

Confiabilidad del software, métricas, mejora de la confiabilidad usando Proceso ágil

Gurpreet Kaur¹ , Kailash Bahl²

Estudiante de PG en CSE en PIET

Docente en CSE en PIET

Resumen: El objetivo de este trabajo de investigación es estudiar las métricas de confiabilidad del software. La confiabilidad es uno de los aspectos importantes de cualquier software que no se puede ignorar y es difícil de medir. Según ANSI, "La confiabilidad del software se define como la probabilidad de que el software funcione sin fallas durante un período de tiempo específico en un entorno específico".

La confiabilidad del software es diferente de la confiabilidad del hardware. Lograr la confiabilidad del software es difícil porque la complejidad del software tiende a ser alta. La confiabilidad del software se puede clasificar en 3 partes: modelado, medición y mejora. Se pueden usar varios enfoques para mejorar la confiabilidad del software; sin embargo, es difícil equilibrar el tiempo y el presupuesto de desarrollo con la confiabilidad del software. Pero el mejor enfoque para asegurar la confiabilidad del software es desarrollar un software de alta calidad a través de todas las etapas del ciclo de vida del software. En este trabajo de investigación discutiremos sobre las métricas de confiabilidad del software. Las métricas utilizadas en forma temprana pueden ayudar en la detección y corrección de fallas en los requisitos que conducirán a la prevención de errores más adelante en el ciclo de vida del software. Este artículo proporciona una descripción general de las técnicas de medición de la confiabilidad del software.

Palabras clave

Software, Confiabilidad del Software, Métricas de Confiabilidad.

1. INTRODUCCIÓN

Hoy en día, en general, los sistemas son sistemas basados en software. El objetivo del sistema es satisfacer a los usuarios del sistema.

El software no se puede ver ni tocar, pero es esencial para el uso exitoso de las computadoras. Es necesario que la confiabilidad del software sea medida y evaluada. Se ha convertido

una parte crucial de muchos aspectos de la sociedad: electrodomésticos, telecomunicaciones, automóviles, aviones, compras, auditoría, enseñanza web, entretenimiento personal, etc.

La confiabilidad del software es la tarea clave para lograr la alta confiabilidad de cualquier industria de software. Aplica los atributos que son útiles para lograr la confiabilidad y se enfoca en las métricas. IEEE define la confiabilidad como "La capacidad de un sistema o componente para realizar sus funciones requeridas bajo condiciones establecidas durante un período de tiempo específico".

La confiabilidad del software se compone de tres actividades:

1. Prevención de errores
2. Detección y eliminación de fallas.
3. Mediciones para maximizar la confiabilidad, específicamente medidas que respaldan las dos primeras actividades

Ha habido un trabajo extenso en la medición de la confiabilidad utilizando el tiempo medio entre fallas y las métricas de tiempo medio hasta la falla.

La confiabilidad del software es un factor importante que afecta la confiabilidad del sistema. Se diferencia de la confiabilidad del hardware en que refleja la perfección del diseño, no la perfección de la fabricación. Este documento intenta dar una idea general de la confiabilidad del software y las métricas utilizadas para medir la confiabilidad del software. El objetivo de medir la confiabilidad del software es mejorar los posibles errores de software durante el proceso de diseño antes de lanzar el software al público. Por lo tanto, este documento también se centra en las técnicas de medición de la confiabilidad para mejorar la confiabilidad del software.

2. CONFIABILIDAD DEL SOFTWARE

La confiabilidad se puede definir como la probabilidad de que un elemento realice una función requerida en condiciones establecidas durante un período de tiempo específico. La confiabilidad del software se define como la probabilidad de falla en la operación del software libre durante un período de tiempo específico en un entorno específico. La falta de confiabilidad de cualquier producto se debe a fallas o presencia de fallas en el sistema. La falta de confiabilidad del software se debe principalmente a errores o fallas de diseño en el software. Ocurre solo cuando el sistema está en uso y no está precedido por advertencias.

