por la Norma Cubana ISO/IEC 25010: 2016 (NC ISO/ IEC 25010, 2016). Tomando como referencia las definiciones tanto de los factores como de las sub-características a las que afectan. La segunda parte de la clasificación se efectuó vinculando los factores con las disciplinas ingenieriles: Modelado de negocio, Requisitos, Análisis y diseño, Implementación y Pruebas, como se puede observar en la Tabla 1.

Tabla 1 - Clasificación de los factores por sub-características y disciplinas.

Sub-características	Modularidad	Reusabilidad	Analizabilidad	Modificabilidad	Testabilidad	Disciplinas
Estandarización			x	x		Implementación
Documentación		x	x	x	x	Requisitos, Análisis y diseño, Implementación y Pruebas
Complejidad	x	x	x	x	x	Modelado de negocio, Análisis y diseño, Implementación y Pruebas
Estructuración	x	x	x		x	Modelado del negocio, Análisis y diseño, Pruebas
Código Duplicado			x	x	x	Implementación
Acoplamiento	x	x	x	x	x	Análisis y diseño
Cohesión	x	x	x	x	x	Análisis y diseño
Comentarios		x	x	x		Implementación

Fuente. Elaboración propia

La clasificación constituyó un aporte significativo para la creación de las actividades correspondientes al proceso para la gestión de la mantenibilidad. También se utilizó como base para la elaboración de la propuesta de requisitos no funcionales de mantenibilidad que seguidamente se presenta.

Propuesta de requisitos no funcionales de mantenibilidad

Para facilitar la identificación de los requisitos no funcionales de mantenibilidad (RNFM), se tuvo como premisa la clasificación realizada y los umbrales propuestos por la herramienta SonarQube. Esta herramienta se usa para medir la mantenibilidad del código fuente (SonarQube, 2019). Los resultados de otras investigaciones muestran que al incorporar SonarQube al desarrollo se mejora la mantenibilidad (González-Carvajal, y otros, 2019). Los requisitos propuestos se agrupan de acuerdo a las sub-características:

Modularidad

1. RNFM 1: Garantizar una baja complejidad entre las clases y los módulos.
2. RNFM 2: Lograr un acoplamiento bajo entre los módulos en las disciplinas de diseño e implementación.
3. RNFM 3: Mantener un alto grado de cohesión entre las clases del sistema.

Reusabilidad

1. RNFM 4: Usar excepciones en el código.
2. RNFM 5: Aplicar reglas de programación y optimización de código.
3. RNFM 6: Descomponer el sistema en pequeñas partes de funcionalidades.

Analizabilidad

1. RNFM 7: Mantener el porcentaje de código duplicado por debajo del 5%.
2. RNFM 8: Garantizar que la densidad de comentarios esté entre el 30% y el 40%.

Modificabilidad

1. RNFM 9: Garantizar la documentación actualizada.
2. RNFM 10: Asignar nombres a las variables consecuentes con la función que realizan.

Testabilidad

1. RNFM 11: Utilizar herramientas automatizadas como SonarQube.
2. RNFM 12: Mantener disponibles y actualizados los diseños de casos de prueba.
3. RNFM 13: Utilizar sentencias de código de fácil lectura.

Después de identificados los requisitos surgieron las actividades encaminadas al cumplimiento de estos. Las actividades junto a los roles, los artefactos de entrada y salida, forman en su conjunto el proceso para la gestión de la mantenibilidad que se describen a continuación.

Descripción del proceso para la gestión de la mantenibilidad

El proceso para gestionar la mantenibilidad desde etapas tempranas incluye los factores que influyen en las sub-características planteadas por la Norma Cubana ISO/IEC 25010: 2016. Describe las actividades en las fases: Inicio, Ejecución y Cierre; y las disciplinas ingenieriles propuestas por la metodología AUP-UCI (Informáticas, 2015). En las figuras 1 y 2 se presentan las actividades correspondientes a la Fase de Inicio y a las disciplinas: Modelado de negocio, Análisis y diseño e Implementación, pertenecientes a la fase de Ejecución.

