

Introducción a la Ingeniería del Software

Daniel Monjas Miguélez

11 de marzo de 2021

Índice

1. El producto Software	3
1.1. Naturaleza del software	3
1.2. Definición de software	3
1.3. Características del software.	3
1.4. Tipos y dominios de aplicación del software	4
1.5. Proceso de producción	7
1.6. Efuerzo requerido por etapas	7
1.7. Problemas	7
2. El concepto de ingeniería del software	9
2.1. Definición de Ingeniería del software	9
2.2. Terminología usada en ingeniería del software	9
2.3. Resumen	10

1. El producto Software

1.1. Naturaleza del software

La naturaleza del software es dual (como la dualidad onda-partícula en física). Dependiendo de la situación la naturaleza se puede tratar como:

- Producto.
 - Proporciona potencial de cómputo.
 - Es un transformador de información.
- Vehículo, para distribuir un producto.
 - Actúa como base para el control de la computadora. Sistemas Operativos.
 - Actúa como base para la comunicación de información. Redes.
 - Actúa como base para la creación y control de otros programas. Herramientas y ambientes de software.

El software distribuye el producto más importante de nuestro tiempo, **la información**.

1.2. Definición de software

Cuando se trata al software como un programa de computadora hablamos de una definición incompleta. El software es:

1. Instrucciones (programas) que cuando se ejecutan proporcionan las funciones características buscadas.
2. Estructuras de datos que permiten a los programas manipular la información adecuadamente.
3. Información en papel o en forma virtual (documentación) que describe la operación y uso de los programas.

1.3. Características del software.

- El software se desarrolla o modifica con intelecto; no se manufactura en el sentido clásico. Aunque hay similitudes entre el desarrollo de software y la fabricación de hardware, las dos actividades son diferentes en lo fundamental. En ambas, la alta calidad se logra a través de un buen diseño, pero la fase de manufactura del hardware introduce problemas de calidad que no existe en el software. Ambas actividades dependen de personas, pero la relación entre los individuos dedicados y el trabajo logrado es diferente por completo. Las dos actividades requieren la construcción de un “producto”, pero los enfoques son distintos. Los costos del software se concentran en la ingeniería.

- El software no se “desgasta”, pero se deteriora. En la figura 1.3 se ve que el hardware presenta una tasa de fallas elevada en una etapa temprana de su vida (atribuible a defectos de diseño o manufactura); los defectos se corrigen y la tasa de fallas se abate a un nivel estable durante cierto tiempo. Conforme pasa el tiempo, la tasa de fallos aumenta de nuevo a medida que los componentes hardware resienten los efectos acumulativos de suciedad, vibración, abuso, temperaturas extremas y muchos otros inconvenientes ambientales.

Por su parte el software no es susceptible a los problemas ambientales que hacen que el hardware se “desgaste”. Por tanto, en teoría, la curva de tasa de fallas adopta la forma de la “curva idealizada” que se aprecia en la figura 1.3. Los defectos ocultos ocasionarán también tasas elevadas de fallas al comienzo de la vida de un programa. Sin embargo, éstas se corrigen y la curva se aplanan, como se indica. Sin embargo es claro que el software se deteriora. Durante su vida, este sufrirá cambios. Es probable que cuando éstos se realicen, se introduzcan errores que ocasionen que la curva de tasa de fallas tenga aumentos súbitos, como se ilustra en la “curva real” (figura 1.3). Antes de que la curva vuelva a su tasa de fallas original de estado estable, surge la solicitud de otro cambio que hace que la curva se dispare otra vez. Poco a poco, el mínimo de la tasa de fallas comienza a aumentar, es decir, el software se deteriora como consecuencia del cambio.

- La mayor parte del software se construye para uso individualizado.

