Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computación

Tarea 1, atributos de calidad y métricas.

Curso:
Aseguramiento de la Calidad del Software

Profesor:
M. Sc. Saúl Calderón Ramírez

Estudiantes:

Jose Enrique Alvarado Chaves, 201129313
Reggie Stewart Barker Guillén, 2014050578
Joel Schuster Valverde, 2014096796
Joel Barrantes Garro, 2013120962

Primer Semestre 2017

Contents

1 Identificación de atributos y su prioridad

1.1 Funcionalidad

Calidad interna y externa		
CATEGORÍA	ATRIBUTO	PRIORIDAD (Baja / Media / Alta)
Funcionalidad	Adecuación	Media
	Exactitud	Alta
	Seguridad de acceso	Media

Table 1: Atributos de Funcionalidad

1.1.1 Adecuación

Este atributo es importante porque las soluciones ofrecidas por el sistema deben adecuarse a las soluciones esperadas por el usuario, o bien a los objetivos reales que este pretende lograr. Su prioridad es media ya que se trata de una aplicación específica y no muy extensa, y las funcionalidades pueden guiarse bajo la retroalimentación de los usuarios, para así garantizar que se adecue a sus requerimientos (diseño orientado al usuario).

1.1.2 Exactitud

Este atributo es importante porque el resultado debe ser suficientemente preciso y legible para el microbiólogo. Su prioridad es alta porque por medio del resultado el microbiólogo tomará una decisión respecto al tratamiento que utilizará.

1.1.3 Seguridad de acceso

Este atributo es importante porque toda información de carácter privado debe resguardarse de usuarios no autorizados, ya sea malintencionados o no. Su prioridad es media porque aunque la información que se maneja no es altamente delicada, debe protegerse de manipulaciones inadecuadas que puedan resultar en su pérdida.

1.2 Confiabilidad

Calidad interna y externa		
CATEGORÍA	ATRIBUTO	PRIORIDAD (Baja / Media / Alta)
Confiabilidad	Madurez	Alta
	Tolerancia a fallos	Media
	Capacidad de recuperación	Media

Table 2: Atributos de Confiabilidad

1.2.1 Madurez

Este atributo es importante porque permite medir la cantidad de fallos del sistema y su prioridad es alta debido a la importancia de poder tener una gran resistencia a los fallos, ya sea en el análisis de los elementos o en la conexión con el sistema.

1.2.2 Tolerancia a fallos

Este atributo es importante porque permite detectar y disminuir el impacto de los fallos en el sistema sin interrumpir la operación del resto de trabajos, y su prioridad es media puesto que no es un sistema crítico, pero es importante detectar y notificar de cualquier eventualidad al usuario.

1.2.3 Capacidad de recuperación

Este atributo es importante porque se espera que el sistema tenga cierta capacidad de recuperación de manera que no se entorpezca las tareas de los usuarios por un periodo prolongado, cuando se dé una falla no controlada. Su prioridad es media porque se espera que el sistema sea tolerante a fallas, y además porque su inoperancia, mientras no sea muy prolongada, no implica grandes pérdidas para la organización.

1.3 Usabilidad

.

Calidad interna y externa		
CATEGORÍA	ATRIBUTO	PRIORIDAD (Baja / Media / Alta)
Usabilidad	Capacidad para ser entendido	Baja
	Capacidad para ser aprendido	Media
	Capacidad para ser operado	Alta

Table 3: Atributos de Usabilidad

1.3.1 Capacidad para ser entendido

Este atributo es importante porque es importante que los usuarios comprendan no solo lo que hace el sistema, sino Su prioridad es baja porque se espera que los usuarios finales (microbiólogos del laboratorio) sean expertos en lo que hace el sistema, y por tanto, que lo comprendan con facilidad.