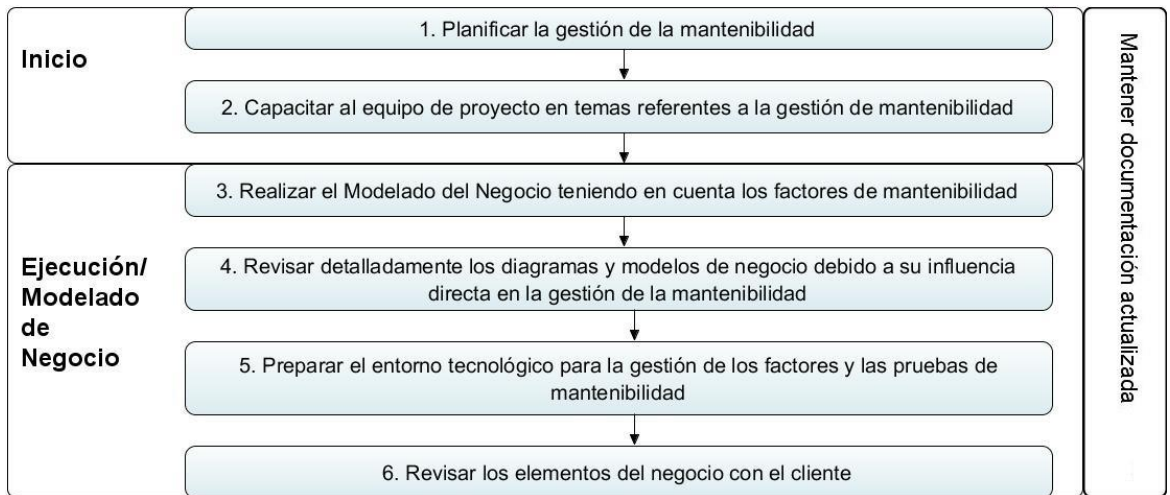


Fig. 1 – Actividades de Inicio y Modelado de negocio.

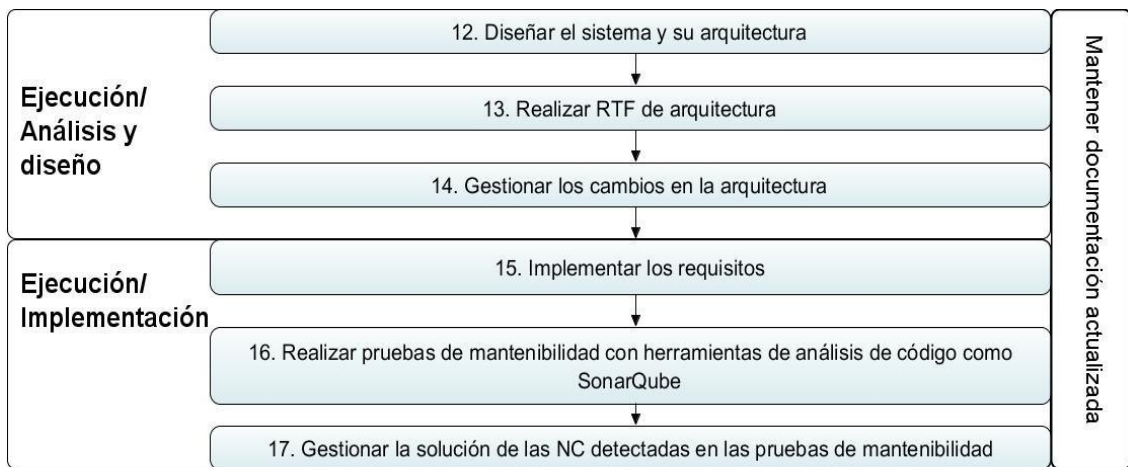


Fig. 2 – Actividades de las disciplinas Análisis y diseño e Implementación.

En las figuras 3 y 4 se observan las actividades de las disciplinas: Pruebas internas, Pruebas de liberación y Pruebas de aceptación, así como la actividad de la fase de Cierre.

