[[1]](#footnote-1)

Diseño de experimentos Trabajo final Arquitectura de Computadores/ Interfaces y Arquitectura de Hardware(Mayo 2020)

Julián David Mabesoy, Juan Pablo Muñoz, Manuel David Quiceno Gallego

***Resumen***—Los estudios de simulación y predicción temprana de la complejidad del hardware y del software es esencial para conocer el comportamiento de ciertos fenómenos bajo diferentes escenarios virtuales. Y es el concepto de ***Análisis y Diseño de experimentos*** ***(DOE)***el término que describe muchos y muy variados casos de medición. Siendo este una medida estadística, estas medidas son aplicables a todo el ciclo de vida del desarrollo, desde la iniciación, cuando se deben estimar los costos, al seguimiento y control de la fiabilidad de los productos finales, y a la forma en que los productos cambian a través del tiempo debido a la aplicación de mejoras. Los casos deben estudiarse y seleccionarse cuidadosamente para cada cadena de herramientas dada. En este documento se utilizará un marco general de cinco (5) versiones con las que se propuso a trabajar, cada una de estas versiones tienen el mismo objetivo, y es invertir su color y almacenar de nuevo en los pixeles de origen.

1. **Introducción**

La metodología de análisis y diseño de experimentos se definen como una herramienta estadística, del grado en que un sistema, componente o proceso posee un atributo determinado.

Si realmente se desea conseguir la calidad de un software la medición es esencial, sirve para diseñar las condiciones ideales de un producto, proceso o servicio para que cumpla con nuestras expectativas usando el mínimo número de experimentos o pruebas y asi poder evaluar, mejorar y clasificar el software final, en donde serán manejadas dependiendo del entorno de desarrollo al cual pretendan orientarse.

Las mediciones son fundamentales para cualquier especialidad de la ingeniería y la ingeniería de software no se excluye de esto. Un ingeniero de software puede utilizar las mediciones para evaluar la calidad en los resultados del proyecto y para ayudar a tomar las decisiones más acertadas para contribuir a la mejora del producto.

Un Diseño de experimentos es una medida estadística aplicable a todo el ciclo de vida del desarrollo, desde la iniciación, cuando debemos estimar los costos, al seguimiento y control de la fiabilidad de los productos finales, y a la forma en que los productos cambian a través del tiempo debido a la aplicación de mejoras. Un ingeniero del Software recopila medidas y desarrolla métricas para obtener indicadores.

En la mayoría de los desafíos técnicos, las métricas ayudan a entender tanto el proceso técnico que se utiliza para desarrollar un producto, como el propio producto. Es decir, el producto se mide para intentar aumentar su calidad.

Sin embargo, no son absolutas ni son comprobaciones científicas sólidas. Proporcionan una manera sistemática de evaluar la calidad a partir de un conjunto de reglas definidas con claridad. En general, la medición persigue tres objetivos fundamentales: ayudarnos a entender qué ocurre durante el desarrollo y el mantenimiento, permitirnos controlar qué es lo que ocurre en nuestros proyectos y poder mejorar nuestros procesos y nuestros productos (Fenton y Pfleeger, 1997).

Razones para medir un producto:

* Para indicar la calidad del producto.
* Información confiable. La información de los factores que intervienen en un proceso (y su interacción entre ellos) es obtenida de manera más rápida, confiable y con evidencia estadística de lo que sería en campo, aplicado al proceso real.
* Para evaluar los beneficios en términos de productividad y de calidad, derivados del uso de nuevos métodos y herramientas de la ingeniería de software.
* Mejoras de procesos al determinar la mejor respuesta de salida que deseamos.
* Reducción de los costos de manufactura

Categorías

Las siguientes atienden las áreas más importantes de las métricas.

* Métricas para el modelo de análisis
* Métricas para el modelo de diseño
* Métricas para el código fuente
* Métricas para pruebas

Métricas para el modelo de análisis

En esta fase las métricas técnicas proporcionan una visión interna a la calidad del modelo de análisis. Estas métricas examinan el modelo de análisis con la intención de predecir el tamaño del sistema resultante. Es probable que el tamaño y la complejidad del diseño estén directamente relacionadas.

Métricas para el modelo de diseño

Proporcionan al diseñador una mejor visión interna. Ayudan a que el diseño evolucione a un nivel superior de calidad. Éstas se concentran en las características de la estructura del programa dándole énfasis a la estructura arquitectónica y en la eficiencia de los módulos. Estas métricas son de caja negra, en el sentido de que no se requiere ningún conocimiento del trabajo interno de ningún modo en particular del sistema.

Clasificación de métricas para el modelo de diseño:

* Métricas arquitectónicas
* Métricas al nivel de componente
* Métricas de diseño de la interfaz
* Métricas especializadas en diseño orientado a objetos

Métricas para el código fuente

Estás métricas asignadas como cuantitativas por Halstead, se derivan después de que se ha generado el código o se estima una vez que el diseño esté completo.

Un programa está compuesto de “tokens”, las instrucciones del lenguaje, los identificadores, constantes, operadores delimitadores de comentario y signos especiales del mismo. De esta forma se obtiene una medida más realista de la cantidad de información contenida en el código fuente.

