UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática

TFG del Grado en Ingeniería Informática

Revisión automática de calidad de proyectos con SonarQube

Presentado por Plamen Petyov Petkov
en Universidad de Burgos — 30 de marzo de 2016
Tutor: Carlos López Nozal

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática

D. Carlos López Nozal, profesor del departamento de Ingeniería Civil, área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Plamen Petyov Petkov, con DNI X7026351N, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Revisión automática de calidad de proyectos con SonarQube.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 30 de marzo de 2016

Vº. Bº. del Tutor:

D. Carlos López Nozal

Resumen

Con la creciente industria del software de los últimos años, las grandes empresas se han encontrado con la necesidad de realizar medidas de calidad que guíen y faciliten el proceso de desarrollo de productos software de calidad. Ésto ha originado la creación de diversos estándares y normas, así como herramientas basadas en éstos, con el fin de realizar las medidas de calidad de forma más automatizada.

En este contexto, el proceso de medición de calidad de software se puede aplicar también a los Trabajos final de Grado del Grado de Ingeniería Informática.

El objetivo principal de este trabajo será la creación de un entorno que permita aplicar diversas medidas de calidad de software a los TFG del Grado de Ingeniería Informática de la Universidad De Burgos, utilizando para ello la herramienta SonarQube.

Descriptores

Calidad de software, métricas, ISO 9126, SonarQube, OpenShift.

Abstract

With the growing software industry in recent years, big software companies had faced the need of quality measurements to guide and make easier the process of quality software product development. This has led to the creation of many standards and regulations, as well as the creation of tools based on those standards, in order to make the quality measurement process more automatic.

In this context, the software quality measurement process can also be applied to the Degree Final Courseworks of the Degree in Computer Science (comprobar traducciones de TFG y Grado en Ingeniería Informática).

The main goal of this coursework will be the creation of an environment which allows the application of different software quality metrics to the Degree Final Courseworks of the Degree in Computer Science of the University of Burgos, using the SonarQube tool for the purpose.

Keywords

Software quality, metrics, ISO 9126, SonarQube, OpenShift.

Índice general

Índice de figuras

Técnicas y herramientas

4.1. Técnicas

En este apartado se hace una breve explicación de las técnicas utilizadas en el desarrollo del trabajo, así como del proceso llevado a cabo para realizar la gestión del mismo. Para el desarrollo del proyecto se ha decidido adoptar la metodología *Scrum*, que se explica a continuación.

Metodología Scrum

Scrum es una metodología iterativa e incremental, utilizada en el desarrollo y gestión ágil de proyectos.

En el desarrollo mediante Scrum se comienza a partir del llamado *product backlog*. El product backlog es básicamente una lista en la que se especifican las características y funcionalidades ordenadas según prioridad, que sirve como guía en el desarrollo del producto. En el product backlog se incluyen tareas relativas al desarrollo, tales como, diseño, imlpementación y pruebas.

Dichas tareas se llevan a cabo por un equipo mediante iteraciones cortas, que normalmente duran entre una y cuatro semanas. Este intervalo de tiempo también se denomina *sprint*.

Al final de cada sprint se realiza una retrospectiva de las tareas completadas obdeniendo así el producto final o un incremento del mismo. También se llevan a cabo reuniones con el propietario para revisar el producto y obtener feedback, que sirve como base para elaborar el product backlog para el siguiente sprint.

Todo el proceso iterativo continúa hasta obtener el producto definitivo.

Proceso de gestión

Como se ha mencionado al inicio de este apartado, se ha optado por utilizar Scrum para la realización y gestión del proyecto.

Toda la gestión del trabajo se ha llevado a cabo en un repositorio en la plataforma Bitbucket (sección 4.2). Para ello se ha utilizado el gestor de incidencias Jira (sección 4.2) que proporciona el repositorio del proyecto.