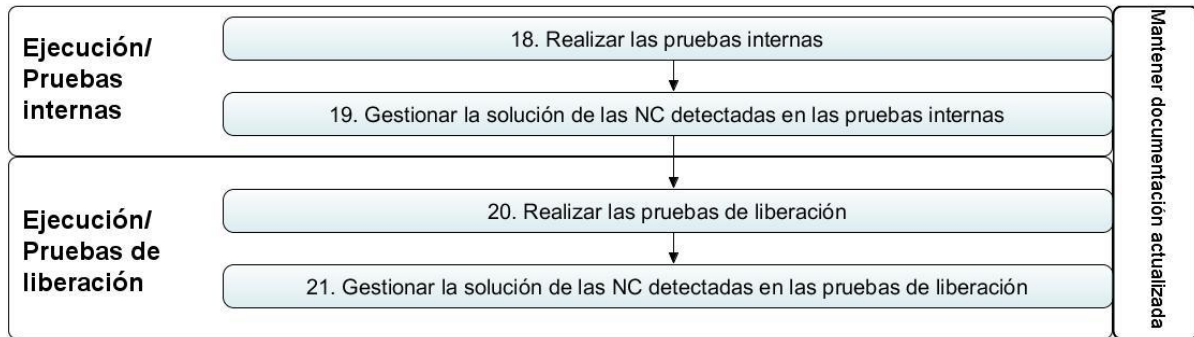


Fig. 3 – Actividades de las disciplinas Pruebas internas y Pruebas de liberación.

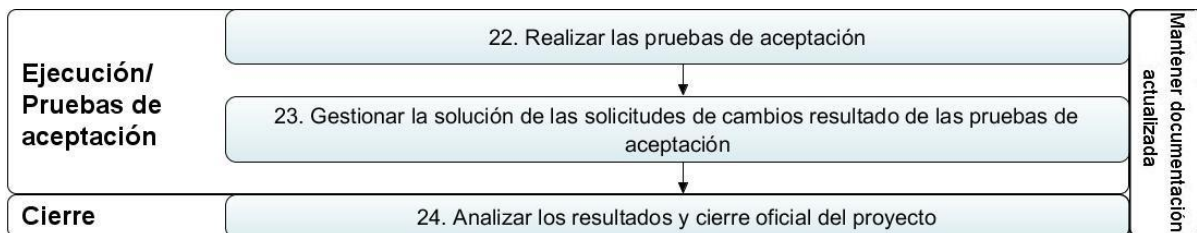


Fig. 4 – Actividades de la disciplina Pruebas de aceptación y de la fase de Cierre.

En la Tabla 2 se definen los roles, así como los artefactos de entrada y salida para las actividades presentadas anteriormente.

Tabla 2 - Descripción del proceso para la gestión de la mantenibilidad.

Fase /Disciplina	Rol	Entrada	Actividad	Salida
Inicio	Jefe de proyecto Administrador de la calidad	- Acta de inicio	1. Planificar la gestión de la mantenibilidad. Definir un plan de mantenibilidad donde se creen las actividades y tareas concernientes a su gestión.	- Plan de mantenibilidad - Cronograma del proyecto
	Jefe de proyecto Equipo de desarrollo Arquitecto	- Plan de mantenibilidad - Cronograma del proyecto	2. Capacitar respecto a los temas de mantenibilidad necesarios para su gestión. Presentar el plan de mantenibilidad. Explicar las responsabilidades y tareas de cada rol.	- Acta de compromiso con el Cronograma del proyecto - Plan de mantenibilidad

