
Atributos de Calidad y Métricas

- Análisis biomédico de imágenes -

Tarea 1

Avila Marcello

Azofeifa Yaxiri

Cordero Kimberly

Román Nazareth

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computación
Aseguramiento de la Calidad del Software

Título:

Atributos de Calidad y Métricas

Tema:

Análisis biomédico de imágenes

Período del Proyecto:

Semestre I, 2017

Participante(s):

Ávila Marcello
Azofeifa Yaxiri
Cordero Kimberly
Román Nazareth

Supervisor(s):

M. Sc. Saúl Calderón Ramírez

Copias: 1

Cantidad de Páginas: 13

Fecha de publicación:

March 13, 2017

Abstract:

The following project refers to metrical analysis and reasons to deal with every small detail in software for the sake of the best possible solution of problems in the program's scope.

Contents

1	Introducción	3
2	Características	5
2.1	Funcionalidad	5
2.1.1	Interoperabilidad	5
2.1.2	Idoneidad	6
2.1.3	Exactitud	6
2.2	Confiabilidad	6
2.2.1	Tolerancia a Fallos	6
2.2.2	Madurez	7
2.2.3	Recuperabilidad	7
2.3	Usabilidad	8
2.3.1	Inteligibilidad	8
2.3.2	Atractividad	8
2.3.3	Capacidad de aprendizaje	9
2.4	Eficiencia	9
2.4.1	Comportamiento del tiempo	9
2.4.2	Recurso	10
2.4.3	Conformidad	10
2.5	Mantenibilidad	10
2.5.1	Analizabilidad	10
2.5.2	Estabilidad	11
2.5.3	Testabilidad	11
2.6	Portabilidad	12
2.6.1	Adaptabilidad	12
2.6.2	Instalabilidad	12
2.6.3	Co-existencia	12

Chapter 1

Introducción

A continuación se detalla cada una de las características de calidad y sus respectivas métricas de medición. Estas métricas tienen el fin de cuantificar el éxito o fracaso de una tarea al finalizar el sistema o durante su desarrollo.

Sirven como pruebas aisladas para desarrollar con un objetivo o alterando el código y utilizando pruebas regresivas.

Chapter 2

Características

2.1 Funcionalidad

2.1.1 Interoperabilidad

Peso: Low

Justificación

Eventualmente se podría querer integrar otros sistemas más adelante.

Métrica

Se medirá con ISIS (Integration of Software Intensive Systems), del Instituto de Ingeniería de Software de la Universidad Carnegie Mellon (EEUU), usando el modelo abstracto de interoperabilidad Levels of Information Systems Interoperability (LISI), LISI es un modelo técnico dirigido a categorizar sistemas; LISI es un modelo técnico dirigido a la definición, evaluación, medición y valoración de la interoperabilidad entre sistemas de información, ampliamente aceptado y que se apoya en un conjunto de herramientas útiles para el diagnóstico de la interoperabilidad a nivel técnico. El modelo LISI será tomado como base para medir la interoperabilidad que se puede dar entre el sistema de procesamiento de imágenes con otros sistemas de apoyo a la formación de interpretación de imágenes.

Herramienta

ISIS

2.1.2 Idoneidad

Peso: High

Justificación

El sistema debe desempeñar las tareas para las cuales fue desarrollado.

Métrica

Para su medición externa se mide la estabilidad en las especificaciones Funcionales, cuan estable son las especificaciones funcionales después de entrar a funcionar: Total de tiempo en ejecutar las pruebas dividido entre el total de funcionalidades probadas.

Herramienta

JUnit

2.1.3 Exactitud

Peso: High

Justificación

Los resultados obtenidos deben ser muy precisos debido a la naturaleza del proyecto.

Métrica

Internamente se medirá el cubrimiento de funcionalidades que sería la cantidad de casos de prueba dividido entre el total de funcionalidades probadas.

Herramienta

EclEmma

2.2 Confiabilidad

2.2.1 Tolerancia a Fallos

Peso: High

Justificación

Si ocurre un error debe ser manejado para no perder el avance que se lleva en el procesamiento de las imagenes.

Métrica

Se medirá utilizando redundancia dinámica, los componentes redundantes se activan cuando se detecta un fallo. Se mide en dos etapas en la detección y posterior recuperación de los fallos para ellos se utilizaran:

- Excepciones: proporcionando recuperación hacia adelante.
- Bloques de recuperación: proporcionando recuperación hacia atrás.

Herramienta

Metrics 3

2.2.2 Madurez

Peso: High

Justificación

El sistema no debería tener fallos recurrentes.

Métrica

Esta métrica mide cuando un atributo con la exención de fallos en el software, causados por la ocurrencia de fallos existentes en el propio software: o Erradicación de fallos: cuantos fallos ha sido corregidos

- Densidad de fallos totales contra casos de prueba: cuantos fallos totales fueron detectados durante un periodo de prueba definido
- Madurez de la prueba: está bien probado el producto

Herramienta

JUnit

2.2.3 Recuperabilidad

Peso: High

Justificación

Debido a la cantidad de señales a procesar, debe ser posible recuperar los datos en caso de fallo.