En el progreso del trabajo se han realizado sprints con el tutor, cada dos semanas. En cada sprint se han revisado las tareas completadas y se han especificado las tareas a realizar para el siguiente. Para cada tarea del sprint se ha creado una incidencia a través del gestor, indicando el tipo (en este caso tarea), hito correspondiente a la tarea y un componente o producto resultante.

Para gestionar el códgio fuente del proyecto se ha utilizado la herramienta de control de versiones Git (sección 4.2) en conjunto con el repositorio del proyecto. De esta forma, cada tarea relacionada con el desarrollo de código fuente tiene asociado uno o varios *commits*.

(Explicar la razón para elegir Scrum como metodología de desarrollo para el proyecto)

4.2. Herramientas

SonarQube

SonarQube [4][1] es una plataforma open source de gestión de calidad de código fuente. La herramienta consta de un servidor web (también denominado *dashboard*) en el que se muestra el estado actual o *snapshot* de la calidad del proyecto través de diversos indicadores.

Inicialmente SonarQube esteba destinado al contexto de Java (sección 4.2), puesto que está desarrollado en este lenguaje de programación; actualmente es una plataforma multilenguaje que soporta más de 20 lenguajes de programación, incluyendo C#, C/C++, Python, PHP y Cobol, entre otros.

Una de las características importantes de SonarQube es su alto grado de personalización. El dashboard puede ser fácilmente adaptados a los criterios de calidad específicos que se necesiten. Toda la persoalización, tanto métricas como lenguajes de programación soportados, se hace a través de plugins proporcionados por la comunidad de SonarQube, la mayoría de forma gratuita. SonarQube también permite el desarrollo de plugins propios a través de su API específica para ello.

En este trabajo se utiliza SonarQube (versión 5.4[1]) como base para la plataforma de evaluación de trabajos fin de grado. Dicha plataforma de eva-

[1] Versión disponible en http://www.sonarqube.org/downloads/

luación será el resultado de desplegar una instancia del servidor de SonarQube en la plataforma OpenShift.

OpenShift

OpenShift [2][3] es una plataforma PaaS (Platform-as-a-Service) destinada al despliegue y hosting de aplicaciones en la nube. Actualmente existen tres versiones diferentes: OpenShift Origin, OpenShift Enterprise y OpenShift Online. Tanto OpenShift Origin, como OpenShift Enterprise requieren de la administración de la plataforma PaaS, por lo que no se contemplan en el trabajo.

OpenShift Online es la versión disponible online en la cual toda la gestión de la plataforma PaaS se lleva a cabo por Red Hat a través de Amazon Web Services. Lo único necesario para utilizar esta versión de la plataforma es una cuenta de usuario en la página web de OpenShift [2].

OpenShift permite interacutar con la plataforma y gestionar las aplicaciones desplegadas de diferentes maneras. Una de ellas es a través de las herramientas de cliente (*RHC Client command-line tools*) basadas en Ruby, que se instalan de forma local. Otras formas de interacutar con la plataforma es a través de la consola web en la página de OpenShift, o bien utilizando el plugin disponible para el entorno de desarrollo Eclipse.

Las aplicaciones desplegadas en la plataforma disponen de un contenedor con un conjunto de recursoso sobre el que se ejecutan, denominado *Gear*. Para poder hacer uso del contenedor es necesario añadir "cartucho" (*cartridge* en terminología de OpenShift). Son componentes o plugins utiliados para desplegar y ejecutar una aplicación. En un gear pueden ejecutarse varios cartuchos a la vez.

Cada gear tiene asociado un repositorio Git en el que se encuentra el código fuente de la aplicación. Actualmente hay disponibles tres tipos de gear en OpenShift Online: pequeño (512 MB de RAM), mediano (1 GB de RAM) y grande (2 GB de RAM). Por defecto los tres tipos disponen de 1 GB de espacio en disco. La cuenta gratuita de OpenShift dispone de un gear pequeño y 1 GB de espacio de almacenamiento para la aplicación.