	Administradores Analista CCC		Exponer los factores que influyen en cada sub-características y su clasificación por disciplinas ingenieriles.	
Ejecución / Modelado de negocio	Jefe de proyecto Analista Arquitecto	- Documentos del negocio entregados por el cliente	3. Realizar el modelado teniendo en cuenta los factores que influyen en la mantenibilidad. Esta actividad se centra en especificar los procesos del negocio recibiendo como fuente la documentación entregada por el cliente y los factores.	- Modelo de negocio - Reglas de negocio - Modelo conceptual - Mapa de procesos - Descripción de los procesos de negocio - Glosario de términos
	Jefe de proyecto Analista Arquitecto Administrador de la calidad	- Modelos - Mapa de procesos - Descripción de los procesos de negocio	4. Revisar detalladamente los diagramas y modelos, debido a su influencia directa en la gestión de la mantenibilidad. Esta actividad se centra en evaluar la capacidad de los modelos para ser entendidos y modificados. Cumplimiento con los estándares de modelado.	- Artefactos del Modelado del negocio revisados y actualizados
	Jefe de proyecto Analista Arquitecto	- Plan de mantenibilidad - Artefactos del Modelado del negocio	5. Preparar el entorno tecnológico para la gestión de los factores y las pruebas de mantenibilidad. Instalar y configurar las herramientas. Disponer de la documentación y todo lo necesario para la gestión de los RNFM.	- Herramientas instaladas y configuradas
	Jefe de proyecto Analista Cliente	- Artefactos del Modelado del negocio	6. Revisar los elementos del negocio con el cliente. Analizar los factores que se tendrían presente para la gestión de la mantenibilidad. Examinar los elementos del negocio para detectar y corregir errores a tiempo.	- Minuta de reunión - Acta de aceptación del negocio - Artefactos del Modelado del negocio revisados y actualizados
Ejecución / Requisitos	Jefe de proyecto Analista	- Artefactos del Modelado del negocio	7. Definir los requisitos funcionales y RNFM, partiendo de los artefactos del negocio previamente aceptados por el cliente, los factores y RNFM propuestos.	- Especificación de los requisitos funcionales y RNFM
	Analista	- Especificación de los requisitos funcionales y	8. Describir los requisitos funcionales y RNFM. Se redacta el documento formal con la descripción detallada de los requisitos.	- Descripción de los funcionales y RNFM

		RNFM		
	Administrador de calidad Jefe de proyecto Analista Equipo revisor	- Expediente de proyecto - Lista de chequeo	9. Realizar la Revisión Técnica Formal (RTF) de los requisitos funcionales y aprobación de los RNFM. Para la RTF se emplearán listas de comprobación que medirán la completitud técnica de los requisitos, y que no existan inconsistencias con los artefactos con los que se relacionan.	- Lista de chequeo - Lista de defectos de los requisitos
	Administrador de calidad Jefe de proyecto Analista CCC	- Lista de defectos de los requisitos	10. Gestionar los cambios en los requerimientos. Evaluar y realizar los cambios a los requisitos de forma eficiente y efectiva, para asegurar la calidad y la mantenibilidad.	- Documentos del expediente de proyecto actualizados
	Jefe de proyecto Analista Cliente	- Documentos del expediente de proyecto	11. Revisar los artefactos de la disciplina de Requisitos con el cliente. En esta actividad se revisan detalladamente los artefactos elaborados y se firma el acta de aceptación de los Requisitos funcionales y los RNFM.	- Minuta de reunión - Acta de aceptación de los requisitos
Ejecución / Análisis y diseño	Jefe de proyecto Analista Arquitecto	- Documentos del expediente de proyecto	12. Diseñar el sistema y su arquitectura. En esta actividad se modelan las vistas de la arquitectura y el diseño, teniendo en cuenta los RNFM referentes a la complejidad, estructuración, acoplamiento y cohesión.	- Modelo de diseño - Vistas de la arquitectura
	Analista Arquitecto Equipo revisor	- Modelo de diseño - Vistas de la arquitectura - Lista de chequeo	13. Realizar RTF de Arquitectura. Para la RTF se emplearán listas de comprobación, que medirán la completitud técnica de los diagramas del diseño y las vistas de arquitectura. Verificar que cumpla con los objetivos de tipo funcional y de mantenibilidad.	- Lista de chequeo - Lista de defectos de la arquitectura
	Administrador de calidad Jefe de proyecto Arquitecto CCC	- Lista de defectos de la arquitectura	14. Gestionar los cambios en la arquitectura. Evaluar y realizar los cambios en la arquitectura de forma eficiente y efectiva, para asegurar la calidad y la mantenibilidad.	- Vistas de la arquitectura actualizadas
	Equipo de	- Estándares de	15. Implementar los requisitos. En esta	- Componentes del producto