Métrica

Grado de solución ante fallos totales: cuantas condiciones de fallo total están resueltas.

Herramienta

Selenium

2.3 Usabilidad

2.3.1 Inteligibilidad

Peso: Medium

Justificación

Debe ser posible un sistema claro para los usuarios puntuales.

Métrica

Se medirá la capacidad de comprensión que tengan los usuarios del sistema, contar las funciones evidentes al usuario y comparar con el número total de funciones:

- Facilidad del manejo del sistema: porcentaje de personas que con facilidad manejaron el sistema dividido entre el porcentaje de personas con difícil manejo del sistema. Esto se mide con el tiempo que toman ejecutando alguna tarea nueva.

- Accesibilidad a demos: A que proporción de tutoriales pueden acceder los usuarios

Herramienta

Pruebas por usuarios guiándolos por las tareas.

2.3.2 Atractividad

Peso: Low

Justificación

La parte estética de la interfaz de gráfica no es necesario que sea muy elaborada.

Métrica

Cumplimiento de estándar de interfaz: como cumple la interfaz gráfica con los reglamentos aplicables para este proyecto.

Herramienta

Crazy Egg. Mapeo de zona de calor.

2.3.3 Capacidad de aprendizaje

Peso: Medium

Justificación

Los usuarios que van a aprender a usar el sistema son muy específicos.

Métrica

Eficiencia de la documentación del usuario y el sistema de ayuda en uso: que proporción de las funciones pueden ser utilizadas correctamente después de leer la documentación o uso del sistema de ayuda.

Herramienta

JMeter.

2.4 Eficiencia

2.4.1 Comportamiento del tiempo

Peso: High

Justificación

Los resultados deben obtenerse en un período corto de tiempo.

Métrica

Evaluar la eficiencia de las llamadas al SO y a la aplicación, estimando el tiempo de respuesta, se medirá:

- Parte de las especificaciones de diseño
- Probar la ruta completa una transacción
- Probar módulos o partes completas del sistema
- Medir la duración de respuesta del sistema en cada tarea: entre más corto mejor

Herramienta

JMeter y JUnit

2.4.2 Recurso

Peso: High

Justificación

El consumo de recursos debe ser de forma eficiente para el funcionamiento adecuado del sistema.

Métrica

Capacidad en tiempo que tarda el sistema para cargar y procesar una imagen entre el consumo recursos que requiere para el procesamiento

Herramienta

Sonarqube

2.4.3 Conformidad

Peso: Medium

Justificación

El uso de recursos debe cumplir con las expectativas del usuario.

Métrica

Porcentaje de aceptación por parte de los usuarios del consumo de recursos.

Herramienta

Crazy Egg.

2.5 Mantenibilidad

2.5.1 Analizabilidad

Peso: Medium

Justificación

Existe la posibilidad de que sea necesario conocer por qué razón existen fallos en el sistema o identificar qué partes necesitan una modificación.

Métrica

- Registrar la proporción de información sobre cambios en el sistema: número de cambios a funciones y módulos dividido entre el total de funciones del sistema
- Registrar los fallos en los módulos del sistema: número de fallos en las funciones dividido entre el total de funciones.

Herramienta

Sonarqube

2.5.2 Estabilidad

Peso: Low

Justificación

Los cambios en el software no apuntan a ser muy recurrentes.

Métrica

Se medirá contando la cantidad de cambios en módulos del sistema dividido entre la cantidad de módulos del sistema.

Herramienta

EclEmma

2.5.3 Testabilidad

Peso: Medium

Justificación

Es importante validar que las funcionalidades requeridas por el sistema trabajen de manera correcta.

Métrica

Número de pruebas ejecutadas con éxito.

Herramienta

JUnit

2.6 Portabilidad

2.6.1 Adaptabilidad

Peso: Medium

Justificación

El sistema debería contar con la capacidad de adecuarse a otros ambientes.

Métrica

Se medirá la eficiencia de adaptación en diversos ambientes:

-Prueba de ambiente para el sistema: tiempo de adaptación

-Fallos de adaptación: cantidad de fallos en las pruebas de adaptación en diferentes ambientes

Herramienta

Selenium

2.6.2 Instalabilidad

Peso: Low

Justificación

El sistema no requiere ser instalado muchas veces, ya que va orientado solamente para el laboratorio de la UCR.

Métrica

Medir el rendimiento de instalación del sistema: desempeño de duración de instalación entre deficiencias de la instalación

Herramienta

Selenium

2.6.3 Co-existencia

Peso: Medium

Justificación

Es posible que el sistema comparta recursos con otros sistemas.

Métrica

Cuán alto es el rendimiento que muestra el sistema al interactuar con otros sistemas. Contar el número de retrasos temporales que se dan dentro del sistema(latencia)

Herramienta

JMeter