

## Definiciones y etapas del proceso de software

### **Proceso**

Antes de definir lo que es un proceso de desarrollo de software, entendamos lo que es un proceso. Una definición sencilla de proceso es “serie de acciones que conducen a un final”. Esta definición parece coincidir con las ideas generales de la gente sobre procesos, pero deja muchas preguntas abiertas. ¿El proceso es la forma en que la organización opera —desde mercadotecnia hasta recursos humanos— o es la forma en que un desarrollador diseña, produce código, o prueba el software? ¿El proceso se refiere a administración, ingeniería, o ambas? ¿El proceso implica demasiada documentación y nos abstiene de desarrollar el producto objetivo?

La respuesta a éstas puede variar dependiendo de la perspectiva. Sin embargo, siempre que para alcanzar algún fin deseado necesitemos ejecutar una serie de acciones, y estas acciones tengan cierto orden, dependencias, roles responsables, resultados, tiempos de ejecución y herramientas de apoyo, estaremos hablando de procesos, que pueden ser predefinidos y personalizados.

La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con grandes posibilidades de éxito. Esta sistematización indica cómo se divide un proyecto en módulos más pequeños para normalizar cómo se administra el mismo.

Así, una metodología para el desarrollo de software son los procesos a seguir sistemáticamente para idear, implementar y mantener un producto de software desde que surge la necesidad del producto hasta que se cumple el objetivo por el cual fue creado.

De esta forma, las etapas del desarrollo de software son las siguientes:

### **Planificación**

Antes de empezar un proyecto de desarrollo de un sistema de información, es necesario hacer ciertas tareas que influirán decisivamente en el éxito del mismo. Dichas tareas son conocidas como el fuzzy front-end del proyecto, puesto que no están sujetas a plazos.

Algunas de las tareas de esta fase incluyen actividades como la determinación del ámbito del proyecto, la realización de un estudio de viabilidad, el análisis de los riesgos asociados, la estimación del coste del proyecto, su planificación temporal y la asignación de recursos a las diferentes etapas del proyecto.

### **Análisis**

Por supuesto, hay que averiguar qué es exactamente lo que tiene que hacer el software. Por eso, la etapa de análisis en el ciclo de vida del software corresponde al proceso a través del cual se intenta descubrir qué es lo que realmente se necesita y se llega a una comprensión adecuada de los requerimientos del sistema (las características que el sistema debe poseer).

## **Diseño**

En esta fase se estudian posibles opciones de implementación para el software que hay que construir, así como decidir la estructura general del mismo. El diseño es una etapa compleja y su proceso debe realizarse de manera iterativa.

Es posible que la solución inicial no sea la más adecuada, por lo que en tal caso hay que refinarla. No obstante, hay catálogos de patrones de diseño muy útiles que recogen errores que otros han cometido para no caer en la misma trampa.

## **Implementación**

En esta fase hay que elegir las herramientas adecuadas, un entorno de desarrollo que facilite el trabajo y un lenguaje de programación apropiado para el tipo de software a construir. Esta elección dependerá tanto de las decisiones de diseño tomadas como del entorno en el que el software deba funcionar.

Al programar, hay que intentar que el código no sea indescifrable siguiendo distintas pautas como las siguientes:

- Evitar bloques de control no estructurados.

- Identificar correctamente las variables y su alcance.

- Elegir algoritmos y estructuras de datos adecuadas para el problema.

- Mantener la lógica de la aplicación lo más sencilla posible.

- Documentar y comentar adecuadamente el código de los programas.

- Facilitar la interpretación visual del código utilizando reglas de formato de código previamente consensuadas en el equipo de desarrollo.

También hay que tener en cuenta la adquisición de recursos necesarios para que el software funcione, además de desarrollar casos de prueba para comprobar el funcionamiento del mismo según se vaya programando.

## **Pruebas**

Como errar es humano, la fase de pruebas del ciclo de vida del software busca detectar los fallos cometidos en las etapas anteriores para corregirlos. Por supuesto, lo ideal es hacerlo antes de que el usuario final se los encuentre. Se dice que una prueba es un éxito si se detecta algún error.

## **Instalación o despliegue**

La siguiente fase es poner el software en funcionamiento, por lo que hay que planificar el entorno teniendo en cuenta las dependencias existentes entre los diferentes componentes del mismo.

Es posible que haya componentes que funcionen correctamente por separado, pero que al combinarlos provoquen problemas. Por ello, hay que usar combinaciones conocidas que no causen problemas de compatibilidad.

## **Uso y mantenimiento**

Esta es una de las fases más importantes del ciclo de vida de desarrollo del software. Puesto que el software ni se rompe ni se desgasta con el uso, su mantenimiento incluye tres puntos diferenciados:

Eliminar los defectos detectados durante su vida útil (mantenimiento correctivo).

Adaptarlo a nuevas necesidades (mantenimiento adaptativo).

Añadirle nuevas funcionalidades (mantenimiento perfectivo).

Aunque suene contradictorio, cuanto mejor es el software más tiempo hay que invertir en su mantenimiento. La principal razón es que se usará más (incluso de formas que no se habían previsto) y, por ende, habrá más propuestas de mejoras.

## **¿Por qué contar con un proceso de software?**

Hasta hace poco tiempo, la producción de software era realizada con un enfoque artístico, a diferencia de un enfoque industrial. Ante la constante presencia de proyectos fallidos, y con el objetivo de mejorar la calidad de los productos, en los últimos años las organizaciones introdujeron los métodos de ingeniería de software (Ver Fundamentos – Desarrollar software es mucho más que programar).

A partir de estos, se formalizó el enfoque de ingeniería de producto para desarrollar software. Factores como la globalización han obligado a las organizaciones a contar con marcos de trabajo que las ayuden hacer las cosas de la manera más eficiente. Fue entonces que se incorporó la ingeniería de procesos al desarrollo de software.

## **Objetivo**

El objetivo de un proceso de desarrollo de programas es la formalización de las actividades relacionadas con el desarrollo del software de un sistema informático. La mayoría de los proyectos que se desarrollan, finalizan tarde, cuesta mucho mas de lo estimado