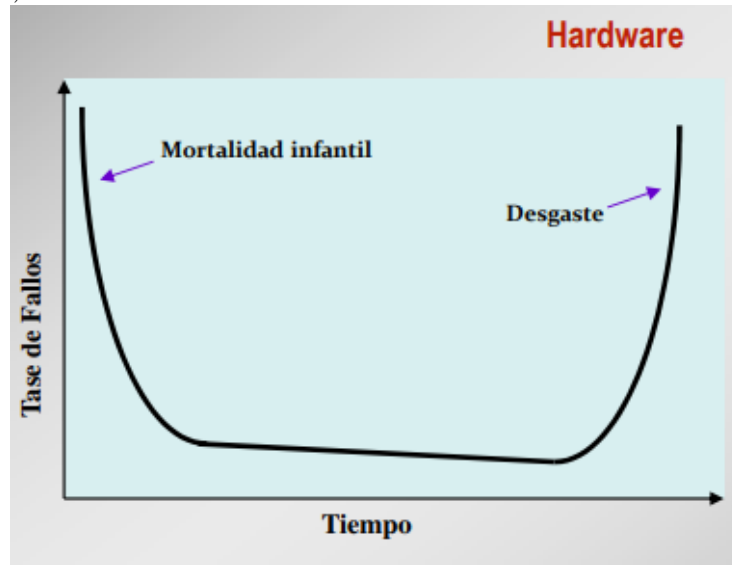
1.4. Tipos y dominios de aplicación del software

- Software genérico. Sistema autónomo producido por una organización de desarrollo y vendido en el mercado abierto a cualquier cliente que pueda comprarlo.
- Software a medida. Sistema desarrollado por una empresa especialmente para un cliente particular.

Dominios de aplicación del software. Actualmente, hay siete grandes categorías de software de computadora que plantean retos continuos a los ingenieros de software:

- Software de sistemas. Conjuntos de programas que proporcionan servicio a otros programas. Determinado software de sistemas (por ejemplo, compiladores, editores ,etc.) procesa estructuras de información complejas pero deterministas. Otras aplicaciones de sistemas (por ejemplo, componentes de sistemas operativos, manejadores, software de redes, procesadores de telecomunicaciones) procesan sobre todo datos indeterminados. En cualquier caso, el área de software de sistemas se caracteriza por: gran interacción

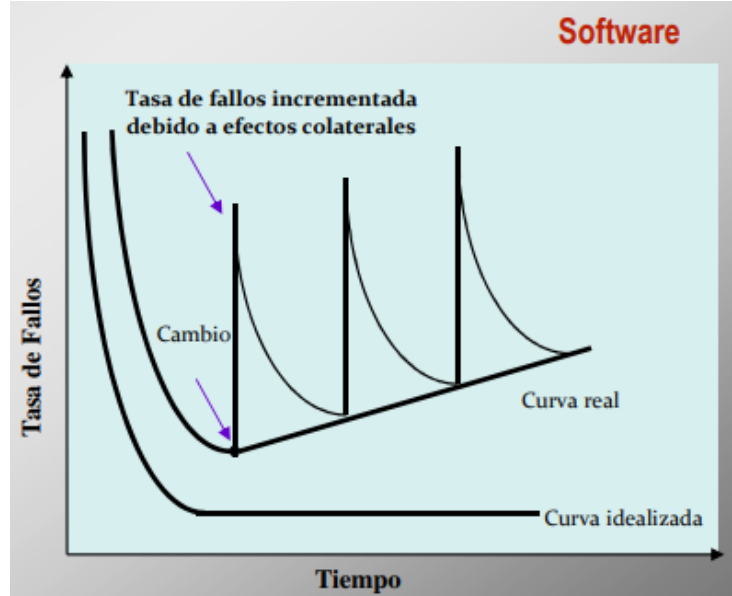
Figura 1: Evolución de la tasa de fallos debidos al hardware en el tiempo (‘‘curva de tina’’)



con el hardware de la computadora, uso intensivo por parte de usuarios múltiples, operación concurrente que requiere la secuenciación, recursos compartidos y administración de un proceso sofisticado, estructuras complejas de datos e interfaces externas múltiples.

