

Minería de proyectos de software gestionados por Git (Resumen)

Héctor Vargas Rojas¹ y Jonathan Sandoval Sánchez¹

Escuela de Ingeniería,
Universidad Latinoamericana de Ciencia y Tecnología,
ULACIT, Urbanización Tournón, 10235-1000
San José, Costa Rica
hvargasr621, jsandovals204@ulacit.ed.cr
<http://www.ulacit.ac.cr>

Abstract. Los procesos de desarrollo y mantenimiento de software son complejos y, por lo general, se llevan a cabo de forma distribuida en diferentes países y regiones. Por lo que el uso de las herramientas de administración de versiones de código son de gran utilidad para apoyar el desarrollo de tareas asíncrona, cuyos resultados luego son combinados para obtener un producto final. Sin embargo, la realización de cambios por parte de un gran número de personas y de forma distribuida requiere que los desarrolladores y administradores de proyectos comprendan lo que están realizando los demás miembros del equipo. En consecuencia, el objetivo de este trabajo ha sido realizar la implementación de una herramienta para obtener detalles sobre los cambios efectuados a los diferentes elementos de los sistemas de software.

Keywords: Gestión de versiones de código, desarrollo de software, mantenimiento de software, minería de repositorios de software

1 Introducción

En la actualidad, existe gran dependencia del uso de dispositivos y sistemas de software, tanto a nivel empresarial como de entretenimiento y el hogar. El crecimiento de esta dependencia implica que cada vez se debe brindar mayor importancia a los procesos de desarrollo y mantenimiento de los sistemas de software. Por lo que el diseño y desarrollo de arquitecturas y herramientas para apoyar estos procesos han sido objeto de la atención, tanto de investigadores como de las empresas productoras de software. Esto se evidencia en el gran número de trabajos de investigación y publicaciones que se han efectuado en los últimos años (Charette, 2005; Hall, 2013; Gartner, 2013, 2014).

La evolución de software es el proceso de cambio y mejoramiento de los sistemas en el tiempo. Estos cambios son registrados mediante el uso de revisiones, pudiendo ser cientos o miles. El análisis de la evolución del software consiste en conocer qué se agregó, eliminó o cambió para el desarrollo o mantenimiento del sistema, así como saber quiénes han sido los responsables de estos eventos y la

fecha precisa cuando se realizaron. Este análisis permite a los administradores de proyectos tomar decisiones de acuerdo con los patrones de los cambios y las variaciones en la calidad del software, la cual es medida por métricas de calidad.

El estudio de la evolución permite el mejoramiento de los procesos de desarrollo, la predicción de errores, estimación de la productividad y la planificación de actividades de desarrollo, así como el recurso técnico y humano necesario.

Por lo cual el uso de herramientas que permita a las personas encargadas del desarrollo y mantenimiento de los sistemas sintetizar y comprender el avance de la evolución del software de forma visual resulta de gran utilidad.

De acuerdo con lo anterior, el objetivo de este trabajo es diseñar e implementar un sistema para realizar la minería de repositorios de software gestionados por Git con el fin de comprender los cambios realizados a los sistemas de software y facilitar los procesos de desarrollo y mantenimiento. En la siguiente sección, se detallan algunos de los resultados y conclusiones del presente trabajo.

2 Resultados y conclusiones

La información sobre el desarrollo y mantenimiento de los sistemas de software es gestionada por herramientas de administración de versiones como Git. Con esta información, es posible comprender la forma como los cambios han afectado el sistema, conocer cuándo fueron realizados y quienes han sido los responsables de efectuarlos.

Los siguientes son algunos de los datos que se pueden obtener haciendo uso de los repositorios de software:

- Ramas del desarrollo del sistema.
- Revisiones.
- Programadores.
- Fechas de los cambios.
- Archivos cambiados en cada revisión y detalles como los métodos y líneas de código afectadas.

De acuerdo con lo anterior, en esta investigación, se llevó a cabo el diseño e implementación de una herramienta para extraer información sobre las revisiones de los proyectos de software, y determinar los cambios que han sido efectuados en cada revisión, qué archivos fueron afectados y quién realizó los cambios, así como las relaciones que se establecen entre los programadores con base en el seguimiento de los proyectos de otros desarrolladores. La implementación se llevó a cabo en JavaScript, Node.js y MongoDB utilizando el API de GitHub para extraer la información.

La implementación realizada es el primer paso para el desarrollo de una herramienta de analítica visual que se integrará como un plugin de un IDE basado en la nube (por definir). Pero además, se integrará en un framework de extracción de información de otras herramientas de gestión de versiones de código.

Con base en lo anterior, el resultado de este trabajo ha permitido evaluar una serie de APIs disponibles para obtener información de GitHub, pero además, ha permitido sentar las bases para una implementación más amplia. Finalmente, la amplia y clara documentación que se ha realizado de cada uno de los componentes desarrollados permitirá la continuación del proyecto por parte de otros investigadores o desarrolladores.

References

- Charette, R. N. (2005, Sept). Why software fails [software failure]. *IEEE Spectrum*, 42(9), 42-49. doi: 10.1109/MSPEC.2005.1502528 pages 1
- Gartner, I. (2013, July). *Gartner says worldwide it spending on pace to reach 3.7 trillion in 2013*. Website. Retrieved from <http://www.gartner.com/newsroom/id/2537815> pages 1
- Gartner, I. (2014, January). *Gartner says worldwide it spending on pace to reach 3.8 trillion in 2014*. Website. Retrieved from <http://www.gartner.com/newsroom/id/2643919> pages 1
- Hall, J. G. E. S. L. (2013, March). *Gartner it key metrics data: 2013 it enterprise summary report*. Website. Retrieved from http://blog.rackspace.co.uk/wp-content/uploads/2013/09/Gartner_Enterprise_Summary_Report_2013.pdf pages 1