

Capítulo

1

Introducción

Centro Asociado Palma de Mallorca

Tutor: Antonio Rivero Cuesta

1 Introducción

1.1 Introducción.

El software es una parte principal del entorno humano. Actualmente sería complicado imaginarse la vida sin toda la tecnología que utilizamos a diario.

El software está presente no sólo en los sistemas informáticos que realizan tareas de tratamiento de información sino en un sinnúmero de sistemas de la más diversa complejidad.

La ingeniería es según la Real Academia de la Lengua el conjunto de conocimientos y técnicas que permiten aplicar el saber científico a la utilización de la materia y de las fuentes de energía.

En esta asignatura se introduce el conjunto de técnicas y procedimientos que se han ido desarrollando a lo largo de las últimas décadas, para poder elaborar de una forma ordenada y eficiente tantas y tantas líneas de código que componen el software.

Todas estas técnicas y procedimientos componen la ingeniería de software.

1.2 Objetivos

En este capítulo se da una visión inicial de lo que es el software y cómo se produce. Igualmente se profundiza en el concepto de ingeniería del software. Se pretende que el lector adquiera una idea clara de los siguientes conceptos:

- Definición de software y requisitos de calidad exigibles.
- No todo el software es igual, según su aplicación y modo de desarrollarlo puede ser muy diverso. Se da una visión de los distintos tipos de software.
- Definición del concepto de Ingeniería del software y comprensión de su origen.
- Existen algunas ideas preestablecidas acerca del software que se analizan.

1.3 ¿Qué es el software?

El software incluye:

- Los programas que gobiernan el funcionamiento del sistema.
- Documentos.
- Bases de datos.
- Los procedimientos de operación o de mantenimiento periódicos.

El software puede ser:

- Un producto que se venda.
- Sólo una parte.

La elaboración del software ocupa a millones de personas en todo el mundo y se puede considerar una actividad económica en sí misma.

1.3.1 Calidad del software

La calidad de un producto puede valorarse desde puntos de vista diversos.

Existe un esquema general de mediciones de la calidad de software propuesto por MacCall y otros [McCall78], basado en valoraciones a tres niveles diferentes

- Factores.
- Criterios.
- Métricas.

Los factores de calidad constituyen el nivel superior, y son la valoración propiamente de la calidad. Esta valoración no se hace directamente, sino en función de ciertos criterios o aspectos de nivel intermedio que influyen en los factores de calidad.

Las métricas están en el nivel inferior, son mediciones puntuales de determinados atributos o características del producto, y son la base para evaluar los criterios intermedios.

Entre los factores de calidad propuestos encontramos los siguientes:

- **CORRECCIÓN.** Es el grado en que un producto software cumple con sus especificaciones. Podría estimarse como el porcentaje de requisitos que se cumplen adecuadamente.
- **FIABILIDAD.** Es el grado de ausencia de fallos durante la operación del producto software. Puede estimarse como el número de fallos producidos o el tiempo durante el que permanece inutilizable durante un intervalo de operación dado.
- **EFICIENCIA.** Es la relación entre la cantidad de resultados suministrados y los recursos requeridos durante la operación. Se mediría como la inversa de los recursos consumidos para realizar una operación dada.
- **SEGURIDAD.** Es la dificultad para el acceso a los datos o a los datos o a las operaciones por parte de personal no autorizado.
- **FACILIDAD DE USO.** Es la inversa del esfuerzo requerido para aprender a usar un producto software y utilizarlo adecuadamente.
- **MANTENIBILIDAD.** Es la facilidad para corregir el producto en caso necesario. Se aplica propiamente el mantenimiento correctivo.
- **FLEXIBILIDAD.** Es la facilidad para modificar el producto software. Se aplica propiamente al mantenimiento adaptativo y al perfectivo.
- **FACILIDAD DE PRUEBA.** Es la inversa del esfuerzo requerido para ensayar un producto software y comprobar su corrección o fiabilidad.
- **TRANSPORTABILIDAD.** Es la facilidad para adaptar el producto software a una plataforma (hardware + sistema operativo) diferente de aquella para la que se desarrolló inicialmente.

- **REUSABILIDAD.** Es la facilidad para emplear partes del producto en otros desarrollos posteriores. Se facilita mediante una adecuada organización de los módulos y funciones durante el diseño.
- **INTEROPERATIVIDAD.** Es la facilidad o capacidad del producto software para trabajar en combinación con otros productos.

Estos factores de calidad se centran en características del producto software.

El proceso de desarrollo influye muy directamente en la calidad del producto obtenido.

Comprobar la calidad de un software es una tarea compleja. Las pruebas o ensayos consisten en hacer un producto software o una parte de él en condiciones determinadas, y comprobar si los resultados son correctos. El objetivo de las pruebas es descubrir los errores que pueda contener el software ensayado.