- Software de aplicación. Programas que resuelven una necesidad específica de negocios. Las aplicaciones en esta área procesan datos comerciales o técnicos en una forma que facilita las operaciones de negocios o la toma de decisiones administrativas o técnicas. Además de las aplicaciones convencionales de procesamiento de datos, el software de aplicación se usa para controlar funciones de negocios en tiempo real (por ejemplo, procesamiento de transacciones en punto de venta).
- Software de ingeniería y ciencias. Implementa algoritmos “devoradores” de números. Las aplicaciones van de la astronomía a la vulcanoogía, del análisis de tensiones en automóviles a la dinámica orbital del transbordador espacial, y de la biología molecular a la manufactura automatizada. Sin embargo, las aplicaciones modernas dentro del área de la ingeniería y las ciencias están abandonando los algoritmos numéricos convencionales. El diseño asistido por computadora, la simulación de sistemas y otras aplicaciones interactivas, han comenzado a hacerse en tiempo real e incluso han tomado características del software de sistemas.
- Software empotrado. Reside dentro de un producto o sistema e implementa y controla características y funciones para el usuario final y para el sistema

Figura 2: Evolución de la tasa de fallos debidos al software en el tiempo



en sí. Ejecuta funciones limitadas y particulares (por ejemplo, control del tablero de un horno de microondas) o provee una capacidad significativa de funcionamiento y control (funciones digitales en un automovil, como el control del combustible, el tablero de control y de los sistemas de frenado).

- Software de gestión. Proporciona una capacidad específica para uso de mucho consumidores diferentes. Se centra en algún mercado limitado y particular o se dirige a mercados masivos de consumidores.
- Aplicaciones Web. llamadas “webapps”, esta categoría de software centrado en redes agrupa una amplia gama de aplicaciones. En su forma más sencilla, las webapps son poco más que un conjunto de archivos de hipertexto vinculados que presentan información con uso de texto y gráficas limitadas. Sin embargo, desde que surgió Web 2.0, las webapps están evolucionando hacia ambientes de cómputo sofisticados que no sólo proveen características aisladas, funciones de cómputo y contenido para el usuario final, sino que también están integradas con bases de datos corporativas y aplicaciones de negocios.
- Software de inteligencia artificial. Implementa algoritmos no numéricos para resolver problemas complejos difíciles de tratar computacionalmente o con análisis directo. Las aplicaciones en este área incluyen la robótica, sistemas expertos, reconocimiento de patrones, redes neurales artificiales, demostración de teoremas y juegos.

1.5. Proceso de producción

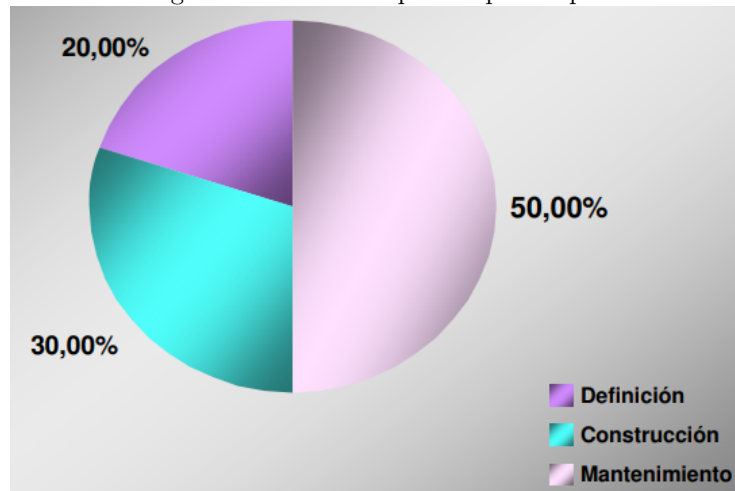
Figura 3: Proceso de producción del software



- Definición. Se debe realizar ingeniería de sistemas, ingeniería de requisitos y planificación de proyectos.
- Construcción. Se debe realizar diseño del software, generación del código y prueba del software.
- Evolución. Se debe realizar corrección, adaptación, mejor y prevención.