1.3.2 Capacidad para ser aprendido

Este atributo es importante porque debe ser fácil aprender a usarlo y su prioridad es media porque es importante que el usuario pueda aprender a usarlo de la mejor manera

1.3.3 Capacidad para ser operado

Este atributo es importante porque la interacción del usuario con el sistema a diario debe ser la más natural y su prioridad es alta porque la experiencia del usuario debe permitir una rápida operatividad de manera que pueda realizar sus labores agilmente.

1.4 Eficiencia

.

Calidad interna y externa		
CATEGORÍA	ATRIBUTO	PRIORIDAD (Baja / Media / Alta)
Eficiencia	Utilización de recursos	Media
	Comportamiento temporal	Alta
	Cumplimiento de la eficiencia	Alta

Table 4: Atributos de Eficiencia

1.4.1 Utilización de recursos

Este atributo es importante porque el entorno computacional del laboratorio de microbiología es descrito como "modesto", lo que indica que la memoria, CPU y disco deben aprovecharse de forma eficiente. Su prioridad es media porque el equipo no se ha catalogado directamente como "deficiente", pero de no considerarse podría resultar en un sistema lento o incapaz de cumplir con algunos lotes de trabajo.

1.4.2 Comportamiento temporal

Este atributo es importante porque la cantidad de imágenes que debe procesar el sistema es considerable (por ejemplo, lotes de 170.000) y debe entregar su resultado en un tiempo razonable. Su prioridad es alta porque cada imagen debe durar unos pocos segundos (alrededor de 10 segundos) en ser procesada para que el tiempo no se torne excesivo, y se debe dar un seguimiento constante a todos los módulos para verificar que ninguno ralentice el proceso.

1.4.3 Cumplimiento de la eficiencia

Este atributo es importante porque reduce el procesamiento sobre los elementos del sistema y su prioridad es alta porque reduce el tiempo o los recursos utilizados en cada uno de los módulos.

1.5 Mantenibilidad

Calidad interna y externa		
CATEGORÍA	ATRIBUTO	PRIORIDAD (Baja / Media / Alta)
Mantenibilidad	Capacidad para ser analizado	Alta
	Capacidad para ser cambiado	Media
	Capacidad para ser probado	Media

Table 5: Atributos de Mantenibilidad

1.5.1 Capacidad para ser analizado

Este atributo es importante porque permite identificar de forma rápida los errores y sus orígenes, a partir de la información obtenida de los fallos, y así brindar una solución sin gran retraso en el procesamiento del trabajo pendiente. Su prioridad es alta porque el sistema debe asegurar la correctitud de los resultados, y para ello, encontrar la razón de un fallo debe ser fácil a pesar de la complejidad de los algoritmos, incluso para un programador sin experiencia previa con el sistema.

1.5.2 Capacidad para ser cambiado

Este atributo es importante porque debe ser posible extender la funcionalidad del sistema para, por ejemplo, incorporar nuevos algoritmos al sistema de segmentación, y su prioridad es media porque el sistema puede requerir modificaciones, pero no con mucha frecuencia.

1.5.3 Capacidad para ser probado

Este atributo es importante porque debe saber las condiciones en las cuales el sistema es más débil y su prioridad es media porque el sistema no es crítico.

1.6 Portabilidad

Calidad interna y externa		
CATEGORÍA	ATRIBUTO	PRIORIDAD (Baja / Media / Alta)
Portabilidad	Coexistencia	Alta
	Adaptabilidad	Media
	Instalabilidad	Baja

Table 6: Atributos de Portabilidad

1.6.1 Coexistencia

Este atributo es importante porque permite compartir información con otros sistemas y es alta prioridad porque permite que los resultados del sistema sean de utilidad para sus dependientes.

1.6.2 Adaptabilidad

Este atributo es importante porque debe ser capaz de ejecutarse en los equipos de los microbiólogos y su prioridad es media porque se espera que los entornos no varíen demasiado entre sí, y tampoco requieran de mucho esfuerzo para ser adaptados y ejecutar el sistema.

