



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 — 4 de abril 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, 4 de abril 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.

El proceso de medición de calidad de software se puede aplicar también a los Trabajos final de Grado del Grado de Ingeniería Informática. La información procedente de aplicar métricas de calidad proporciona un valor añadido al producto software de cada TFG y abre una nueva dimensión en el proceso de evaluación, en la cual el propio estudiante es quien guía y analiza un indicador de calidad durante el desarrollo de su trabajo. Desde el punto de vista pedagógico, se busca aplicar técnicas de aprendizaje invertido (*flipped learning*) en este tipo especial de asignaturas, donde el aprendizaje se basa en proyectos de distinta naturaleza.

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.

The software quality measurement process can also be applied to Final Year Projects in Computer Science Degree. The information from applying quality metrics adds value to the software product of each Final Year Project and opens a new dimension in the evaluation process, in which, the student guides and analyzes a quality indicator during the development of his/her project. From a pedagogical point of view, the process seeks to apply *flipped learning* techniques to this particular kind of subjects, where the learning process is based on different types of projects.

The main goal of this project 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

---

<b>Índice general</b>	<b>III</b>
<b>Índice de figuras</b>	<b>IV</b>
<b>Conceptos teóricos (Revisar con el tutor)</b>	<b>1</b>
3.1. Calidad y Software . . . . .	1
3.2. Gestión y Aseguramiento de la Calidad . . . . .	2
3.3. Estándares de calidad . . . . .	3
3.4. Métricas de calidad . . . . .	6
<b>Bibliografía</b>	<b>7</b>

---

# Índice de figuras

---

3.1. Desarrollo de software y gestión de calidad. . . . .	3
3.2. Esquema del estándar ISO 9001 . . . . .	4
3.3. Modelo de calidad para la Calidad Externa e Interna . . . . .	6
3.4. Modelo de calidad para la Calidad en Uso . . . . .	6



---

# Conceptos teóricos (Revisar con el tutor)

---

## 3.1. Calidad y Software

Hay muchas y diferentes formas de definir la calidad de un producto, por lo que a menudo resulta complicado dar una definición concreta. Por ejemplo, para un usuario, la satisfacción al utilizar un producto es sinónimo calidad. Si el producto satisface las necesidades del usuario, se dice que presenta calidad. Desde el punto de vista de un desarrollador, la calidad se define en términos de especificaciones del producto, es decir, si el producto final cumple las especificaciones inicialmente establecidas, presenta calidad.

A pesar de que se trata de un concepto subjetivo, existe una definición formalizada por el estándar ISO (*International Organization for Standardization*), que define la calidad como "el conjunto de propiedades y características de un producto o servicio que le confinen su aptitud para satisfacer unas necesidades expresadas o implícitas"<sup>2</sup>. Se asume además, que la calidad está estrechamente relacionada con las distintas etapas en el desarrollo de un producto. Por eso es importante la adopción de políticas de gestión y aseguramiento de la calidad durante el proceso de desarrollo.

La preocupación por la calidad de software surge en los años 60 del siglo pasado con el desarrollo de los primeros sistemas software de gran tamaño. El software que se creaba era lento, de baja fiabilidad, difícil de mantener y reutilizar. Esto originó la adopción de técnicas de gestión de la calidad de software, derivadas de metodologías usadas en la manufacturación industrial. En concreto, en la gestión de la calidad de software, se adoptaron normas del

---

<sup>1</sup>Página oficial de ISO: <http://www.iso.org/iso/home.html>

<sup>2</sup>Fuente de la definición: *Medición para la gestión en la Ingeniería del Software* [1], capítulo 12

estándar ISO 9000<sup>3</sup>, que posteriormente dieron lugar a otras normas específicas para evaluar la calidad de productos software, que se basaban en el proceso de desarrollo como medio de asegurar la calidad. Una de estas normas es el estándar ISO 9126<sup>4</sup>(sección 3.3), desglosado en cuatro partes, cada una de las cuales se ocupa de aspectos diferentes de la calidad.

En un sentido general, la calidad de software puede definirse como *”la aplicación de un proceso efectivo de desarrollo de software a partir del cual se obtenga un producto útil, que proporcione valores medibles para quienes lo producen y para quienes lo usen”*<sup>5</sup>.



## 3.2. Gestión y Aseguramiento de la Calidad

La gestión de la calidad de software se ocupa principalmente de dos cosas:

- A nivel de organización, la gestión de la calidad de software tiene como objetivo establecer un marco que conduzca al desarrollo de software de alta calidad. Esto implica la definición de los procesos de desarrollo a utilizar así como los estándares a aplicar a los distintos productos obtenidos (código fuente, documentación, requisitos, etc.).
- A nivel de proyecto, la gestión de la calidad involucra la aplicación de procesos de calidad específicos, comprobar que se han seguido dichos procesos y asegurar que los productos obtenidos se ajustan a los estándares aplicables a ese proyecto. La gestión de la calidad también se ocupa de definir un plan de calidad específico a seguir durante el desarrollo del proyecto.

