INVESTIGACIÓN __X__ PROFUNDIZACIÓN ____

Desarrollo de herramienta de corrección de defectos simples de software mediante análisis dinámico de código, ciencia de datos y machine learning

Autor

Luis Ángel Rodríguez Hernández

Universidad Distrital Francisco José De Caldas

Maestría en Ciencias de la Información y las Comunicaciones

Énfasis en Ingeniería de Software

Bogotá, Colombia

11 de noviembre de 2019

RESUMEN

Hay defectos de software que pueden detener la operación de un sistema y muchos de ellos tienen una corrección simple que puede ser automatizada.

La presente propuesta consiste en el desarrollo de una herramienta que resolverá los defectos mencionados mediante la combinación de análisis dinámico de código, ciencias de datos y machine learning.

Está solución traerá beneficios de tipo económico, de ahorro de tiempo y de generación de conocimiento para la industria del software.

1. PROBLEMA DE INVESTIGACIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

Todo software ha tenido y tiene defectos, pues estos son inevitables. No se puede decir que alguna aplicación no los tiene por el simple hecho que no se han identificado, pues será cuestión de tiempo para que se manifiesten o sean hallados.

Históricamente los defectos han acompañado a los sistemas informáticos; tal es el caso del fallo provocado por una polilla (bug) en un relé electromagnético en el Mark II en 1947 en la Universidad de Harvard. Otro de muchos casos conocidos por el año de 1962, es el defecto de transcripción de código (específicamente de una fórmula) causando que un cohete propulsor se desviará provocando la destrucción de la sonda Mariner I propiedad

de la NASA. Se pueden citar muchos casos conocidos y otros desconocidos; pero lo que tienen en común es que su solución es simple, es decir que, la diferencia entre la versión corregida y la versión con defecto es muy, pero muy pequeña.

Muchos defectos se corrigen con un pequeño ajuste en el código fuente, tal como: la inserción de una instrucción, realizar una validación, la modificación o eliminación de unas pocas líneas de código, o quizás un cambio en el orden de las mismas, entre otros casos; es decir, su solución, en la mayoría de los casos es simple.

Si bien, la corrección de los casos mencionados es simple, obtener la solución puede no serlo, pues requiere:

- Reproducir el fallo, que algunas veces no se logra tan fácilmente en un ambiente de pruebas.
- Tiempo de los ingenieros de software, de los profesionales de pruebas y muchas veces, de los usuarios finales.
- Habilidad técnica, experiencia y otras competencias intelectuales y profesionales para la depuración, investigación técnica y análisis causal.

Por otro lado, en la actualidad, el recurso humano dedicado a ingeniería de software es escaso y tampoco se dispone de tiempo; sumado a lo anterior, los usuarios y clientes finales demandan solución rápida a sus necesidades.

Solventar la problemática planteada involucra generación automática de código, sin embargo, los esfuerzos en este tipo de tecnologías han sido más enfocados en la creación de nuevo código fuente, tales como: Herramientas CASE, Model Driven Development, Model Driven Architecture, entre muchos otros; sin embargo, también se requiere la automatización de las correcciones, es decir, la modificación del código fuente existente, más que la creación de nuevo.

De ser posible lo anterior, traerá beneficios de tipo:

- Económicos, porque no se requiere de un ingeniero de software para dar solución a defectos simples reportados.
- De tiempo, ya que usuarios y clientes finales no tendrán que esperar demasiado por la solución del defecto.
- Mejora en el training profesional. Resulta extraño pero podrá ayudar en el entrenamiento de nuevos desarrolladores recién integrados a los equipos técnicos, porque podrán aprender de la solución obtenida por automatización.

1.2 FORMULACIÓN DEL PROBLEMA

Si bien, el proceso de desarrollo de software es una actividad intelectual y creativa, la corrección de la mayoría de defectos involucra pequeños cambios en el código fuente de las aplicaciones, los cuales podrían ser generados por herramientas software. Por lo tanto, queda la incógnita: **¿La corrección de defectos simples puede ser automatizable?**

1.1 SISTEMATIZACIÓN DEL PROBLEMA

Si la respuesta a la pregunta de la formulación de este problema es **sí**, entonces surgen otras incógnitas tales como:

- ¿Pueden ayudar las tecnologías de machine learning en dicha automatización?
- ¿Pueden ayudar las tecnologías de ciencias de datos a lograr el objetivo?
- ¿Se puede extraer conocimiento experto de los históricos y repositorios de código de fuente para la corrección automática de defectos?
- ¿Podría el análisis dinámico de código arrojar información relevante para lograr el objetivo?

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Desarrollar una herramienta que automatice la corrección de defectos simples de software.

2.2 OBJETIVOS ESPECÍFICOS

- Construir un módulo software que permita recolectar información del comportamiento de una aplicación mediante análisis dinámico de código, específicamente de defectos de software identificados.
- Emplear técnicas de ciencia de datos para la extracción de conocimiento experto desde históricos de código de fuente y desde repositorios de los mismos.
- Aplicar tecnologías de aprendizaje maquinal para obtener corrección automática a defectos de software.

FIN