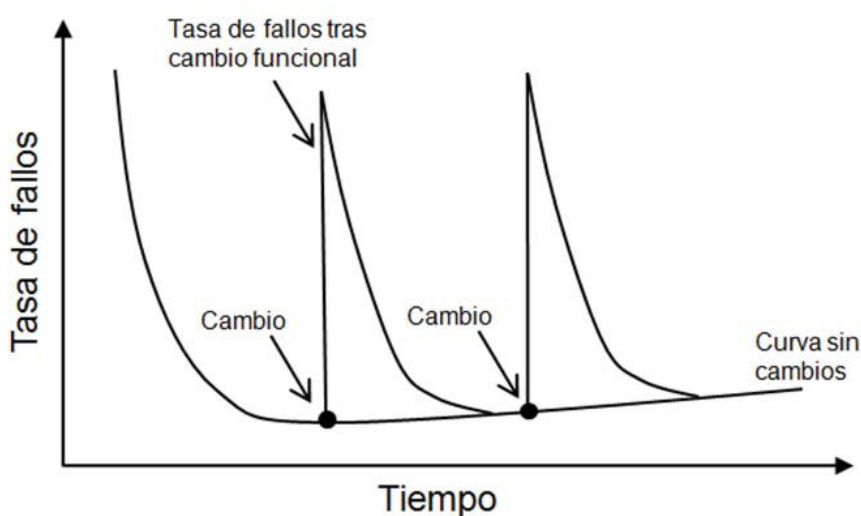
Las pruebas no permiten garantizar la calidad de un producto. Puede decirse que una prueba tiene éxito si se descubre algún error, con lo que se sabe que el producto no cumple con algún criterio de calidad. Por el contrario, si la prueba no descubre ningún error, no se garantiza con ello la calidad del producto, ya que pueden existir otros errores que habrían de descubrirse mediante pruebas diferentes.

En la figura 1.1 podemos observar de forma simplificada la evolución de la tasa de fallos de un software en el tiempo.

Inicialmente esta tasa es muy alta. Según se van corrigiendo los errores se reduce rápidamente.

Sin embargo, a lo largo de la vida de un software es frecuente realizar mejoras funcionales, dando lugar a distintas versiones.

Cada vez que introducimos cambios en las nuevas versiones, el número de errores del software se dispara, haciendo de nuevo necesario la corrección de los mismos.



1.3.2 Tipos de software

Clasificar el software no es una tarea fácil debido a la gran variedad de aplicaciones y métodos de desarrollo que existe. Una de las clasificaciones más completas agrupa el software en siete grandes categorías:

- Software de Sistemas

Lo forman todos aquellos programas necesarios para dar soporte a otros programas, como los sistemas operativos, los compiladores o los programas de gestión de redes.

Su principal característica es su alto grado de interacción con el hardware, ya que en muchos casos deben gestionar de forma eficiente el acceso al hardware por parte de otros programas o usuarios.

- Software de Aplicación

Son aplicaciones desarrolladas para resolver problemas específicos de los negocios.

En esta categoría incluiríamos el software de gestión de los bancos o de las grandes empresas en general.

- Software de Ingeniería y Ciencias

El objetivo es la programación de elaborados algoritmos matemáticos para modelar y simular complejos sistemas o procesos, tales como reacciones nucleares, modelos meteorológicos, la red eléctrica de un país o el diseño de un avión.

- Software Incrustado

Reside en el interior de un producto o sistema, y su objetivo es controlarlo, definir su comportamiento.

Suele ser muy específico y de pequeñas dimensiones, con la necesidad de operar en tiempo real.

Desde el regulador de temperatura de una vivienda hasta el sistema de frenos de un vehículo, están gobernados por este tipo de software.

- Software de Línea de Producto

Su objetivo es dar una determinada funcionalidad al consumidor.

En esta categoría encontramos procesadores de texto, hojas de cálculo o las aplicaciones de contabilidad para pequeñas empresas.

- Aplicaciones Web, WebApps

En los últimos años se ha extendido su utilización con la generalización de los aparatos móviles con acceso a redes.

Inicialmente simplemente se componían de archivos de hipertexto para la presentación de información, sin embargo hoy día tienen capacidad de cómputo y están integradas con aplicaciones y bases de datos corporativas.

A través de ellas se puede operar una cuenta bancaria, realizar todo tipo de compras, utilizar juegos muy elaborados ó conocer el tiempo en cualquier parte del mundo. La comodidad, rapidez y vistosidad son determinantes a la hora de que tengan éxito.

- Software de Inteligencia Artificial

Incluye aplicaciones de robótica, visión artificial, redes neuronales o sobre la teoría de juegos.

Utilizan algoritmos no numéricos para la resolución de los problemas, como por ejemplo árboles lógicos de búsqueda.

1.4 ¿Cómo se fabrica el software?