3. CONFIABILIDAD DEL SOFTWARE VS CONFIABILIDAD DEL HARDWARE

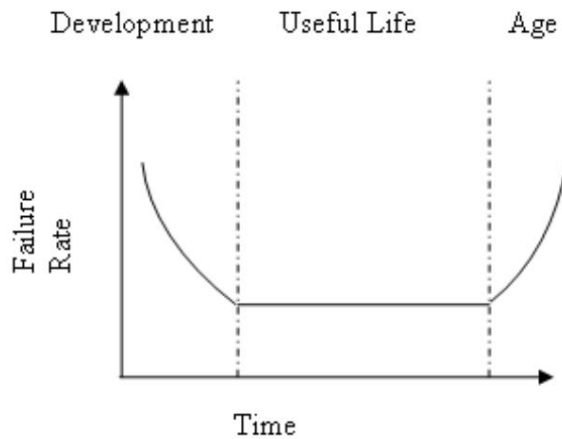
La confiabilidad del software se define como la función probabilística y viene con la noción de tiempo, pero en comparación con la confiabilidad del hardware, no es la función directa del tiempo.

El software no cambiará con el tiempo hasta que el programador lo cambie o actualice intencionalmente. Las partes electrónicas y mecánicas pueden volverse "viejas" y desgastarse con el tiempo y el uso, pero el software no se oxidará ni desgastará durante su ciclo de vida.

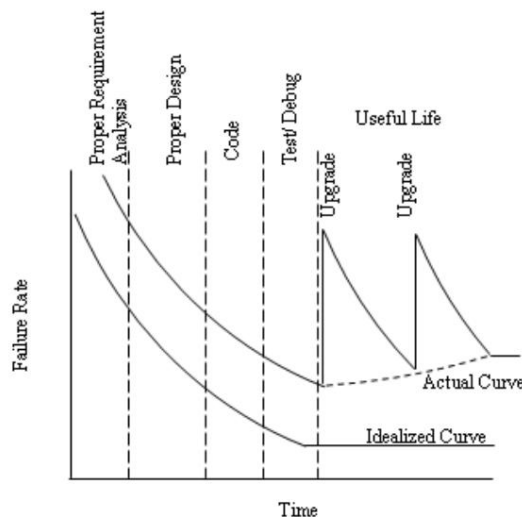
3.1 CURVA DE LA BAÑERA PARA LA FIABILIDAD DEL HARDWARE Y DEL SOFTWARE

Los errores de software van desde una interfaz de usuario mal diseñada hasta errores directos de programación. El software no se oxida, envejece, desgasta ni deforma. A diferencia de las partes mecánicas, el software permanecerá como está a menos que haya problemas en el diseño o en el hardware. El software no cambiará con el tiempo a menos que se cambie o actualice intencionalmente. Las fallas del software pueden ser

debido a errores, ambigüedades o mala interpretación de la especificación que se supone que debe satisfacer el software, descuido o incompetencia al escribir el código, pruebas inadecuadas, uso incorrecto o inesperado del software u otros problemas imprevistos. Con el tiempo, el hardware presenta las características de falla que se muestran en la Figura Conocido como curva de bañera.



Curva de bañera "Figura 1" para herrajes
Fiabilidad



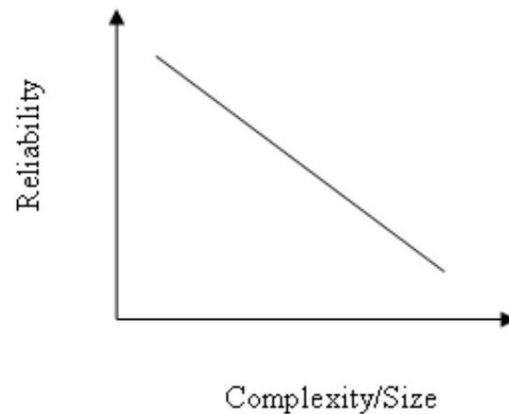
Curva de confiabilidad del software "Figura 2"

Hay dos diferencias principales entre las curvas de hardware y software.

