

1. 1. Ingeniería de Software / Revisión /Ingeniería de Software

Ingeniería del Software

La Ingeniería del Software, es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software. (IEEE , 2010)

que se enfoca en: diseño, análisis, implementación, mantenimiento
d esoft

¿Por qué la ingeniería del software no es simplemente una norma o un estándar? ¿Por qué es una disciplina, con todo lo que esto conlleva?

Cuando el software empezó a desarrollarse, por los años 50, se comienza con programas pequeños e intuitivos y esencialmente artesanales.

El desarrollador no tenía ni metodologías ni herramientas ni nada que le ayudara en su proceso ni tampoco en la gestión o control del proyecto.

Esto, unido a que la demanda era cada vez mayor, el hardware cada vez más potente y los sistemas más complejos, tuvo como consecuencia una productividad muy baja y una calidad de software entregado muy pobre.



La película Figuras Ocultas representa el desarrollo de software de los años '60.

Debido a esta situación, la OTAN decidió celebrar una conferencia en 1968 para poner sobre la mesa todos estos problemas, a los que denomina en su conjunto **Crisis del software**.

Para poner solución a esta Crisis del software propone precisamente una nueva disciplina a la que llama **Ingeniería del Software**.

Y así es como nace.

A partir de aquí se empiezan a definir metodologías y procedimientos, a implementar técnicas y a construir herramientas sobre las que fundamentar la Ingeniería del Software y que, por supuesto, hoy en día sigue en evolución y en desarrollo.

La **Ingeniería del software** tiene, por tanto, como objetivo proporcionar métodos, herramientas, procedimientos y técnicas con el fin de reducir el coste de los proyectos y mejorar la calidad del software.

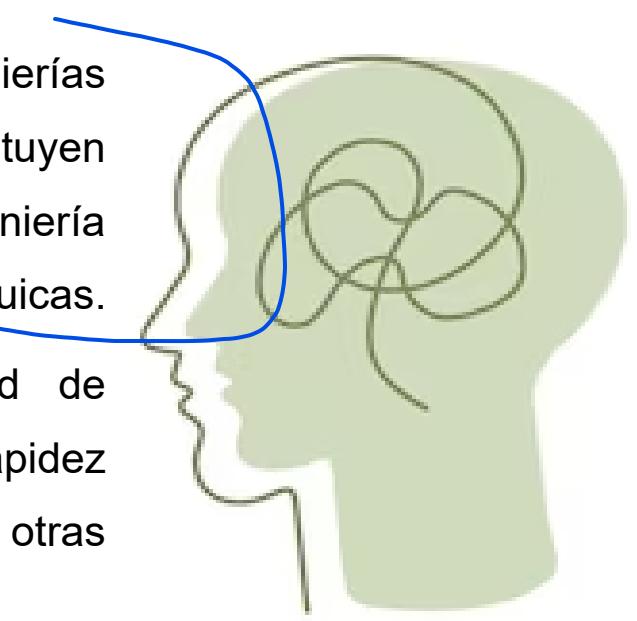
En otras palabras, ayuda a desarrollar y mantener su sistema informático asegurando su calidad, su fiabilidad y su productividad, entre otras características.

De esta forma, el humano siempre dirigirá al ordenador y no al revés.

Al igual que las demás ingenierías, la ingeniería del software ayuda a construir elementos que imitan, aumentan, ayudan, facilitan o sustituyen capacidades del ser humano.

La diferencia es que las demás ingenierías imitan, aumentan, ayudan, facilitan o sustituyen capacidades físicas mientras que la ingeniería del software lo hace con capacidades psíquicas.

Estas son, por ejemplo, la capacidad de memoria, la resolución de problemas, la rapidez de decisiones o la velocidad de cálculo, entre otras muchas.



Prácticas de desarrollo de Software

- Ingeniería de requisitos ↪
- Análisis de sistemas ↪
- Diseño/arquitectura a alto nivel ↪
- Diseño a bajo nivel ↪
- Codificación ↪
- Integración ↪
- Diseño y revisiones de código ↪
- Pruebas ↪
- Mantenimiento ↪
- Gestión de proyectos ↪
- Gestión de la configuración ✓

La Ingeniería del Software es el proceso de desarrollo que seguimos para construir un sistema informático y posteriormente mantenerlo, ajustándonos siempre a diferentes factores: recursos, coste, duración, calidad, etc.