1.6.3 Instalabilidad

Este atributo es importante porque el sistema debe ser fácil de instalarse y su prioridad es baja porque no presenta ningún riesgo a la hora de instalarlo.

2 Identificación de métricas

2.1 Funcionalidad

Calidad interna y externa			
ATRIBUTO	MÉTRICA	NIVEL REQUERIDO	HERRAMIENTA
Adecuación	Porcentaje de aceptación de los usuarios microbiólogos del laboratorio.	90% de los usuarios aprueban la funcionalidad del sistema.	Formulario(Google Forms)
Exactitud	Diferencia promedio entre la cantidad de células en una imagen detectadas por un especialista, con respecto a las detectadas por el sistema.	Diferencia promedio menor a 5 células en 100 imágenes.	JUnit
Seguridad de acceso	Cantidad de vulnerabilidades de acceso a la información privada del sistema.	Sin vulnerabilidades críticas de seguridad.	Nmap

Table 7: Métricas de Funcionalidad

2.1.1 Adecuación

Esta métrica se escogió porque cuanto más alta es esta métrica, más se ajusta el sistema a los requerimientos de los usuarios. Dada que la aprobación surge del usuario, se utilizará la herramienta Google Forms para obtener la opinión veraz y anónima de los usuarios del laboratorio y comprobar si el sistema está conforme a sus objetivos.

2.1.2 Exactitud

Esta métrica se escogió porque es necesario encontrar un margen de error promedio para los lotes de imágenes procesadas. El plugin JUnit permite diseñar pruebas unitarias respecto a referencias controladas, para así verificar la precisión de los resultados del sistema.

2.1.3 Seguridad de acceso

Esta métrica se escogió porque la cantidad de vulnerabilidades permite evaluar qué tanto riesgo corre la información a ser vista, manipulada o eliminada por usuarios o terceros que no deberían tener esos privilegios. La herramienta Nmap es muy útil, puesto que permite hacer un análisis automático de las vulnerabilidades más comunes y recomienda acciones correctivas frente a estas.

2.2 Confiabilidad

Calidad interna y externa			
ATRIBUTO	MÉTRICA	NIVEL REQUERIDO	HERRAMIENTA
Madurez	Porcentaje de aceptación de los usuarios microbiólogos del laboratorio.	Menos de 10 bugs y menos de 10 olores de código.	SonarQube:
Tolerancia a fallos	Cantidad de fallos controlados por el sistema por cada 1000 imágenes procesadas.	99% de fallos controlados.	Bitácora interna del sistema.
Capacidad de recuperación	Tiempo promedio que tarda el sistema en recuperarse de una falla.	Menos de 15 minutos en retomar el lote de trabajo.	Cronómetro interno del sistema.

Table 8: Métricas de Confiabilidad

2.2.1 Madurez

Esta métrica se escogió porque la cantidad de errores y malas prácticas de programación en el código estático determina qué tan propenso será a fallas e interrupciones del servicio, y la herramienta SonarQube analiza de forma automática los principales defectos y olores sin necesidad de ejecutar el sistema, es decir, se obtiene un diagnóstico instantáneo sin la necesidad de esperar a una etapa avanzada del desarrollo.

2.2.2 Tolerancia a fallos

Esta métrica se escogió porque un fallo no controlado representa la necesidad de actualización y mejoramiento del código. La herramienta usada nos permite encontrar la causa de los fallos de

forma sencilla.

2.2.3 Capacidad de recuperación

Esta métrica se escogió porque es un buen índice de la eficiencia del proceso de recuperación, al indicar el tiempo perdido, en el que sistema está inactivo hasta que vuelve a funcionar. La herramienta usada es usada porque permite llevar un control preciso de los tiempos en los que se originaron los fallos hasta que se reanudan los procesos, por lo que se calcula sin error esta métrica.

