

Audit Quality and Management of Information Systems - Appunti

Francesco Lorenzoni

Febrero 2025

Contents

I	Introduction to ACG	5
1	Calidad	9
1.1	¿Qué entendemos por “sistema de información”	9
1.2	Calidad	9
1.2.1	Modelos de Calidad	9
1.2.2	Métricas	9
1.3	Requisitos	10
1.4	Gestión de la Calidad	10
1.4.1	ISO 9001	11
1.4.1.1	Quality control and assurance	11
1.4.1.2	SPI - Software Process Improvement	11
1.4.2	Coste de la calidad	12

Part I

Introduction to ACG

Información General para la evaluación

Di lunedì seminarios fino a fine marzo... I seminari sono parte del 25% della observación Da marzo iniziano le prácticas, che sono un altro 25% 30% è una prova scritta 20% è una prova orale

Chapter 1

Calidad

1.1 ¿Qué entendemos por “sistema de información”

Definition 1.1 (Sistema de información) *Conjunto único de hardware, software, bases de datos, telecomunicaciones, personas y procedimientos configurado para recolectar, manipular, almacenar y procesar datos para convertirlos en información*

Entonces un sistema de información necesita de una **entrada** de datos, un **procesamiento** de los mismos y una **salida** de información.

1.2 Calidad

Definition 1.2 (Calidad - 1) *La calidad para Pressman (1998) es el cumplimiento con:*

- ◊ los requerimientos funcionales y de rendimiento explícitamente establecidos,
- ◊ los estándares de desarrollo explícitamente documentados
- ◊ con las características implícitas que se esperan de todo software desarrollado profesionalmente.

Definition 1.3 (Calidad - 2) *Según las standards ISO e IEEE la calidad es El grado con el que un sistema, componente o proceso cumple con los requisitos especificados y las necesidades o expectativas del cliente o usuario.*

Definition 1.4 (Calidad - 3) *Según la ISO 91260, la calidad es el conjunto de propiedades La totalidad de características de un producto de software que tienen como habilidad, satisfacer necesidades explícitas o implícitas.*

1.2.1 Modelos de Calidad

Los modelos de calidad son herramientas que permiten evaluar la calidad de un producto o servicio. Ellos apuntan a identificar características estándar relacionadas con la calidad del software a través de atributos de calidad. Atributos de calidad incluyen:

- ◊ Adecuación Funcional
- ◊ Seguridad
- ◊ Fiabilidad (reliability)
- ◊ Usabilidad
- ◊ Eficiencia
- ◊ Mantenibilidad
 - Reparabilidad
 - Adaptabilidad
 - Portabilidad

En relación con Mantenibilidad, el OPEN/CLOSED principle dice que un software debe estar abierto para extensión pero cerrado para modificación.

- ◊ Compatibilidad

Estos atributos pueden cambiar según los modelos

En general los atributos pueden ser externos o internos. Los primeros derivados de la relación entre el entorno y el sistema (para ello, el proceso o el sistema debe ejecutarse), e.g. reliability, robustness, usability. Los segundos derivados directamente de la descripción del producto o del proceso.

1.2.2 Métricas

Es necesario desarrollar métricas de calidad, que deben ser:

- ◊ Simples y fáciles de usar
- ◊ Empírica e intuitivas
- ◊ Consistente y objetivas

Por ejemplo, centrémonos en la mantenibilidad. Podemos medirla con las siguientes métricas:

- ◊ Acoplamiento
- ◊ Cohesión
- ◊ Complejidad Ciclomática de McCabe
- ◊ Código Chum
- ◊ Code Coverage
- ◊ Código Muerto
- ◊ Duplicación de Código
- ◊

1.3 Requisitos

Los requisitos son fundamentales en el software, y pueden ser utilizados para medir la calidad de este. Pero es importante notar que es necesario poder verificar si los requisitos están satisfechos con la implementación. Además, necesitamos también algunos controles sobre los requisitos, como complejidad, consistencia, completitud, corrección, claridad, verificabilidad, rastreabilidad, prioridad, viabilidad, flexibilidad, no ambigüedad, no redundancia, no contradicción, no vaguedad, no sobre-especificación, no sub-especificación.