Implementación	desarrollo Arquitecto Administrador de la Configuración Analista	codificación - Artefactos de Análisis y diseño	actividad se desarrollan todas las funcionalidades del sistema. Se deben tener presente los RNFM, referentes a la estandarización, complejidad, código duplicado, comentarios y trazabilidad.	
	Equipo de desarrollo Administrador de calidad	- Componentes del producto	16. Realizar pruebas de mantenibilidad con herramientas de análisis de código como SonarQube. Evaluar el cumplimiento de los RNFM referente a los atributos complejidad, código duplicado y comentarios.	- No conformidades (NC) detectadas.
	Administrador de calidad Equipo de desarrollo CCC	- Componentes del producto - Listado de NC detectadas	17. Gestionar la solución de las NC detectadas en las pruebas de mantenibilidad. Evaluar y realizar los cambios de manera eficiente y efectiva, para asegurar la calidad y la mantenibilidad.	- Componentes del producto corregidos - Documentos del expediente de proyecto actualizados
Ejecución / Pruebas internas	Analista Equipo de desarrollo Administrador de calidad	- Componentes del producto	18. Realizar las pruebas internas. El objetivo de las pruebas y el proceso de evaluación es verificar que el sistema desarrollado cumple con los requisitos funcionales y de mantenibilidad, utilizando herramientas automatizadas como SonarQube.	- Listado de NC detectadas.
	Administrador de calidad Equipo de desarrollo CCC	- Componentes del producto - Listado de NC detectadas	19. Gestionar la solución de las NC detectadas en las pruebas internas. Evaluar y realizar los cambios de manera eficiente y efectiva, para asegurar la calidad y la mantenibilidad.	- Componentes del producto corregidos - Documentos del expediente de proyecto actualizados
Ejecución / Pruebas de liberación	Analista Equipo de desarrollo Administrador de calidad Equipo revisor	- Componentes del producto	20. Realizar las pruebas de liberación. El objetivo de las pruebas y el proceso de evaluación es verificar que el sistema desarrollado, cumple con los requisitos funcionales y de mantenibilidad, utilizando herramientas automatizadas como SonarQube.	- Listado de NC detectadas.
	Administrador de calidad Equipo de	- Componentes del producto - Listado de NC	21. Gestionar la solución de las NC detectadas en las pruebas de liberación. Evaluar y realizar los cambios de manera	- Componentes del producto corregidos - Documentos del expediente

	desarrollo CCC	detectadas	eficiente y efectiva, para asegurar la calidad y la mantenibilidad.	de proyecto actualizados
Ejecución / Pruebas de aceptación	Jefe de proyecto Cliente	- Componentes del producto - Documentos del cliente	22. Realizar las pruebas de aceptación. Verificar que el software está listo para ser usado por los usuarios finales.	- Documentación de las pruebas - Acta de aceptación
	Administrador de calidad Equipo de desarrollo CCC	- Componentes del producto - Documentos de pruebas	23. Gestionar la solución de las solicitudes de cambios, resultado de las pruebas de aceptación. Evaluar y realizar los cambios de manera eficiente y efectiva, para asegurar la calidad y la mantenibilidad.	- Producto final
Cierre	Jefe de proyecto Cliente	- Producto final - Documentos del sistema	24. Analizar los resultados. Cierre oficial del proyecto. En esta actividad se analizan todos los resultados y se firma el cierre del proyecto.	- Acta de cierre del proyecto - Minuta de reunión

Fuente. Elaboración propia

La correcta gestión de la mantenibilidad incrementa las facilidades de modificación, al obtener productos de software con documentación actualizada y suficientes comentarios en el código. Provee beneficios en la reducción de los costos y recursos destinados a los cambios que se producen durante la fase de mantenimiento. El proceso para gestionar la mantenibilidad desde las fases tempranas, aumenta el valor de los proyectos y propicia la adaptación del producto a los cambios propios de la evolución en el desarrollo de software.