Métricas para pruebas

Aunque se ha escrito mucho sobre métricas del software para pruebas, la mayoría de las métricas propuestas se concentran en el proceso de pruebas, no en las características técnicas de las pruebas mismas. Los responsables de la prueba, se guían por las métricas de análisis, diseño y código.

* Puntos de Función
* Bang
* Halstead
* Complejidad ciclomática

1. **Pregunta de investigación**

Dado el proyecto inicialmente mencionado y las importantes ventajas descritas que proporcionan *El Diseño de Experimentos* se obtiene la pregunta a partir de la cual se genera este proyecto de investigación: ¿Cómo es el desempeño en las versiones dadas a la hora de invertir el color de una matriz que representa una imagen de mapa de Bits?

1. **Objetivos**

El trabajo final tiene como objetivo la aplicación del diseño y análisis de experimentos con el fin de evaluar el desempeño de invertir el color de una matriz que representa una imagen de mapa de bits.

Cuantificar y explicar el impacto en el desempeño que tiene 5 versiones equivalentes de algoritmos en varían en la localidad espacial al recorrer sus pixeles. También se desea estudiar el impacto del tamaño de las imágenes en el tiempo de respuesta normalizado en la ejecución de la operación de invertir el color. Otra variable de interés es la afectación de la profundidad de color en el tiempo de ejecución, para ello se utilizará imágenes de 8, 16 y 32 bits por canal RGB.

OBJETIVOS GENERALES:

* Obtener un análisis de métrica por tiempo, en los cuales se evidencien cambios de comportamiento para un despliegue real.

OBJETIVOS ESPECÍFICOS:

* Determinar diferentes escenarios de relevancia para ejecutar pruebas a las cuales tomar las métricas.

1. **Metodología**

Elección de las métricas a implementar:

A partir del conjunto y clasificación de métricas anteriormente expuestos, se definen cinco de estas métricas a aplicar al videojuego, las cuales son:

* Latencia
* **Uso de RAM**
* Utilización y rendimiento de red
* **Carga de CPU**
* Rendimiento y capacidad del servidor

La elección presentada se hizo a partir del contexto del problema, dada la complejidad del producto medido, se considera únicamente pertinente la medición al código fuente y estabilidad de la aplicación, métricas de TI.

Elección de las herramientas de medición a usar:

Para el contexto del problema observado, un videojuego desarrollado en Java, se realizó una búsqueda y selección de una herramienta que permita tomar las mediciones correspondientes y que a su vez presente de manera significativa y clara los resultados obtenidos. De lo cual, se elige la librería Zabbix, la cual es un Sistema de Monitorización de Redes que facilita el proceso de medición en tiempo de ejecución de una aplicación Java,

1. **RESULTADOS**

Las

1. **conclusión**

Las métricas que podemos obtener del software son fundamentales para tener datos tangibles con los que hacer estimaciones, ya sean de recursos y por consiguiente de los costes, en Ingeniería del Software. Además de estas métricas también se encuentran las métricas KPI, para proyectos de mayor tamaño y por ende de mayores riesgos, en los que se debe asegurar el buen rendimiento del software, midiendo su performance constantemente, sin embargo, para una aplicación de menor tamaño como el videojuego analizado se concluye que las métricas más dicientes son las presentadas de TI.

Además, al usar un servidor de monitoreo como Zabbix, se encuentra la capacidad de prever múltiples fallas que se puedan presentar en un tiempo de ejecución ya sea en demostraciones de funcionalidad del videojuego o en un despliegue para jugadores reales, dado que se tiene un historial de cómo el programa ha funcionado a lo largo de tiempos controlados para diferentes configuraciones de escenarios reales que podrían ocurrir, respecto a recursos de máquina necesarios para su buen desempeño. Es decir, dado que en un despliegue real el objetivo y condiciones de usuario no es ejecutar únicamente el videojuego esta información es valiosa para que el usuario conozca las condiciones mínimas a reservar para un desempeño normal de la aplicación.

Finalmente, también es importante mencionar, tras hacer un primer uso de Zabbix, las observaciones sobre la herramienta elegida y las expectativas que se tenían sobre la misma en comparación a otra alternativa conocida en un momento tardío del proyecto de investigación, Prometheus. Esta herramienta presenta características como el análisis a un puerto específico, la entrega de resultados en gráficos disponibles en una imagen Docker generada por el mismo programa de monitoreo, es open-source y está disponible para diferentes sistemas operativos, lo cual son características de las que carece Zabbix, limitada al uso por linux. Dichas ventajas podrían aportar gran valor a la metodología de una futura investigación similar a la presente.

Referencias

1. K. Kaur, S. Anand. Review on Software and Hardware Reliability and Metrics. International Journal of Science, Engineering and Technology Research (IJSETR). Volume 2, Issue 5, May 2013.
2. IBM Knowledge Center. Rendimiento. (2015).
3. Zabbix Documentation 1.8. Taken from: [www.zabbix.com/documentation/1.8/manual/log\_file\_monitoring/how\_it\_works](http://www.zabbix.com/documentation/1.8/manual/log_file_monitoring/how_it_works).
4. IEEE (1990) IEEE Estándar Glossary of Software Engineering Terminology (Reporte Técnico del IEEENo. Std 610.12-1990).

1. [↑](#footnote-ref-1)