1.6. Esfuerzo requerido por etapas

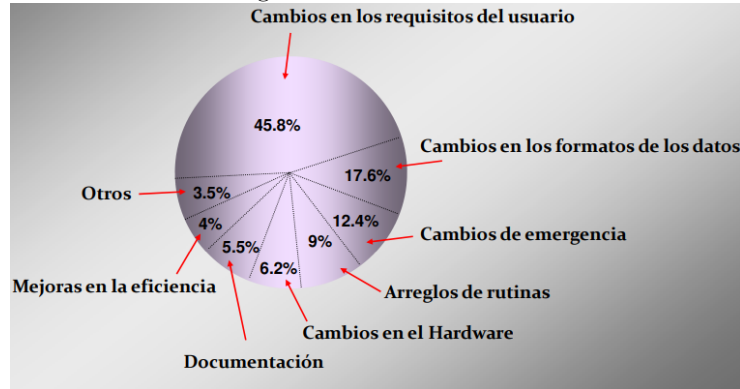
Figura 4: Esfuerzo requerido por etapas



1.7. Problemas

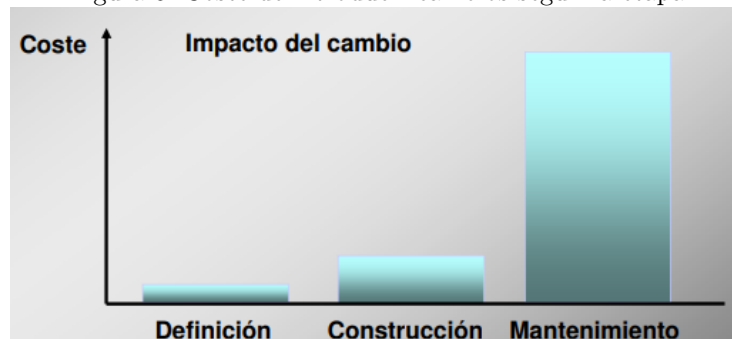
En el desarrollo del software pueden ocurrir varios tipos de problemas distintos.

Figura 5: Mantenimiento



- Comunicación entre personas, bien sea entre los desarrolladores y los clientes o directamente entre los desarrolladores.
- Incumplimiento de la planificación. Normalmente los plazos de entrega son decididos por directivos que realmente no conocen el tiempo requerido en el desarrollo de un proyecto. Cuando se acerca la fecha límite y ante el miedo de perder a un cliente importante es común contratar más personal con el fin de terminar el proyecto a tiempo, esta práctica se conoce como “horda mongoliana”, y lo normal es que en vez de ayudar perjudica. Esto se debe a que más medios son más difíciles de gestionar y al incorporar una persona nueva al equipo de desarrollo tendrá que aprender como se trabaja en la empresa, ponerse al día con los procedimientos, familiarizarse con el proyectos, etc. Finalmente la solución solamente hace más grande el problema.
- Incorporar cambios en etapas avanzadas del proceso. Cuanto más se haya avanzado en el proyecto mayor será el coste de introducir cambios en el mismo.

Figura 6: Coste de introducir cambios según la etapa



2. El concepto de ingeniería del software

La ingeniería del software se propuso por primera vez en 1968 en una conferencia en la que se habló sobre ‘a crisis del software’. Se volvió claro que las aproximaciones individuales al desarrollo de programas no escalarían hasta sistemas software grandes y complejos. Estas no eran confiables, costaban más de lo esperado, se entregaban con retraso, eran difíciles de mantener y muy complejas y el software era de pobre ejecución.

Se concluyó que se debía entender el problema antes de desarrollar una aplicación. El diseño pasaba de ser una actividad crucial, ya que el software debía tener alta calidad y ser fácil de mantener. Para conseguir esto se debía utilizar la ingeniería del software.

Durante los años setenta y ochenta, una variedad de nuevos métodos y técnicas de ingeniería del software fueron desarrollados, tales como la programación estructurada, la ocultación de información y el desarrollo orientado a objetos. Se desarrollaron herramientas y notaciones estándar que actualmente son ampliamente usadas.

2.1. Definición de Ingeniería del software

Definiciones:

- “Establecimiento y uso de principios fundamentales de la ingeniería con objeto de desarrollar en forma económica software que sea fiable y que trabaje en máquinas reales” (Fritz Bauer, 1972).
- “Aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada y requerida para el desarrollo, operación y mantenimiento del programa” (B. Boehm, 1976).
- “Aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; es decir, aplicación de la ingeniería del software” (estándar - IEEE, 1993).

2.2. Terminología usada en ingeniería del software

- Sistema. Conjunto de elementos relacionados entre sí y con el medio, que forman una unidad o un todo organizativo.
- Sistema basado en computadora. Conjunto o disposición de elementos organizados para cumplir una meta predefinida al procesar información.
- Sistema software. Conjunto de piezas o elementos software relacionados entre sí y organizados en subsistemas.