Requisitos no funcionales son más fáciles de verificar que los funcionales, porque son más objetivos.

Definition 1.5 (Requisito no funcional verificable) *Una frase que incluye alguna medida que puede ser objetivamente probada.*

Matrices de trazabilidad son herramientas que permiten verificar la trazabilidad de los requisitos. Hay muchas matrices en las slides, algunas relacionan requisitos con casos de uso, otros con pruebas, otros con componentes del sistema.

Ejercicio 1 / Analisis Requisitos

Javier dice cosas correctas. Aparte de las cosas que mencionó, se puede ver cómo no hay muchos números en los requisitos; el documento dice que es necesario hacer backups, pero, ¿cuántos backups? ¿con qué frecuencia? ¿con qué software?.

1.4 Gestión de la Calidad

La calidad del proceso contribuye a la calidad del producto, y la calidad del producto contribuye a la calidad en uso.

- ◊ **Producto:** entregado al cliente
- ◊ **Proceso:** conjunto de actividades que se realizan para producir un producto

Hay algunos pasos importantes en la gestión de la calidad:

- ◊ Preparación y utilización de recursos
 - Humanos
 - Económicos
 - Infraestructura
 - Conocimientos y experiencia
- ◊ Procesos
 - Estratégicos
 - Operativos
 - Soporte
- ◊ Políticas de trabajo

Una política de trabajo de calidad es una manifestación que realiza una empresa acerca de cómo actúa y cuáles son las pautas o reglas que establecen en el día a día para trabajar que le lleven a mejorar continuamente y hacer las cosas bien a la primera.

Una parte crítica de la política de calidad es detectar y analizar los problemas en una empresa y los resultados no alcanzados
- ◊ Objetivos

“Conseguir una productividad del 100% de los desarrolladores” o “Aumentar la satisfacción de los cliente” son objetivos mal planteados.

1.4.1 ISO 9001

La ISO 9001 es una norma internacional que especifica los requisitos para un sistema de gestión de la calidad (SGC). Las organizaciones utilizan la norma para demostrar la capacidad para proporcionar productos y servicios que cumplen con los requisitos del cliente y los reglamentarios aplicables. El principal objetivo de la ISO 9001:2015 es lograr que una compañía consiga la satisfacción del cliente mediante el establecimiento de procesos de mejora continuada dentro de la misma.

Propone también un método de **mejora continua**, el PDCA (Plan, Do, Check, Act).

1.4.1.1 Quality control and assurance

- ◊ **Quality Control:** A part of quality management focused on fulfilling quality requirements
Parte de la gestión de calidad de software que comprueba que el proyecto sigue sus estándares, procesos y procedimientos, y que el proyecto produce los productos (entregables) internos y externos requeridos.
- ◊ **Quality Assurance:** A part of quality management focused on providing confidence that quality requirements will be fulfilled
Parte de la gestión de calidad de software que asegura que los estándares, procesos y procedimientos son apropiados para el proyecto y se han implementado correctamente.
- ◊ los *procesos* incluyen todas las actividades involucradas en el diseño, codificación, pruebas y mantenimiento;
- ◊ los *productos* incluyen software, datos asociados, documentación, y toda la documentación para soporte y reportes.

Los ingenieros del software realizan el trabajo técnico. En cambio, un grupo de SQA (Software Quality Assurance) se responsabiliza en la planificación de aseguramiento de la calidad, supervisión, mantenimiento de registros, análisis e informes.

El principal propósito del control de calidad es asegurar que el producto satisface los requisitos mediante **pruebas** y **revisiones** de los requisitos funcionales y no funcionales.