Descripción de los roles

Para la realización de las actividades del proceso y como propuesta del estudio de la problemática, surgió la idea de definir un conjunto de roles. Primeramente, se analizaron los definidos en el grupo Ingenieril y se detallaron sus responsabilidades durante el desarrollo de software. Se brindó especial atención a la definición de las tareas correspondientes a la gestión de la mantenibilidad. Algunas de las responsabilidades se modificaron, con el objetivo de garantizar que las actividades planteadas tuvieran un responsable para su ejecución (Blanco, 2013; Informáticas, 2015; UCI, 2017; Benítez, 2018; González, 2019):

Jefe de proyecto:

1. Participa en la definición del plan de mantenibilidad con las actividades y tareas concernientes a su gestión.
2. Revisa los elementos del negocio con el cliente, analizando los factores que se tendrían presente para la gestión de la mantenibilidad.

Analista:

1. Captura las necesidades de los clientes y las transforma en requisitos.
2. Realiza el modelado del negocio, diseña las pruebas, modela las vistas de la arquitectura y el diseño teniendo en cuenta los RNFM.
3. Participa en la capacitación con respecto a los temas de mantenibilidad.

Arquitecto de software:

1. Elabora el documento de arquitectura de software, participa en la RTF y gestiona los cambios en la arquitectura.
2. Participa en la capacitación de mantenibilidad, modela las vistas de la arquitectura y el diseño teniendo en cuenta los RNFM.
3. Prepara el entorno tecnológico para la gestión de los RNFM y las pruebas de mantenibilidad.

Administrador de la calidad:

1. Guía el diseño y la ejecución de las pruebas, monitorea la solución de las no conformidades y solicitudes de cambio.
2. Planifica la gestión de los RNFM y organiza la capacitación de mantenibilidad.
3. Participa en las pruebas de mantenibilidad y evalúa el cumplimiento de los RNFM.

Administrador de la configuración:

1. Planea el proceso de administración de la configuración y crea las líneas base.
2. Participa en la capacitación con respecto a los temas de mantenibilidad.
3. Prepara el entorno tecnológico para la gestión de los RNFM y las pruebas de mantenibilidad.

Equipo de Desarrollo:

1. Participa en la capacitación de mantenibilidad y realiza seguimiento de los RNFM.
2. Implementa los componentes del producto teniendo en cuenta los RNFM.

Comité de Control de Cambios (CCC):

1. Participa en la capacitación con respecto a los temas de mantenibilidad.
2. Analiza el impacto de los cambios, realiza su seguimiento y control.

Conclusiones

La clasificación de los factores por sub-características de mantenibilidad, favoreció la identificación de los requisitos no funcionales y la elaboración del proceso propuesta. Se elaboró un proceso para la gestión de la mantenibilidad desde etapas tempranas, teniendo en cuenta las fases del ciclo de vida del software, con las entradas y salidas para cada una de las actividades definidas por roles. La identificación de las actividades del proceso logró incluir los factores a las fases y disciplinas ingenieriles, así como a las sub-características de mantenibilidad. Se recomienda para trabajos futuros ampliar en el estudio de los factores que influyen en la mantenibilidad, así como identificar técnicas y herramientas para evaluarlos en las disciplinas de requisitos, análisis y diseño del software.

Referencias

Piattini, M., García, F., Garzás, J., & Genero, M. Medición y estimación del software: técnicas y métodos para mejorar la calidad y productividad del software. Ra-Ma (ed.), 2008, 121-127.