2.3 Usabilidad

Calidad interna y externa			
ATRIBUTO	MÉTRICA	NIVEL REQUERIDO	HERRAMIENTA
Capacidad para ser entendido	Nota promedio del usuario sobre la facilidad de entender el sistema	Nota promedio de 90 o más	Classmarker
Capacidad para ser aprendido	Nota promedio de los usuarios según un examen sobre la funcionalidad del sistema.	Nota promedio de 90 o más	Classmarker
Capacidad para ser operado	Cantidad de uso del mouse y el teclado para acceder a cualquier funcionalidad.	3 o menos clicks y usos del teclado.	Mousotron

Table 9: Métricas de usabilidad

2.3.1 Capacidad para ser entendido

La métrica se escogió porque la nota promedio de los usuarios es un buen estimador de qué tan claras son las instrucciones para poder acceder a las funcionalidades del sistema y si conocen realmente qué hace cada funcionalidad. La herramienta Classmaker permite crear exámenes, quices y cuestionarios para evaluar de forma didáctica la comprensión y conocimientos de los usuarios finales sobre el sistema, a partir de preguntas elaboradas para tal fin.

2.3.2 Capacidad para ser aprendido

La métrica se escogió porque le indica al desarrollador cuánto tiempo deben invertir los usuarios antes de poder usar el sistema con proficiencia, y permite saber si hay partes del software que deberían ser simplificadas. La herramienta Classmaker permite crear exámenes, quices y cuestionarios para evaluar de forma didáctica la comprensión y conocimientos de los usuarios finales sobre el sistema, a partir de preguntas elaboradas para tal fin.

2.3.3 Capacidad para ser operado

La métrica se escogió porque la cantidad de clics es un buen indicador de los pasos necesarios para acceder a una funcionalidad del sistema. Un número alto de clics indica que existe un grado alto de complejidad respecto al uso del sistema. La herramienta Mousotron permite medir de forma muy precisa acciones como la distancia recorrida con el mouse, los clics, las pulsaciones sobre teclas, el movimiento de la rueda central y otras interacciones de usuario que indican cuánto esfuerzo ha invertido el usuario, y puede ser útil para comparar este apartado en cada versión del sistema.

2.4 Eficiencia

Calidad interna y externa			
ATRIBUTO	MÉTRICA	NIVEL REQUERIDO	HERRAMIENTA
Utilización de recursos	Cantidad de memoria y espacio en disco utilizado	3GB o menos de RAM y 200GB o menos en disco	Resource Monitor
Comportamiento temporal	Tiempo promedio de procesamiento por imagen	Menos de 4 segundos	JUnit
Cumplimiento de la eficiencia	Cantidad de algoritmos por debajo del límite máximo de tiempo para procesar una imagen.	Menos de 1 segundo por cada algoritmo.	JUnit

Table 10: Métricas de eficiencia

2.4.1 Utilización de recursos

La métrica se escogió porque se indica que el laboratorio tiene un equipo no deficiente, pero sí modesto, que dependiendo de detalles específicos podría ser bastante limitada, es decir, indica qué tanto deben aprovecharse los recursos disponibles. La herramienta Resource Monitor permite llevar un seguimiento y registro de los recursos empleados durante toda la ejecución del sistema.

2.4.2 Comportamiento temporal

La métrica se escogió porque el sistema debe procesar una gran cantidad de imágenes por lotes (un lote puede tener 170.000 imágenes) y debe hacerlo dentro de un tiempo razonable para que los microbiólogos obtengan resultados sin esperar mucho más de una semana. La herramienta JUnit

permite medir los tiempos y resultados de ejecución de distintas pruebas y puede elaborarse para distintas dimensiones de imagen, por lo que es muy útil para analizar el rendimiento por cada unidad procesada.