El aseguramiento de la calidad (del inglés *Quality Assurance, QA*) es la definición de procesos y estándares que tienen como objetivo garantizar el desarrollo de software de calidad. También implica aplicar procesos de verificación y validación al producto obtenido. El aseguramiento de la calidad es utilizado en el proceso de gestión como herramienta para evaluar la adecuación del proceso de desarrollo a los planes de calidad establecidos.

---

<sup>3</sup>ISO 9000 en Wikipedia: [https://en.wikipedia.org/wiki/ISO\\_9000](https://en.wikipedia.org/wiki/ISO_9000)

<sup>4</sup>ISO 9126 en Wikipedia: [https://en.wikipedia.org/wiki/ISO/IEC\\_9126](https://en.wikipedia.org/wiki/ISO/IEC_9126)

<sup>5</sup>Definición adaptada de Pressman [6], capítulo 14, pág. 400

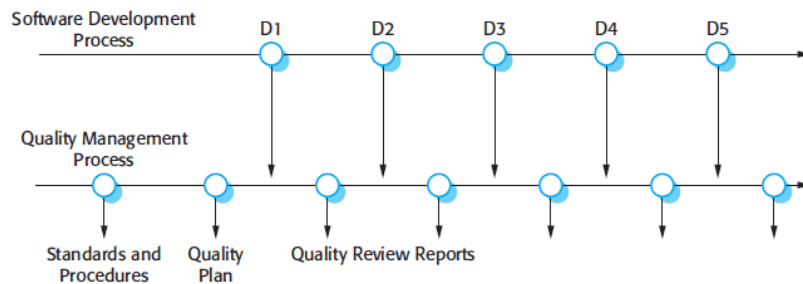



Figura 3.1: Desarrollo de software y gestión de calidad.

Normalmente el proceso de gestión de la calidad es independiente del proceso de desarrollo. La gestión de la calidad se encarga de comprobar los productos del proyecto para asegurar que son consistentes con los objetivos y los estándares establecidos. Es por ello, que el equipo de aseguramiento de calidad debería ser independiente del equipo de desarrollo, con el fin de evaluar el producto de forma objetiva. 

### 3.3. Estándares de calidad

Los estándares juegan un papel muy importante en la gestión de la calidad. Una de las tareas del aseguramiento de la calidad de software es seleccionar los estándares que van a ser aplicados tanto al proceso de desarrollo como al producto final. La importancia de los estándares se debe a que aportan conocimiento acerca de prácticas apropiadas a tener en cuenta durante el proceso de desarrollo y proporcionan un marco para poder definir lo que es la calidad en cada proyecto particular. Existen dos tipos de estándares utilizados en la gestión de la calidad de software:

- **Estándares de producto:** son estándares que se aplican al producto. Incluyen estándares sobre la documentación, estándares de código y estándares sobre la estructura de documentos.
- **Estándares de proceso:** son estándares que definen los procesos a seguir durante el desarrollo de software. En ellos se especifican prácticas habituales, utilizadas en el desarrollo.

## ISO 9001

ISO 9001<sup>6</sup> es un estándar del grupo de estándares ISO 9000<sup>7</sup>, aplicable en empresas que desarrollan y mantienen productos, incluyendo empresas del software. No es un estándar por sí solo, sino que proporciona un marco para el desarrollo de estándares de software. Define principios de calidad y procesos de calidad de forma general, que sirven de base para el desarrollo de guías de calidad.

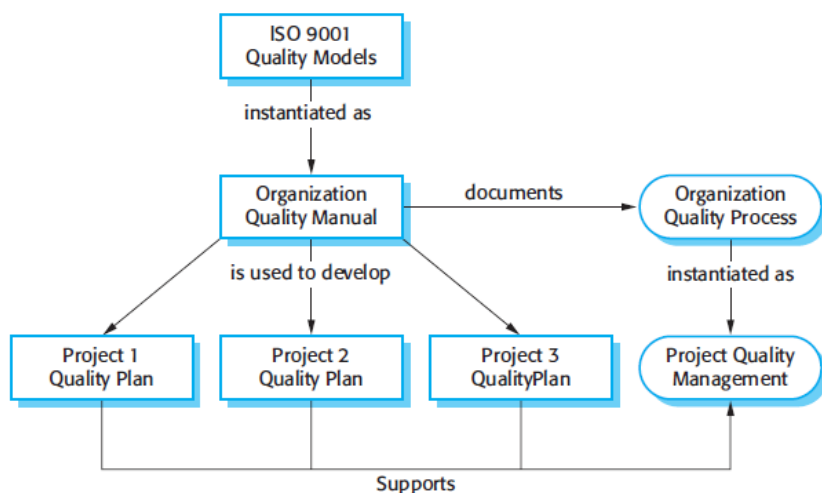


Figura 3.2: Esquema del estándar ISO 9001

## ISO 9126

ISO 9126<sup>8</sup> es un estándar utilizado en la evaluación de la calidad de software. El estándar está dividido en cuatro partes que tratan los conceptos de *modelo de calidad*[2], *métricas externas*[3], *métricas internas*[4] y *métricas de uso de calidad*[5], respectivamente.