En los principios de la informática la labor de desarrollo de software se planteaba como una actividad artesanal basada en la labor de personas habilidosas y más o menos creativas que actuaban en forma individual y relativamente poco disciplinada.

Al aumentar la capacidad de los computadores gracias a los avances del hardware aumentó también la complejidad de las aplicaciones a programar y se apreció la necesidad de una mejor organización de la producción de software basada en el trabajo en equipo con la consiguiente división y organización del trabajo y el empleo de herramientas apropiadas que automaticen las labores triviales y repetitivas.

La identificación formal del problema origina una frenética actividad para la creación de metodologías concretas de desarrollo y en general en la concepción de la ingeniería del software como disciplina.

A finales de los años 60 se acuña el termino Ingeniería del Software en un congreso de la OTAN de manera formal.

Con esta denominación se designa el empleo en el desarrollo del software de técnicas y procedimientos típicos de la ingeniería en general.

El software tiene una particularidad especial frente a cualquier producto físico que podamos imaginar: una vez diseñado este se puede replicar con tremenda facilidad sin necesidad de un proceso de fabricación propiamente dicho.

A pesar de ello la ingeniería del software se ha desarrollado a partir de la ingeniería industrial.

La ingeniería del software amplía la visión del desarrollo del software como una actividad esencialmente de programación, contemplando además otras actividades de análisis y diseño previos, y de integración y verificación posteriores.

La distribución de todas estas actividades a lo largo del tiempo constituye lo se ha dado en llamar ciclo de vida del desarrollo de software.

Ingeniería de software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software, y el estudio de estos enfoques, es decir, *la aplicación de la ingeniería al software*.

A lo largo de los años 70 aparecen las herramientas CASE (Computer Aided Software Engineering) de diseño asistido por ordenador, que se aplican ampliamente durante los 80.

Las herramientas CASE tradicionales apoyaban a las actividades anteriores a la programación o codificación, para la que se seguías empleando herramientas tradicionales, como los compiladores, que funcionaban de forma totalmente separada de la herramienta CASE.

En los 90 se amplía el campo de aplicación de las herramientas de ingeniería de software, reconociendo la necesidad de automatizar aún más la labor de desarrollo mediante la formalización del proceso completo de producción de software y el empleo de herramientas que soporten todo el ciclo de vida de desarrollo.

Estas herramientas se designan con las siglas IPSE (Integrated Project Support Environment) o, más recientemente, ICASE (Integrated CASE).

Con la llegada del siglo XXI, el desarrollo e implantación de Internet, muchos de los planteamientos de las herramientas CASE han evolucionado.

Por un lado la posibilidad de distribuir el desarrollo de los proyectos en diferentes localizaciones y por otro la obligada necesidad de la utilización de esta plataforma como marco para el funcionamiento de la mayoría de las aplicaciones.

Otro de los grandes retos que ha aparecido y triunfado recientemente son los "smartphones" o teléfonos inteligentes.

Por el momento sólo se utilizan como plataformas de funcionamiento, pero no hay duda de que en breve formarán parte de las infraestructuras empleadas para el desarrollo del software.

A lo largo de estos períodos de tiempo fue surgiendo una importante comunidad científica en torno a la ingeniería de software.

Dos organizaciones han liderado dicha comunidad, ACM e IEEE Computer Society, que han promovido activamente la puesta en práctica de esta disciplina.

IEEE ha desarrollado la guía SWEBOK con el objeto de crear una acreditación para la profesión de ingeniero del software en Estados Unidos.

Dicha guía da forma a los conocimientos necesarios para dominar esta disciplina y los diferenciales frente a otras relacionadas con el software, como las ciencias de la computación o la gestión de proyectos de software.

1.5 Mitos falsos sobre el software

- El hardware es mucho más importante que el software.
- El software es fácil de desarrollar.
- El software consiste exclusivamente en programas ejecutables.
- El desarrollo del software es solo una labor de programación.
- Es natural que el software contenga errores.

1.6 Conclusiones

El software lo componen el conjunto de programas que gobiernan el comportamiento de cualquier sistema basado en computador.

En muchos casos el software tiene entidad en sí misma y puede ser considerado un producto propiamente dicho.

La aplicación del conocimiento y del método científico a la elaboración del software dan lugar a la disciplina que se conoce como *ingeniería del software*

El ciclo de vida del software igual que el de cualquier otro producto que se pueda elaborar es la evolución del mismo desde el momento que se concibe hasta que se deja de utilizar

Contenido

1	Introducción.....	2
1.1	Introducción.....	2
1.2	Objetivos	2
1.3	¿Qué es el software?.....	2
1.3.1	Calidad del software.....	3
1.3.2	Tipos de software.....	5
1.4	¿Cómo se fabrica el software?.....	6
1.5	Mitos falsos sobre el software	7
1.6	Conclusiones.....	7