Figura 7: Esquema de sistema basado en computadora

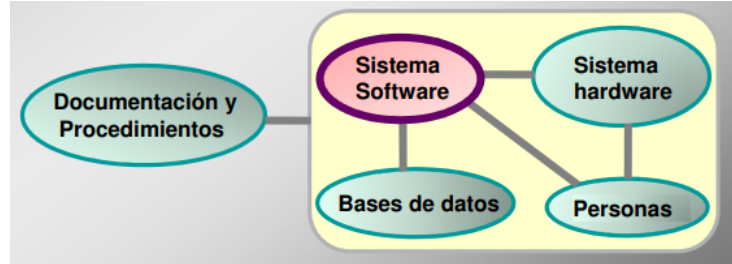
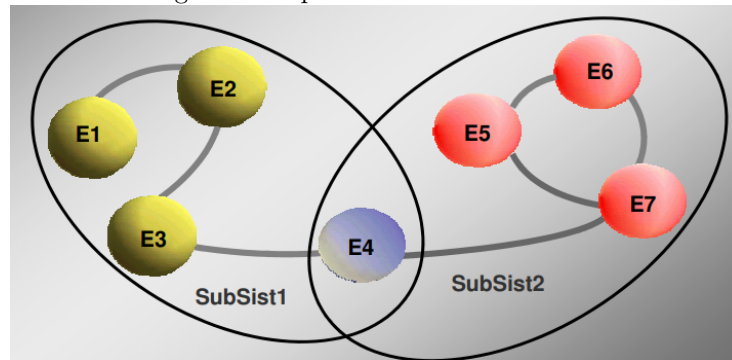


Figura 8: Esquema de sistema software

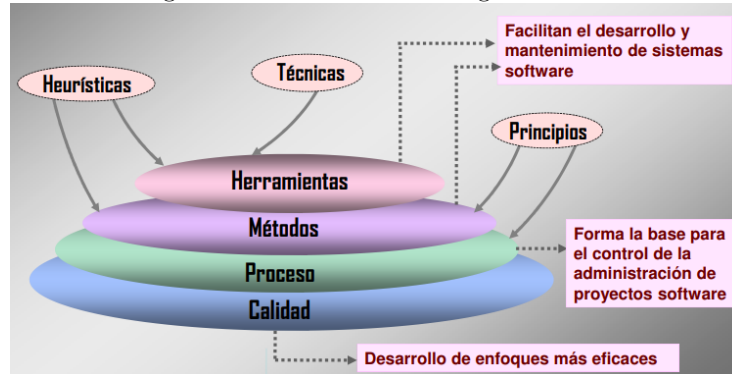


- Modelos. Representación de un sistema en un determinado lenguaje: de un mismo sistema se pueden construir muchos modelos.
- Principios. Elementos adquiridos mediante el conocimiento, que definen las características que debe poseer un modelo para ser una representación adecuada de un sistema.
- Herramientas. Instrumentos que permiten la representación de modelos.
- Técnicas. Modo de utilización de las herramientas.
- Heurísticas. Conjunto de reglas empíricas, que al ser aplicadas producen modelos que se adecuan a los principios.
- Métodos. Secuencia de actividades para la obtención de un producto (modelo), que describen cómo usar las herramientas y las heurísticas.

2.3. Resumen

- Modelos de ciclo de vida del software:
 - El modelo en cascada.

Figura 9: Resumen terminología software



- Desarrollo incremental.
- Ingeniería del software orientada a la reutilización

■ Modelos orientados al cambio

- Prototipo del sistema.
 - Entrega incremental.
 - Proceso unificado.
 - Métodos ágiles.
- Proceso de desarrollo o ciclo de vida del software. Estrategia que define la división y ubicación temporal de las etapas (o actividades) que se realizan durante el desarrollo del software.
 - Etapas. Se caracterizan por las tareas que se realizan por el producto (documento, artefacto, ...) que se obtiene.
 - Modelos (o paradigmas) de ciclo de vida del software. Son representaciones abstractas (simplificadas) del proceso de desarrollo del software.

Existen diversos modelos según las etapas que se consideren, posición relativa de las mismas y las tareas a realizar en ellas.

Referencias

- [1] . Sommerville. Software Engineering. Addison Wesley, 2011, ISBN 13: 978-0-13-703515-1
- [2] . Pressman. Ingeniería del Software. McGraw Hill, 2010, ISBN: 978-607-15-0314-5