1.4.1.2 SPI - Software Process Improvement

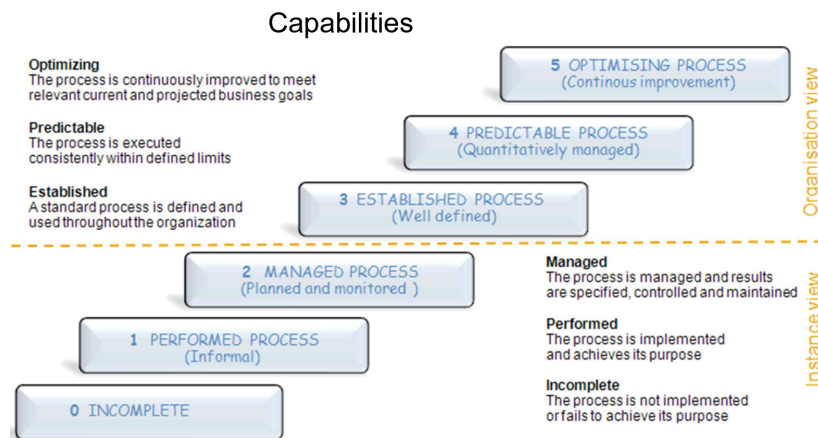


Figure 1.1: Spice Model for Process Improvement

Modelos de Madurez establecen marcos de referencia que evalúan la capacidad de una organización para desarrollar productos o servicios de calidad de manera consistente y predecible; van a evaluar la madurez de los procesos de desarrollo de software o sistemas.

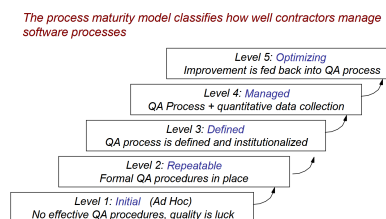


Figure 1.2: Capability Maturity CMMI

Complaints about Models

- ◇ No siempre se ven como tan relevantes y actuales por los ingenieros del software
- ◇ Pueden conllevar mucha burocracia
- ◇ Pueden requerir trabajo manual tedioso si no tienen herramientas software de soporte

Entonces, *to effectively apply standards, limit overhead!*

1.4.2 Coste de la calidad

$$\text{Coste de la calidad} = \text{Coste de conformidad} + \text{Coste de no conformidad} \quad (1.1)$$

Definition 1.6 (Coste de conformidad) *costes de las actividades para:*

- ◇ Valoración/Estimación: detección de defectos (testeo)
- ◇ Prevención: de errores (aseguramiento de calidad, etc)

Definition 1.7 (Coste de no conformidad) *costes de las actividades para:*

- ◇ Corrección: de defectos (reparación de errores)
- ◇ Fallos: en el producto (retrabajo, etc)
- ◇ Pérdida de clientes
- ◇ Costes de juicios
- ◇ etc.

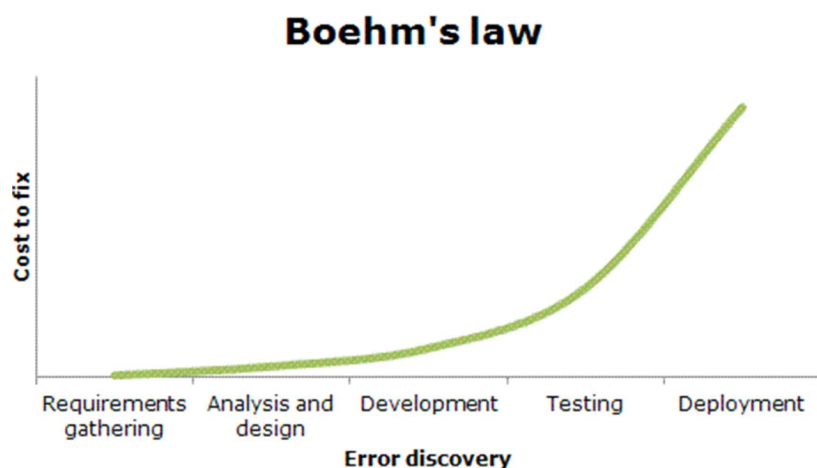


Figure 1.2: Según Boehm, el coste de corregir un error aumenta exponencialmente con el tiempo. Entonces, el **testing** es un momento crítico en el desarrollo de software, porque permite de encontrar errores antes de que el coste de corregirlos sea más alto.