- 1) Una diferencia es que en la última fase, el software no tiene una tasa de falla creciente como lo hace el hardware. En esta fase, el software se acerca a la obsolescencia; no hay motivación para ninguna actualización o cambio en el software. Por lo tanto, la tasa de fracaso no cambiará.
- 2) La segunda diferencia es que en la fase de vida útil, el software experimentará un aumento drástico en

tasa de error cada vez que se realiza una actualización. La tasa de fallas se estabiliza gradualmente, en parte debido a los defectos encontrados y corregidos después de las actualizaciones.

Dado que la confiabilidad del software sigue disminuyendo con el aumento de la complejidad del software, en la Figura 3 se muestra una posible curva.



"Figura 3"

4. MEDIDAS BÁSICAS DE FIABILIDAD

Las métricas de confiabilidad se utilizan para expresar cuantitativamente la confiabilidad del producto de software. La elección de qué métrica se utilizará depende del tipo de sistema al que se aplica y de los requisitos del dominio de la aplicación.

Medir la confiabilidad del software es un problema difícil porque no tenemos una buena comprensión de la naturaleza del software. Es difícil encontrar una forma adecuada de medir la confiabilidad del software y la mayoría de los aspectos relacionados con la confiabilidad del software. Incluso los tamaños de software no tienen una definición uniforme. Si no podemos medir la confiabilidad directamente, se puede medir algo que refleje las características relacionadas con la confiabilidad.

Algunas métricas de confiabilidad que se pueden usar para cuantificar la confiabilidad del producto de software se analizan a continuación:

4.1 TIEMPO MEDIO HASTA EL FALLO (MTTF)

MTTF se define como el intervalo de tiempo entre fallas sucesivas. Un MTTF de 200 significa que se puede esperar una falla cada 200 unidades de tiempo.

Las unidades de tiempo dependen totalmente del sistema e incluso se pueden especificar en el número de transacciones. MTTF es relevante para sistemas con transacciones largas. Por ejemplo, es adecuado para sistemas de diseño asistido por computadora en los que un diseñador trabajará en un diseño durante varias horas, así como para sistemas de procesadores de texto.

4.2 TIEMPO MEDIO ENTRE FALLAS (MTBF)

Podemos combinar las métricas MTTF y MTTR para obtener la métrica MTBF.

$$MTBF = MTTF + MTTR$$

Por lo tanto, un MTBF de 300 indica que una vez que ocurre la falla, se espera que la próxima falla ocurra solo después de 300 horas. En este caso, las medidas de tiempo son en tiempo real y no el tiempo de ejecución como en MTTF.

4.3 TASA DE OCURRENCIA DE FALLA

(ROCOF)

Es el número de fallas que ocurren en la unidad de intervalo de tiempo. El número de eventos inesperados durante un tiempo particular de operación. ROCOF es la frecuencia de ocurrencia con la que es probable que ocurra un comportamiento inesperado. Un ROCOF de 0.02 significa que es probable que ocurran dos fallas en cada 100 pasos de unidad de tiempo operativo. También se le llama métrica de intensidad de falla.

4.4 TIEMPO MEDIO DE REPARACIÓN (MTTR)

Una vez que ocurre la falla, se requiere algún tiempo para corregir el error. MTTR mide el tiempo promedio que lleva rastrear los errores que causan la falla y corregirlos.

4.5 PROBABILIDAD DE FALLO BAJO DEMANDA

(POFOD)

POFOD se define como la probabilidad de que el sistema falle cuando se solicita un servicio. Es el número de fallas del sistema dado un número de entradas del sistema.

POFOD es la probabilidad de que el sistema falle cuando se realiza una solicitud de servicio. Un POFOD de 0,1 significa que una de cada diez solicitudes de servicio puede fallar.

POFOD es una medida importante para los sistemas críticos de seguridad. POFOD es apropiado para sistemas de protección donde los servicios se demandan ocasionalmente.