En este trabajo, OpenShift se utiliza como plataforma para el hosting de la instancia de SonarQube y por lo tanto es una parte fundamental en ell desarrollo.

Se ha seleccionado OpenShift por su facilidad en la gestión de aplicacones y sobre todo, por ser una plataforma open source.

Git

Git[2] es una herramienta de control de versiones open source que se utiliza en conjunto con plataformas de repositorios como Bitbucket (sección 4.2).

Hay dos razones principales para utilizar Git durante el desarrollo del trabajo:

- Es relativamente fácil de utilizar para llevar el control de versiones del código fuente
- Se utiliza en conjunto con las herramientas de cliente RHC de OpenShift en la gestión de aplicaciones

Durate el desarrollo del trabajo se ha utilizado la versión 1.9.4 para sistema operativo Windows.

Bitbucket

Bitbucket[3] es una plataforma de repositorios de código utilizada en la gestión ágil de proyectos. Ofrece la posibilidad de crear repositorios, tanto públicos como privados, y es gratuito para pequeños equipos de desarrolladores de hasta cinco personas.

Dispone de herramientas como Jira (sección 4.2) para llevar la gestión de las incidencias y tareas del proyecto y es fácilmente utilizable a través de sistemas de control de versiones como Git.

Bitbucket se ha utilizado como gestor de proyecto durante el desarrollo del trabajo ya que se tenía previos conocimientos sobre la plataforma. Además, ofrece una buena gestión de las incidencias del proyecto a través de la integración con Jira.

Jira

Java

Java[4] es un lenguaje de programación orientado a objetos ampliamente utilizado en la industria del software. Su característica más importante es que se ejecuta sobre una máquina virtual independiente de la plataforma y por lo tanto puede ser utilizado tanto en sistemas Unix como Windows.

Java está basado en dos componentes:

[2]Página oficial de Git: https://git-scm.com/
[3]Página oficial de Bitbucket: https://bitbucket.org/
[4]Página oficial de Java: https://www.oracle.com/java/index.html

- **JRE (*Java Runtime Environment*):** es el entorno de ejecución o máquina virtual sobre la que se ejecutan las aplicaciones desarrolladas en Java. Se ha utilizado la versión 1.8.0-77 de JRE
- **JDK (*Java Development Kit*):** es el API que proporciona Java para poder desarrollar aplicaciones utilizando el lenguaje de programación. Se ha utilizado la versión 1.8.0-65 de JDK

En este trabajo, Java se ha utilizado en el desarrollo de un plugin para SonarQube, que se incluye en la plataforma de evaluación de trabajos fin de grado.

IntelliJ IDEA

IntelliJ IDEA[5] es un entorno de desarrollo para el lenguaje de programación Java, creado por JetBrains.

En el trabajo se ha utilizado la versión 15.0.2 de IntelliJ IDEA para el desarrollo de un plugin para SonarQube.

LaTeX

LaTeX [5] es un sistema utilizado en la creación y composición de documentos de alta calidad tipográfica. Realmente, LaTeX es un paquete de macros para el lenguaje de marcas TeX. LaTeX es ampliamente utilizado en la creación de documentos de carácter científico.

En el trabajo, LaTeX se ha utilizado para crear la documentación y los anexos, a través de la herramienta MiKTeX[6].

[5] Página oficial de IntelliJ IDEA: https://www.jetbrains.com/idea/
[6] Página oficial de MiKTeX: http://www.miktex.org/

Bibliografía

[1] G. Ann Campbell and Patroklos P. Papapetrou. *SonarQube in Action*. Manning, 2014.

[2] Red Hat. OpenShift Web Page. https://www.openshift.com.

[3] Steven Pousty and Katie J. Miller. *Getting Started with OpenShift*. O' Reilly, 2014.

[4] SonarQube. SonarQube Web Page. www.sonarqube.org.

[5] Wikipedia. Latex — wikipedia, la enciclopedia libre, 2015. [Internet; descargado 30-septiembre-2015].