

Facultad de Informática

Universidad Politécnica de Madrid

Estándar de Calidad

Ingeniería del Software

Grupo 5

Tabla de contenido

[1. Estructura del Documento 2](#_Toc355545028)

[1.1. Introducción 2](#_Toc355545029)

[1.2. Economía de la calidad software 2](#_Toc355545030)

[1.3. Control de la Calidad software 3](#_Toc355545031)

[1.4. Gestión de la calidad software 4](#_Toc355545032)

# Estructura del Documento

## Introducción

Para permitir obtener un nivel óptimo de calidad empleando en el desarrollo software controles de manera rápida, fácil y sencilla, para poder garantizar el cumplimiento de los requisitos estipulado y en sus fechas previstas. Se considera oportuno redactar este documento con la finalidad de mejorar la productividad, calidad y afianzar el control sobre las acciones realizar y desarrollar sobre el producto final.

## Economía de la calidad software

Hoy en día como en la mayoría de los negocios, actividades están presentes los productos software, es decir la industria del software está presente en diversos ámbitos, resulta por ello es necesario tener en cuesta costes, plazos y la calidad pues estos factores son críticos para un negocio.

De tal manera se ha de tener en cuenta los siguientes parámetros, para satisfacer las necesidades de costes relacionados con la calidad, pues no es rentable realizar un software con 0 errores, por el tiempo requerido para su desarrollo y más aún para superar pruebas de control, se perdería la noción de plazo.

Parámetros a tener en cuenta:

* Los proyectos grandes fallan y son más propensos a tener errores, por ello, se dividirán por sectores de control, encargándose de cada sector a un integrante del grupo, se procurara dividir el código por ramas funcionales, aun así se inspeccionará de manera general por todos.
* Los compromisos que se pactan en el inicio han de ser reales y no ficticios ni imposibles de conseguir, ello mejorara el tiempo y un correcto plazo de ejecución para obtener el producto final.
* Resulta importante tener un líder específico dentro del control de calidad, pues este motivara al equipo y se encargar de gestionar los subsectores creados para los controles de calidad.
* Cuantificar y cualificar el trabajo realizado, para poder luego hacer estimaciones sobre tiempos necesarios para la búsqueda de una mejor calidad.
* Verificar y analizar los defectos más relevantes que comprometan la estabilidad del sistema antes que los menos importantes o no cruciales que permitirían una ejecución del software medianamente funcional.

## Control de la Calidad software

En el contexto de los controles de calidad previsto para un proyecto software resultan de vital importancia las revisiones, así pues definiremos, la manera en que abordaremos dichas revisiones en nuestro proyecto.

Premisas a tener en cuenta para realizar las revisiones:

* Realizar pruebas en búsqueda de comportamientos anómalos, buscando casos limites o exagerados, pues el usuario no se ajustara en muchas veces a casos normales.
* Tener claro que la lógica a seguir será de manera estructurado, empezando a probar las funcionalidades elementales y cruciales para el sistema para luego ir puliendo hacia los demás aspectos.
* Cualificar y cuantificar el proceso de revisión, para el soporte fututo de estas y ampliaciones del software.
* Usar una lista de comprobación propia del proyecto común para todos los sectores designados anteriormente.
* Verificar lógica estipulada en la especificación de requisitos.

Estrategia de revisión y lista de comprobación tendrá los siguientes campos:

* Comprobación de que este completo el código a revisar antes de revisar
* Verificación de la estructura y flujo , es decir entender la lógica del código
* Comprobación de llamadas a funciones, método y procedimientos, es decir que el flujo del programa llegue a todos los sitios y no haya código explicito puesto que nunca se ejecute es decir obsoleto.
* Comprobación de variables. parámetros, tipo de paso por valor, referencia y el uso de ficheros o BBDD si es el caso y su correcta lectura y manipulación.

Estas premisas y estrategias se llevaran a cabo en la herramienta GITHub , pues ella provee de herramientas , disponibles para llegar a un consenso sobre que modificaciones son correctas , tras el primer escrutinio de un código funcional, además de esta manera todos los integrantes del proyecto podremos tener siempre la versión más reciente y no será un trabajo independiente sino uno en grupo.

## Gestión de la calidad software

La gestión de la calidad requiere que se hayan realizado medidas cuantitativas y cualitativas en dichas revisiones realizadas, tal es así que se describirá que medidas fueron obtenidas de la siguiente forma y como se han de tener en cuenta las medidas obtenidas de calidad para obtener una revisión eficiente y lograr una buena productividad y calidad.

Durante los procesos de revisión se han de buscar medir parámetros, tales como el Yield (rendimiento), velocidades de revisión, defectos encontrados por unidades de tamaño (Bloque de código funcional), defectos eliminados por hora ,coste de calidad(COQ).

Estas medidas resultaran útiles para tener un verdadero control sobre el proceso y la calidad del desarrollo software.

Así pues distribuiremos el proceso de medición entre los integrantes del grupo , cada uno se encargara de obtener una medida , esto se conseguirá recopilando información sobre los distintos procesos de revisión de cada uno en sus sector , es decir cada integrante de grupo cuando realiza revisiones del sector seccionado ya definido antes , enviara los datos de estas mediciones al correspondiente miembro del grupo encargado de esa medida.

De tal manera siempre será mejor prevenir y corregir los defectos antes que encontrarlos y corregirlos, pues los costes que representan los análisis en la búsqueda de defectos para su posterior corrección siempre seránmayores que el caso de la prevención, surge así la necesidad de tener muy en cuenta una estrategia de prevención.

Estrategia de Prevención de defectos:

* Dividir los defectos según sus prioridades, es decir: defectos con una determinada frecuencia de existencia, problemáticos, prevenibles e irritantes (no comprometidos con la estabilidad del sistema).
* Determinar cuáles son los defectos mas frecuentes iniciales encontrados, para poder empezar corrigiendo ellos
* Una vez determinados estos defectos más frecuentes y expliciticos , continuar estableciendo parámetros de búsqueda que deriven de estos,
* Apuntar estos parámetros de prevención y sus resultados en las listas de comprobación, pues servirá de apoyo para futuras inspecciones.

Como resumen en todo el proceso de inspección de calidad en sus diferentes fases , podemos discurrir que es mejor prevenir que buscar defectos , pero como esto no es posible debemos estructurar los diferentes niveles de calidades dentro del proyecto en sistemas fáciles para su futura revisión a fin de valorar su calidad y realizar medidas cualitativas y cuantitativas que comprometan la calidad , pues nos apoyaremos en estas para predecir el tiempo y coste empleado en solucionar los defectos buscados .