

# Guía de Ingeniería de Software Unidad 1

Antonio Emiko Ochoa Adame

26 de febrero de 2019

## 1. Introducción

**Requisito.-** Propiedad que debe ser exhibida por un software para resolver un problema. (Se necesita para resolver un problema)

**Requisito.-** Condición o capacidad que necesita el usuario para resolver un problema o conseguir un objetivo determinado. (Lo necesita usuario para conseguir sus objetivos).

**Ingeniería de requisitos.-** Conjunto de actividades para descubrir, documentar y mantener un conjunto de requisitos. (Actividades para mantener bien los requisitos).

**Ingeniería de requisitos.-** Establecer los servicios que el cliente requiere de un sistema y las restricciones bajo las cuales opera y es desarrollado. (Restricciones y cosas necesarias para que se desarrolle el software).

**Proceso de ingeniería de requisitos.-** Conjunto estructurado de actividades de cuya ejecución se obtiene, valida y mantiene el documento de requisitos de un sistema. (Proceso que se lleva para mantener correcto el documento de requisitos).

**Gestión de requisitos.-** Actividad para gestionar los cambios en los requisitos de un sistema. (Gestión de cambios de requisitos).

## 2. Revisión de especificación de requerimientos

¿Qué implican los requisitos?

- Costo 15 %-20 % del total del proyecto.
- Error en requisitos es 100 veces peor que en error de código.
- Un error en requisitos genera errores en las demás etapas.
- Necesita gestión y soporte automatizado.

## 2.1. Especificación de requerimientos (ERS)

Es la descripción completa del comportamiento del sistema que se va a desarrollar. Aquí están involucrados los requisitos funcionales (casos de uso) y los requisitos no funcionales (complementarios; restricciones en el diseño o en la implementación).

Técnicas de levantamiento de requerimientos:

- Entrevistas
- Prototipos
- Casos de uso
- Cuestionarios
- Lluvia de ideas
- Etc.

## 2.2. Requerimientos según algunos estándares

### **IEEE 830**

Indica los elementos que debe tener una especificación de requerimientos.

### **PMBOK**

Indica los procesos para la gestión de proyectos.

### **BABOK**

Indica los procesos para el análisis del negocio.

## 2.3. Norma IEEE 830

Está enfocado en recomendaciones prácticas para la especificación de requerimientos.

Ayuda a describir de forma precisa qué quieren en el software (para el cliente) y establecer una estructura estándar para la ERS (para el desarrollador).

- Ayuda a elaborar el SRS (Software Requirements Specification) o ERS (Especificación de Requerimientos de Software).
- Guía de redacción de ERS (no es obligatorio).
- Hecha en 1998 por la Software Engineering Standards Committee del IEEE Computer Society.

### **¿Para qué sirve esta norma?**

- Para saber claramente lo que el cliente quiere

- Para establecer bases de un contrato
- Para reducir análisis, diseño y programación
- Para tener una referencia para validación de software
- Permite hacer mejoras o innovaciones

#### ¿Quién la debe utilizar?

- Un cliente
- Un desarrollador de “software a la medida”
- Un desarrollador de “software de paquete”

### 2.4. Trazabilidad de requisitos

Permite relacionar y establecer **dependencias** entre los requisitos y con otros elementos importantes del proyecto (diseño, componentes, testing, releases etc.).

#### Beneficios

- Gestión óptima del alcance de la solución
- Gestionar cambios con mínimo impacto
- Ayuda a reducir riesgos en el proyecto
- Ayuda a mantener consistencia entre requisitos
- Permite monitorizar y controlar el ciclo de vida de los requerimientos

#### Matriz de trazabilidad

## 3. Descripción de procesos actuales (Modelado de negocio)

Se hace mediante diagramas de flujo utilizando la metodología sipoc.

Diagrama sipoc consta de:

- Suppliers (Proveedores)
- Inputs (Entradas)
- Process (Procesos)

- Output (Salidas)
- Customers (Clientes)

**Proceso.-** Serie de tareas o actividades interrelacionadas para alcanzar un determinado fin.

**Análisis de procesos.-** Permite entender y medir las actividades de un proceso.

**Modelado de procesos.-** Requiere habilidades y técnicas que permitan comprender, comunicar, medir y gestionar los principales componentes de los procesos más importantes del negocio.

## 4. Estudio de factibilidad

Antes de proceder con un desarrollo debe evaluarse su viabilidad y analizar los riesgos que implica.

Aspectos a considerar:

- Económico (El beneficio compensa los costo?, vale la pena invertir?)
- Técnico (Funcionalidad, rendimiento o restricciones realizables?)
- Legal (Requisitos rompen ley o regla?)
- Operativa (Se puede implantar de manera efectiva en la empresa?, encaja en la filosofía de la empresa?, el personal está motivado a usar el software?)
- Plazos y calendario (El plazo es realista?, fecha elegidas son apropiadas?)

Hay 6 fases en el estudio de factibilidad.

## 5. Análisis costo-beneficio

Permite saber si una inversión vale la pena o no para el negocio.

Cuando se considera una desición, el costo de dicha desición se le resta al beneficio de la misma.

Requiere mucha experiencia técnica y de gestión.

Se deben considerar los elementos **tangibles** (se pueden valorar directamente) y los **intangibles** (subjetivos).

**Tangibles.-** Gastos de equipo, tiempo empleado, etc.

**Intangibles.-** Mayor competitividad, ventajas económicas, etc.

## **6. Roles**

### **6.1. Administrador del proyecto**

Administra y controla los recursos asignados a un proyecto para que se cumplan correctamente las metas establecidas.

### **6.2. Analista**

Se encargan de estudiar un problema de complejidad determinada, descomponiendo el problema en subproblemas de menor complejidad.

### **6.3. Diseñador**

Encargado de generar el diseño del sistema.

Sus funciones:

- Generar el diseño arquitectónico y diseño detallado del sistema, basándose en los requisitos
- Generar prototipos rápidos
- Generar el documento de diseño
- Asegurarse de que el producto final se apegue al diseño realizado

### **6.4. Programador**

Debe convertir la especificación del sistema en código fuente ejecutable utilizando uno o más lenguajes de programación, así como herramientas de software de apoyo a la programación.

### **6.5. Téster**

Objetivos:

- Construir y aplicar los planes de prueba unitarios, de módulo, de sistema, y aceptación parcial, manteniéndolos actualizados durante el proyecto
- Velar por la adhesión al estándar adoptado para el desarrollo
- Velar por la calidad del producto final (cumplimiento de los requisitos)
- Coordinar las inspecciones

## 6.6. Asegurador de calidad

Se encarga de que la calidad del software no se vea comprometida debido a la presión por cumplir con las fechas o por tratar de reducir los costos, entre otros.

**Disclaimer:** La finalidad de este documento es servir como apoyo de estudio. El autor de la versión original de este documento no se hace responsable del uso indebido del mismo.