4.6 DISPONIBILIDAD (AVAIL)

La disponibilidad es la probabilidad de que el sistema esté disponible para su uso en un momento dado. Tiene en cuenta el tiempo de reparación y el tiempo de reinicio del sistema. Una disponibilidad de 0,995 significa que de cada 1000 unidades de tiempo, es probable que el sistema esté disponible para 995 de ellas. El porcentaje de tiempo que un sistema está disponible para su uso, teniendo en cuenta el tiempo de inactividad planificado y no planificado. Si un sistema está inactivo un promedio de cuatro horas de 100 horas de operación, su DISPONIBILIDAD es del 96%.

5 TÉCNICAS DE MEDICIÓN DE LA CONFIABILIDAD DEL SOFTWARE

Las métricas de confiabilidad se utilizan para expresar cuantitativamente la confiabilidad del producto de software. La elección de qué métrica se utilizará depende del tipo de sistema al que se aplica y de los requisitos del dominio de la aplicación.

Medir la confiabilidad del software es un problema difícil porque no tenemos una buena comprensión de la naturaleza del software. Es difícil encontrar una forma adecuada de medir la confiabilidad del software y la mayoría de los aspectos relacionados con la confiabilidad del software. Incluso los tamaños de software no tienen una definición uniforme. Si no podemos medir la confiabilidad directamente, se puede medir algo que refleje las características relacionadas con la confiabilidad.

Algunas métricas de confiabilidad que se pueden usar para cuantificar la confiabilidad del producto de software se analizan a continuación:



Fig. 1: Factores de mejora de la calidad del software

5.1 MÉTRICAS DEL PRODUCTO

Las métricas del producto son aquellas que se utilizan para construir los artefactos, es decir, documentos de especificación de requisitos, documentos de diseño del sistema, etc. Estas métricas ayudan a evaluar si el producto es lo suficientemente bueno a través de informes sobre atributos como usabilidad, confiabilidad, mantenibilidad y portabilidad. En esto las medidas se toman del cuerpo real del código fuente. • Se cree que el

tamaño del software refleja la complejidad, el esfuerzo de desarrollo y la confiabilidad. Líneas de código (LOC), o LOC en miles (KLOC), es un enfoque inicial intuitivo para medir el tamaño del software.

La base de LOC es que la duración del programa se puede utilizar como predictor de las características del programa, como el esfuerzo y la facilidad de mantenimiento. Es una medida de la complejidad funcional del programa y es independiente del lenguaje de programación. • La métrica de puntos de función es un método para medir la funcionalidad de un desarrollo de software propuesto basado en el recuento de entradas, salidas, archivos maestros, consultas e interfaces.

- La métrica de cobertura de prueba estima fallas y confiabilidad realizando pruebas en productos de software, asumiendo que la confiabilidad del software es una función de la porción del software que se verifica o prueba con éxito.
- La complejidad está directamente relacionada con la confiabilidad del software, por lo que es importante representar la complejidad. Las métricas orientadas a la complejidad son un método para determinar la complejidad de la estructura de control de un programa, simplificando el código en un gráfico. representación. La métrica representativa es la métrica de complejidad de McCabe.
- Las métricas de calidad miden la calidad en varias etapas del desarrollo del producto de software. Una métrica de calidad importante es la eliminación de defectos



eficiencia (EDR). DRE proporciona una medida de calidad debido a varias actividades de control y garantía de calidad aplicadas a lo largo del proceso de desarrollo.