Todo proyecto, y en particular un proyecto software, necesita de una planificación; es decir, debe definirse previamente paso a paso.

Esto también implica que tanto usuarios/clientes como desarrolladores, obtengan un beneficio del proyecto y sus resultados.

Si no se utiliza por el usuario, el proyecto es un fracaso.

Realizar un proyecto software no es sólo programar. Programar es una fase más dentro del proceso de desarrollo. pero el comenzar y terminar con éxito un proyecto software implica muchas más actividades, tareas y responsabilidades.

1.2. Ingeniería de Software / Metodología

Metodología

La metodología es la manera en que se aplica la Ingeniería de Software.

“La metodología es el conjunto de métodos que se utilizan en una determinada actividad con el fin de formalizarla y optimizarla.”

Es decir, la **metodología** determina los pasos a seguir y cómo realizarlos para poder finalizar con éxito el proyecto. En nuestro caso, esta actividad será nuestro proyecto software.

Cuando aplicamos metodologías a la ingeniería del software, lo que optimizamos es tanto el proceso como el propio producto software.

Además, nos proporciona **métodos** que guían en la gestión, el desarrollo y el control del proyecto. En definitiva, define qué hacer, cuándo y cómo hacerlo durante todo el desarrollo y mantenimiento de un proyecto. A este esquema se le llama “**ciclo de vida**”.

“El ciclo de vida es el conjunto de fases por las que pasa el sistema que se está desarrollando, desde que la idea inicial nace hasta que el sistema software es retirado o reemplazado.”

Por lo tanto, un ciclo de vida determina el orden de las fases del proceso software. También define las entradas y salidas de cada una de estas fases. En un principio, la salida de una fase será la entrada a la siguiente. Y, por último, establece los criterios de transición para pasar de una fase a otra.

Entonces tenemos tres conceptos clave: **proceso**, **ciclo de vida** y **metodología**.

El proceso únicamente nos da un conjunto de actividades que tenemos que realizar para conseguir un fin, que sería nuestro producto software.

El ciclo de vida va un poco más allá, y además de identificar estas actividades o fases, define el orden en que deben ejecutarse y los resultados intermedios y finales.



Por su parte, **las metodologías** nos dicen, además de lo anterior, el **cómo llevar a cabo el proyecto**. Para ello, nos proporciona un conjunto de métodos, herramientas y técnicas que facilitan las actividades de software y aumentan su productividad.

1.3. Ingeniería de Software / Naturaleza del Software

Naturaleza del Software

Para construir una silla, podríamos definir previamente las medidas, el color, el material, la forma, etc. Se puede dibujar un esquema que será fiel a la realidad, se puede construir, tocar, probar, sentarse encima, intentando romperla, etc.

Pero nada de esto se puede hacer con un software.

No se ve ni se toca lo que se está desarrollando. Y tampoco lo ve ni lo toca el usuario final, por lo que es difícil saber, ya no solo si le va a gustar o no, sino si es lo que quería o no lo es. Es más, ni siquiera al inicio del proyecto el cliente tiene claro lo que quiere y cómo lo quiere.

Por lo tanto, ¿qué hace que el software sea distinto? Varias características:

- Es un elemento lógico y no físico, por lo que no se puede ver ni tocar. Es inmaterial e invisible.
- Además, es desarrollado, no fabricado. No tenemos una cadena de producción donde se van incorporando piezas y los operarios se turnan.
- Esto quiere decir que no hay “piezas de repuesto”.
- Se deteriora, no se rompe. Se va degradando poco a poco.
- Y se construye a medida.
- El software está “vivo”, es decir, siempre está cambiando mientras se está utilizando.
- Tampoco es fácil comprender ni predecir el comportamiento que va a tener.
- Además, es complejo. Los sistemas actuales están formados por miles de funciones con interfaces complejas entre ellas.

Finalmente, el comprador no lo puede evaluar hasta que no haya sido desarrollado.

Salvo en el caso de las herramientas, no puede ver ningún sistema funcionando que sea idéntico al que se va a desarrollar para él.

Es por estas características intrínsecas del software y diferentes a cualquier otro material, que la ingeniería del software define metodologías, procedimientos y técnicas específicos para el desarrollo y gestión del software.

Referencias:

- Pressman, R. 2005. "Ingeniería del Software". 6^a Ed. McGraw-Hill.