- Irrazábal, e. Mejora de la mantenibilidad con un modelo de medición de la calidad: resultados en una gran empresa. Xxi congreso argentino de ciencias de la computación, junín, 2015.
- Ebert, C. How Open Source Tools Can Benefit Industry: Open source software. IEEE software, 2009, 26(2): 50-51.
- Mendoza, C. C. ISO 25000. Facultad de Ingeniería E.A.P. Ingeniería de Sistemas e Informática, Nuevo Chimbote, Perú, 2018.
- NC ISO/ IEC 25010, N. I. I. Ingeniería de software y sistemas-Requisitos de la calidad y evaluación de software (SQuaRE)-Modelos de la calidad de software y sistemas. 2016.
- Ieee. IEEE Standard Glossary of Software Engineering Terminology. Institute of Electrical and Electronics Engineers, 1990.
- Arya, Y., & Khan, M. Measuring testability of object oriented design: A systematic review. International Journal of Scientific Engineering and Technology, 2014, 3(10): 1313-1319.
- Soni, N., & Khaliq, M. Maintainability Estimation of Object-Oriented Software: Design Phase Perspective”. International Journal of Advanced Research in Computer and Communication Engineering, 2015.
- Abdullah, M., & Srivastava, R. Testability Measurement Model for Object Oriented Design. 2015.
- Dohmen, T., Bruntink, M., Ceolin, D., & Visser, J. Towards a benchmark for the maintainability evolution of industrial software systems. Conference of the International Workshop, 2016.
- Pardo Mesias, S. R. Mantenibilidad de productos de software según el modelo ISO/IEC 25000. Universidad Nacional Agraria de la Selva, 2018.
- Calabrese Julieta, R. M. Asistente para la evaluación de calidad de producto de software según la familia de normas ISO/IEC 25000 utilizando el enfoque GQM. Facultad de Informática, 2018.
- Sommerville, I. Software Engineering. Addison-Wesley, 6th edition, 2001.
- Erazo, J., Florez, A., & Pino, F. J. Generando productos software mantenibles desde el proceso de desarrollo: El modelo de referencia MANTuS. Revista chilena de ingeniería, 2015, 24(3): 420-434.
- Erazo, J., Florez, A., & Pino, F. Análisis y clasificación de atributos de mantenibilidad del software: una revisión comparativa desde el estado del arte. Entre Ciencia e Ingeniería, 2016, 10(19): 40-49.
- English, M., Buckley, J., & Collins, J. J. Investigating software modularity using class and module level metrics. 2016.

González, I., Eréndira, A., Olivares, S. A., Salazar, M. A., & Mollinedo, J. M. Colaboraciones de Cuerpos Académicos en Innovación Empresarial. 2019.

Informáticas, U. D. L. C. Metodología de desarrollo para la Actividad productiva UCI. 2015.

Rodríguez, M., Pedreira, O., & Fernández, C. M. Certificación de la mantenibilidad del producto software: Un caso práctico. Revista Latinoamericana de Ingeniería de Software, 2015, 3(3): 127-134.

Sonarqube. Sonarqube. [En línea], 2019. [Consultado el 20/02/2019]. Disponible en: [\[http://www.sonarqube.org\]](http://www.sonarqube.org).

Gonzalez, Y. C., Gonzalo, M., & Manso, E. Mantenibilidad y productividad en la enseñanza de la Ingeniería del Software: análisis cuantitativo de un enfoque práctico. Departamento de Informática, Universidad de Valladolid, España, 2019.

Blanco, K. R. Proceso base de Ingeniería de Requisitos para las pequeñas y medianas empresas de desarrollo de software, Universidad de las Ciencias Informáticas, 2013.

Uci. Mejora de Procesos de Software. [En línea] Título de la publicación electrónica, 2017. [Consultado el 19/02/2019]. Disponible en: [\[http://mejoras.prod.uci.cu/\]](http://mejoras.prod.uci.cu/).

Benítez, Y. N. Guía para la gestión del requisito no funcional seguridad en el desarrollo de aplicaciones web. Universidad de las Ciencias Informáticas, 2018.

González, L. R. Guía metodológica para gestión de las modificaciones post-implementación en sistemas de gestión de software. Universidad de las Ciencias Informáticas, 2019.

Contribuciones de los autores

1. Conceptualización: Lisandra Tamayo Espinosa
2. Curación de datos: -
3. Análisis formal: Lisandra Tamayo Espinosa
4. Adquisición de fondos: -
5. Investigación: Lisandra Tamayo Espinosa
6. Metodología: -
7. Administración del proyecto: Lisandra Tamayo Espinosa, Nemury Silega Martínez
8. Recursos: Lisandra Tamayo Espinosa

9. Software: Lisandra Tamayo Espinosa
10. Supervisión: Lisandra Tamayo Espinosa, Nemury Silega Martínez
11. Validación: Lisandra Tamayo Espinosa
12. Visualización: Lisandra Tamayo Espinosa
13. Redacción – borrador original: Lisandra Tamayo Espinosa
14. Redacción – revisión y edición: Nemury Silega Martínez