# requerimientos análisis diseño desarrollo pruebas implementación mantenimiento

# Definición de ingeniería de software.

Aplicación disciplinada de principios de ingeniería, ciencia y matemática, métodos y herramientas a la producción económica de software.. Humphrey]

# Características de calidad de software. FUMEPC

## Calidad del software:

-es un conjunto de características de un producto que satisfacen las necesidades de

los clientes y, en consecuencia, hacen satisfactorio el producto

-consiste en no tener deficiencias en el producto o en el proceso

Funcionalidad: Cumplir con las necesidades del cliente y poder satisfacer al mismo en la mayor cantidad de ámbitos posibles y de la mejor manera

Usabilidad: El software debería ser fácil de usar, y rápido de aprender a maniobrar con el mismo

Eficiencia: Que realice sus tareas en el menor tiempo posible, de la mejor manera. Ante un problema del sistema el mismo deberá poder revertir las operaciones

Confiabilidad: Debe brindarle confianza al usuario. El mismo debería poder confiar plenamente en que el funcionamiento del programa es correcto

Mantenibilidad: El código debe ser mantenible. Deben haber comentarios, nombres nemotécnicos y mas que nada se deberán unificar funcionalidades

Portabilidad: Debe ser fácil de instalar y poder coexistir con otros sistemas. Debe tener facilidad de instalación y pocas restricciones

# Tipos y niveles de requerimientos funcionales, no funcionales: negocio, usuario y funcionales.

### Requerimientos de Negocio:

Representan los objetivos de alto nivel que la organización o cliente requiere. Estos requerimientos provienen en general de los dueños, administradores, gerentes o patrocinadores del proyecto. Incluyen los objetivos del negocio, la visión del producto y el alcance del proyecto. Estos son los más importantes.

### Requerimientos del Usuario:

Los requerimientos del usuario especifican las tareas que los usuarios esperan poder realizar con el sistema, así como también los atributos o características que el sistema deberá mostrar para satisfacer las necesidades del usuario.

### Requerimientos de Sistema

Definición detallada de las funciones, servicios y restricciones operativas del sistema

#### Requerimientos Funcionales:

Los requerimientos funcionales describen el comportamiento o funcionalidades que el producto debe mostrar ante cada condición específica. El analista deriva y evalúa la funcionalidad que el sistema debe brindar al usuario final.

#### Requerimientos no funcionales:

Los requerimientos no funcionales describen las propiedades o restricciones sobre el comportamiento del sistema. En muchas ocasiones terminan siendo cosas muy simples como aspecto y diseño del producto final. Pero otras veces terminan siendo de alta demanda para el desarrollador.

### Características de requerimientos CCCNAV

* No ambiguo: El texto debe ser claro, preciso y tener una única interpretación posible.
* Conciso: Debe redactarse en un lenguaje comprensible por los inversores en lugar de uno de tipo técnico y especializado, aunque aun así debe referenciar los aspectos importantes.
* Consistente: Ningún requisito debe entrar en conflicto con otro requisito diferente, ni con parte de otro. Asimismo, el lenguaje empleado entre los distintos requisitos debe ser consistente también.
* Completo**:** Los requisitos deben contener en sí mismos toda la información necesaria, y no remitir a otras fuentes externas que los expliquen con más detalle.
* Alcanzable: Un requisito debe ser un objetivo realista, posible de ser alcanzado con el dinero, el tiempo y los recursos disponibles.
* Verificable: Se debe poder verificar con absoluta certeza, si el requisito fue satisfecho o no. Esta verificación puede lograrse mediante inspección, análisis, demostración o testeo.

# Ciclo de vida del software, tipos.

## Ciclo de vida

– Es un modelo de referencia de alto nivel de las actividades necesarias para el desarrollo de software.

– Un modelo de ciclo de vida:

• Describe las principales fases y actividades del desarrollo.

• Define el orden de las fases.

• Sirve para “saber donde estamos” durante un proyecto.

## Ciclo de vida scrum

Ciclo de vida que busca la entrega del proyecto perfecto, con el menor consumo de recursos posible. El equipo de trabajo se autogestiona y sus integrantes se complementan.

Scrum master: persona que lidera el equipo guiándolo para que cumpla las reglas y procesos de la metodología;

### Equipo scrum

Producto owner: Representante de los accionistas y clientes que usan el software, obtiene los requerimientos para luego parasarlos a un blacklog.

### Blacklog

### Sprints

### Scrum poker

### Dayly scrum meeting

## Cascada

Las fases del ciclo de vida (requisitos, análisis, diseño, etc.) se realizan (en teoría) de manera lineal, una única vez, y el inicio de una fase no comienza hasta que termina la fase anterior.

## Iterativo o incremental

Requerimiento y análisis, diseño arquitectonico

Se va liberando partes del producto (prototipos) periódicamente, en cada iteración, y cada nueva versión, normalmente, aumenta la funcionalidad y mejora en calidad respecto a la anterior.

## Evolutivo

Este modelo acepta que los requerimientos del usuario pueden cambiar en cualquier momento

# Herramientas de gestión de proyectos.

El trabajo de gestión de la configuración puede apoyarse en herramientas, que por ejemplo hagan el control de versiones y control de cambios en forma automática.

– Implementar un repositorio centralizado y de publicarlo para equipos distribuidos. – Controlar el flujo de trabajo definido en el circuito de cambios.

– Asignar permisos y estados.

Ejemplos de herramientas

## – Implementación de repositorio y control de versiones

• Microsoft Visual Source Safe

• ClearCase

• Surround SCM

• CVS (Concurrent Versioning System)

• Subversion

## – Seguimiento de incidentes y control de solicitudes de cambio

• Mantis

• Bugzilla

• Test Track Pro