En la primera parte del estándar se especifica un modelo de calidad de software de dos partes: *Modelo de calidad para la Calidad Externa e Interna* y *Modelo de calidad para Calidad en Uso*. En estos modelos de calidad se identifican características clave para la calidad del software, que se clasifican en una estructura de atributos y subatributos. El estándar identifica seis atributos principales:

<sup>6</sup>Página Web de ISO 9001: <http://www.iso9001.com/>

<sup>7</sup>Página Web de ISO 9000: [http://www.iso.org/iso/home/standards/management-standards/iso\\_9000.htm](http://www.iso.org/iso/home/standards/management-standards/iso_9000.htm)

<sup>8</sup>ISO 9126 en Wikipedia: [https://en.wikipedia.org/wiki/ISO/IEC\\_9126](https://en.wikipedia.org/wiki/ISO/IEC_9126)

- **Funcionalidad:** es la capacidad del producto software de proporcionar funcionalidades que satisfacen las necesidades explícitas e implícitas al utilizarlo bajo determinadas condiciones. Este atributo a su vez se divide en los siguientes subatributos: *Adecuación, Exactitud, Interoperabilidad, Seguridad y Cumplimiento funcional.*
- **Fiabilidad:** es la capacidad del producto software de mantener un nivel específico de prestaciones al utilizarlo bajo determinadas condiciones. La fiabilidad se divide en los siguientes subatributos: *Madurez, Tolerancia a fallos, Recuperabilidad y Cumplimiento de Fiabilidad.*
- **Usabilidad:** es la capacidad del producto software de ser aprendido, entendido y usado por el usuario. La usabilidad se divide en los siguientes subatributos: *Aprendizaje, Comprensión, Operatividad y Atractividad.*
- **Eficiencia:** es la capacidad del producto software de proporcionar la eficiencia apropiada, relativa a la cantidad de recursos utilizados, bajo condiciones específicas. La eficiencia a su vez se divide en los siguientes subatributos: *Exactitud, Comportamiento en el tiempo, Utilización de recursos y Cumplimiento de Eficiencia.*
- **Mantenibilidad:** es la capacidad de un producto software de ser mantenido o modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación de producto a un entorno concreto. La mantenibilidad se divide en los siguientes subapartados: *Facilidad de análisis, Facilidad de cambio, Escalabilidad, Facilidad de pruebas y Cumplimiento de la Mantenibilidad.*
- **Portabilidad:** es la capacidad del producto software de ser transferido de un entorno a otro. La portabilidad se divide en los siguientes subapartados: *Adaptabilidad, Instalabilidad, Coexistencia, Capacidad de reemplazamiento y Cumplimiento de la Portabilidad.*
- **Calidad en uso:** es la capacidad del producto software que permite a los usuarios conseguir sus objetivos de forma efectiva, productiva, segura y satisfactoria. La calidad de uso se divide en los siguientes subapartados: *Efectividad, Productividad, Seguridad y Satisfacción.*

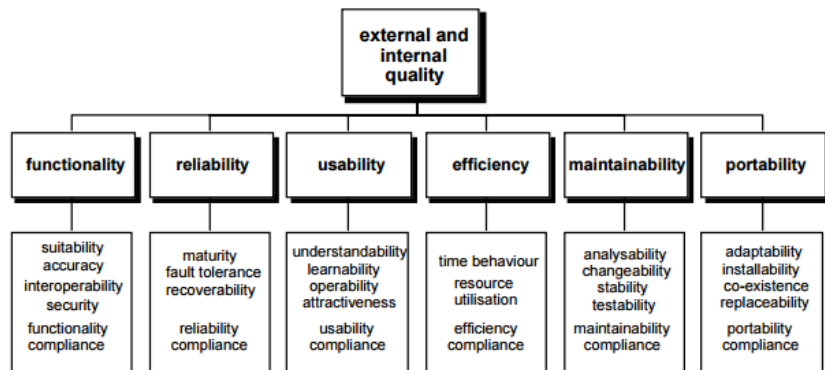


Figura 3.3: Modelo de calidad para la Calidad Externa e Interna



Figura 3.4: Modelo de calidad para la Calidad en Uso

### 3.4. Métricas de calidad

La forma más tradicional de asegurar la calidad es a través de métricas.

---

## Bibliografía

---

- [1] J.J. Fernández Dolado. *Medición para la Gestión en la Ingeniería del Software*. Ra-Ma, 2000.
- [2] ISO and IEC. *ISO/IEC 9126-1:2001, Software Engineering – Product Quality, Part 1: Quality Model*. 2001.
- [3] ISO and IEC. *ISO/IEC 9126-2:2003, Software Engineering – Product Quality, Part 2: External Metrics*. 2003.
- [4] ISO and IEC. *ISO/IEC 9126-3:2003, Software Engineering – Product Quality, Part 3: Internal Metrics*. 2003.
- [5] ISO and IEC. *ISO/IEC 9126-4:2004, Software Engineering – Product Quality, Part 4: Quality in use metrics*. 2004.
- [6] Roger S. Pressman. *Software Engineering: A Practitioner’s Approach*. McGraw-Hill, 2009.