2.4.3 Cumplimiento de la eficiencia

La métrica se escogió porque cada uno de los algoritmos debe apegarse a las limitaciones de recursos y tiempo, y se debe velar porque cada uno de los módulos del sistema sea rápido, eficiente, y capaz de rendir dentro del rango esperado, sin ocasionar retrasos al resto de los módulos ante una entrada inesperada. La herramienta JUnit es útil para medir los tiempos y resultados de ejecución de forma individual para cada componente del sistema, por lo que puede usarse para analizar el rendimiento de cada módulo en particular e identificar "cuellos de botella".

2.5 Mantenibilidad

Calidad interna y externa			
ATRIBUTO	MÉTRICA	NIVEL REQUERIDO	HERRAMIENTA
Capacidad para ser analizado	Relación entre la cantidad de comentarios por líneas de código.	Relación 1:10 entre comentarios y código	Cloc
Capacidad para ser cambiado	Índice de cohesión y acoplamiento LCOM*	LCOM debe ser nivel 1	Codecity 0.9.0alpha
Capacidad para ser probado	Índice de complejidad ciclomática	El índice de complejidad de cada función ciclomática no debe ser mayor a 10	Codecity 0.9.0alpha

Table 11: Métricas de mantenibilidad

2.5.1 Capacidad para ser analizado

La métrica se escogió porque una cantidad adecuada de comentarios permite entender con relativa facilidad cada módulo del sistema, incluso sin tener experiencia previa con el mismo, lo cual es vital si existe rotación en el equipo de desarrollo. La herramienta Cloc indica de forma automática la cantidad de líneas de código, líneas vacías y líneas de comentario que hay en el código fuente, a partir de lo cual es trivial calcular la relación entre líneas de código y líneas de comentario.

2.5.2 Capacidad para ser cambiado

La métrica se escogió porque el índice LCOM* (Lack of Cohesion Metric) indica el número de componentes conectados en una clase. Si es un índice mayor a 2, indica que la clase debe ser subdividida en otras clases, con el fin de obtener mayor granularidad en los componentes del sistema. El plugin de Eclipse, Metrics, calcula múltiples índices de rendimiento y métricas de calidad para evaluar atributos de calidad. En este caso, se usó para calcular el valor LCOM*.

2.5.3 Capacidad para ser probado

El índice de complejidad ciclomática se escogió como métrica porque mide las bifurcaciones en el flujo del código, y a partir de ello se puede determinar qué tan intrincado es desarrollar pruebas para comprobar y darle mantenimiento al sistema. La herramienta Polyscape calcula de forma automática el índice de complejidad ciclomática del código, y en caso de que este sea mayor a 2 sirve de alerta de que se deben hacer cambios y modularizar los métodos del sistema.

2.6 Portabilidad

Calidad interna y externa			
ATRIBUTO	MÉTRICA	NIVEL REQUERIDO	HERRAMIENTA
Coexistencia	Cantidad de programas que pueden leer la salida del sistema	Más de 10 programas	Repositorio Launchpad
Adaptabilidad	Cantidad de líneas de código modificadas para que sea funcional en otros sistemas.	Funcional en Linux y Windows	Github
Instalabilidad	Cantidad de pasos necesarios para instalar el sistema.	Menos de 10 pasos.	Manual de instalación

Table 12: Métricas de portabilidad

2.6.1 Coexistencia

La métrica se escogió porque es un valor que expresa de manera concisa cuán compatible es el sistema con herramientas existentes en el entorno y la herramienta launchpad permite paquetes de software con características deseables, como por ejemplo, paquetes que procesen archivos .csv..

2.6.2 Adaptabilidad

La métrica se escogió porque indica el esfuerzo necesario para poder adaptar el sistema a múltiples ambientes de trabajo. La herramienta para control de versiones GitHub permite visualizar los cambios realizados de una versión del sistema a otra.

2.6.3 Instalabilidad

La métrica se escogió porque el número de pasos para instalar un nuevo sistema indica que tan complicado o intuitivo es el proceso de instalación de dicho sistema. El manual de instalación del sistema indica la cantidad de pasos requeridos para el proceso de instalación del sistema.