5.2 MÉTRICAS DE GESTIÓN DE PROYECTOS

Las métricas del proyecto describen las características y la ejecución del proyecto. Si hay una buena gestión del proyecto por parte del programador, esto nos ayudará a lograr mejores productos. Existe relación entre el proceso de desarrollo y la capacidad de completar los proyectos a tiempo y dentro de los objetivos de calidad deseados. Aumento de costos cuando los desarrolladores utilizan procesos inadecuados. Se puede lograr una mayor confiabilidad utilizando un mejor proceso de desarrollo, proceso de gestión de riesgos, proceso de gestión de configuración. Estas métricas informan sobre: -

- Número de desarrolladores de software
- Patrón de dotación de personal durante el ciclo de vida del software
- Costo y cronograma
- Productividad

5.3 MÉTRICAS DEL PROCESO

Las métricas de proceso cuantifican los atributos útiles del proceso de desarrollo de software y su entorno. Indican si el proceso está funcionando de manera óptima, ya que informan sobre atributos como el tiempo de ciclo y el tiempo de reelaboración. El objetivo de la métrica del proceso es hacer el trabajo correcto la primera vez durante el proceso. La calidad del producto es una función directa del proceso. Por lo tanto, las métricas de procesos se pueden usar para estimar, monitorear y mejorar la confiabilidad y la calidad del software. Las métricas de proceso describen la eficacia y la calidad de los procesos que producen el producto de software. Los ejemplos son:

- Esfuerzo requerido en el proceso.
- Tiempo para producir el producto
- Eficacia de la eliminación de defectos durante el desarrollo
- Número de defectos encontrados durante las pruebas
- Madurez del proceso

5.4 MÉTRICAS DE FALLAS Y FALLAS

Una falla es un defecto en un programa que surge cuando el programador comete un error y provoca fallas cuando se ejecuta bajo condiciones particulares. Estas métricas se utilizan para determinar el software de ejecución sin fallas.

Para lograr este objetivo, se recopilan, resumen y analizan la cantidad de fallas encontradas durante las pruebas y las fallas u otros problemas informados por el usuario después de la entrega.

Las métricas de fallas se basan en la información del cliente con respecto a las fallas encontradas después del lanzamiento del software. Por lo tanto, los datos de falla recopilados se utilizan para calcular la falla.

densidad, tiempo medio entre fallas (MTBF) u otros parámetros para medir o predecir la confiabilidad del software.

6 MÉTRICAS DE SOFTWARE PARA CONFIABILIDAD

Las métricas se utilizan para mejorar la confiabilidad del sistema al identificar las áreas de requisitos. Los diferentes tipos de métricas de software que se utilizan son: -

6.1 FIABILIDAD DE LOS REQUISITOS MÉTRICO

Los requisitos indican qué características debe contener el software. Especifica la funcionalidad que debe incluirse en el software. Los requisitos deben estar escritos de tal manera que no haya malentendidos entre el desarrollador y el cliente. Los requisitos deben contener una estructura válida para evitar la pérdida de información valiosa. Los requisitos deben ser minuciosos y detallados para que sea fácil para la fase de diseño.

Los requisitos no deben contener información inadecuada. Las métricas de confiabilidad de requisitos evalúan lo anterior factores de calidad del documento requerido.

6.2 FIABILIDAD DEL DISEÑO Y DEL CÓDIGO MÉTRICO

Los factores de calidad que existen en el plan de diseño y codificación son la complejidad, el tamaño y la modularidad. Los módulos complejos son difíciles de entender y existe una alta probabilidad de que se produzcan errores. La confiabilidad disminuirá si los módulos tienen una combinación de alta complejidad y gran tamaño o alta complejidad y pequeño tamaño. Estas métricas también son aplicables al código orientado a objetos, pero en este se requieren métricas adicionales para evaluar la calidad.

6.3 MÉTRICA DE CONFIABILIDAD DE LAS PRUEBAS

Estas métricas utilizan dos enfoques para evaluar la confiabilidad. En primer lugar, se asegura de que el sistema esté equipado con las funciones que se especifican en los requisitos. Debido a esto, los errores por falta de funcionalidad disminuyen.

El segundo enfoque es evaluar el código, encontrar los errores y corregirlos. Para garantizar que el sistema contenga la funcionalidad especificada, se escriben planes de prueba que contienen varios casos de prueba. Cada caso de prueba se basa en un estado del sistema y prueba algunas funciones que se basan en un conjunto relacionado de requisitos. El objetivo de un programa de verificación efectivo es garantizar que se pruebe cada requisito, lo que implica que si el sistema pasa la prueba, la funcionalidad del requisito se incluirá en el sistema entregado.

7 CONCLUSIÓN La

confiabilidad del software es un área de investigación importante y la confiabilidad del software es una parte clave de la calidad del software. Las métricas de confiabilidad del software se pueden usar para evaluar la confiabilidad actual y pronosticar el futuro. Para cualquier industria de software, lograr la confiabilidad del software es la tarea clave. Lograr la confiabilidad del software es difícil porque la complejidad del software tiende a ser alta.

Estas métricas miden la confiabilidad del software en los requisitos, el diseño y la codificación, y las fases de prueba. El alcance de este documento es el conocimiento de la medición y las métricas que son necesarias para garantizar la confiabilidad del proceso ágil de software. Entonces, utilizando varias técnicas de medición de software mencionadas en este documento, podemos eliminar cualquier error o falla del proceso de software, mejorando así la confiabilidad del producto de software.

REFERENCIAS

- [1] Michael R. Lyu, "Ingeniería de confiabilidad de software: una hoja de ruta".
- [2] Jiantao Pan, "Confiabilidad del software", 18-849b Sistemas integrados confiables, CMU, 1999.
- [3] Musa, Iannino y Okumoto, "Ingeniería de confiabilidad de software: medición, predicción, aplicación", Mc Graw Hill, 1987.
- [4] Michael R. Lyu, "Manual de ingeniería de confiabilidad de software". Publicación de McGraw-Hill, 1995, ISBN 0-07-039400-8.
- [5] Goutam Kumar Saha, "Problemas de confiabilidad del software: Mapa conceptual", IEEE Reliability Society 2009 Anual Informe de tecnología
- [6] Pardeep Ramachandran, Sarita Adve, Pradip Bose, Jude Rivers y Jayanth Srinivasan "Métricas para la confiabilidad de por vida" Agosto de 2006.
- [7] Vinay Tiwari¹, Dr. RK Pandey² Revista internacional de investigación avanzada en ingeniería informática y de comunicaciones vol. 1, número 10, diciembre de 2012 "software de código abierto y métricas de confiabilidad".
- [8] Nasib Singh Gill "Ingeniería de software: confiabilidad, pruebas y garantía de calidad del software.
- [9] Ritika Wasen P. Ahemed, M. Qasim Rafiq "Nuevo paradigma para la estimación de la confiabilidad del software" Volumen 44, No. 14, abril de 2012.
- [10] John C. Munson, Taghi M. Khashgoftear "Métricas de software para la evaluación de la confiabilidad.
- [11] Asia Quyoum, Mehraj – Ud - Din Dar, MK Quadri Di. "Mejora de la confiabilidad del software utilizando el enfoque de ingeniería de software: una revisión", International Journal of Computer Applications (0975 - 8887) Volumen 10– No.5, noviembre de 2010.
- [12] E Eduardo Valido-Cabrera, "Métodos de fiabilidad del software", Universidad Politécnica de Madrid Agosto, 2006.
- [13] ANSI / IEEE "Glosario estándar de terminología de ingeniería de software".
- [14] Vasilescu, B., Serebrenik, A., van den Brand, M. (2011). De ninguna manera: un estudio sobre la agregación de métricas de software. WETSoM'11 (24 de mayo de 2011), Waikiki, Honolulu, HI, EE. UU.
- [15] Mair, C., Shepperd, M. (2011) Juicio humano y métricas de software: visión para el futuro. WETSoM'11 (24 de mayo de 2011), Waikiki, Honolulu, HI, EE. UU.
- [16] Kececioglu, Dimitri, Manual de ingeniería de confiabilidad. Volumen 2, Prentice-Hall, 1991.
- [17] Musa, John, Anthony Iannino y Kazuhira Okumoto, Confiabilidad del software McGraw-Hill, 1987.
- [18] H. Pham, Confiabilidad del software, Springer, Singapur, 2000.