

Ingeniería de Software I (semana 1)

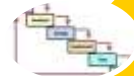
Agenda



Ingeniería de Software



Modelos de Procesos – Ciclo de vida del Software



Métodos tradicionales



Paradigmas de la orientación a objetos



Proceso de Desarrollo de Software



Análisis de requisitos (Requerimientos)

¿Qué es Ingeniería de Software?

Según [1], *software engineering (Ingeniería de Software)* se define como:

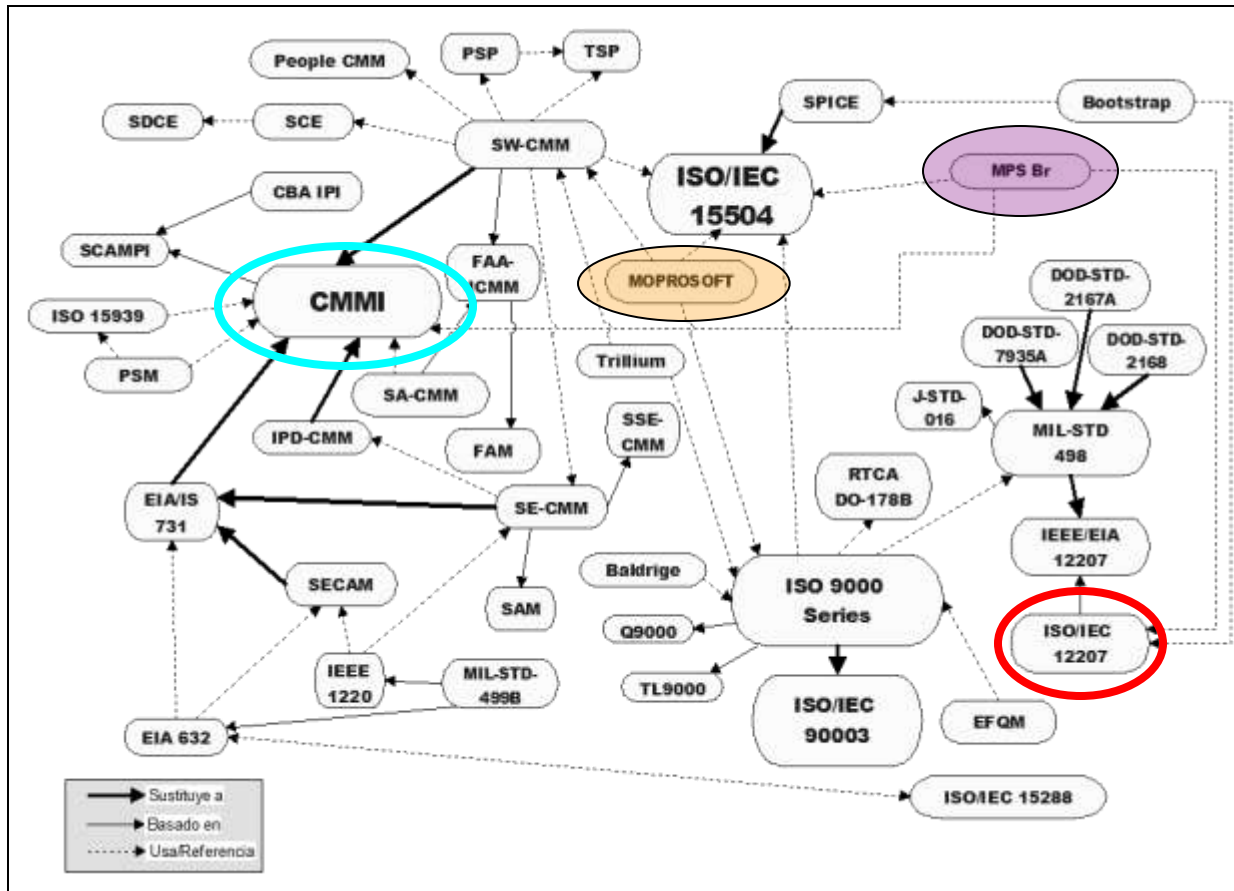
“(1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.

(2) The study of approaches as in (1)”.

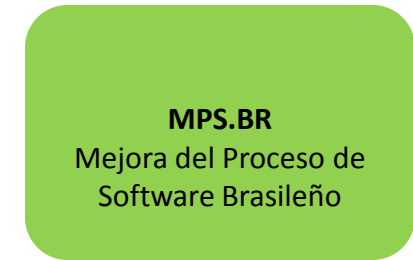
[1] 610.12-1990 - IEEE Standard Glossary of Software Engineering Terminology

<http://www.idi.ntnu.no/grupper/su/publ/ese/ieee-se-glossary-610.12-1990.pdf> página 67

Modelos de procesos del ciclo de vida del Software, normas y estándares



MOPROSOFT
Modelo de Procesos de
Software mexicano



Procesos de Contexto del Sistema

Procesos de Contratación

Proceso de Adquisición
(Cláusula 6.1.1)

Proceso de Suministro
(Cláusula 6.1.2)

Procesos Organizacionales de Habilitación del Proyecto

Proceso de Gestión del Modelo
de Ciclo de Vida (Cláusula 6.2.1)

Proceso de Gestión de la
Infraestructura (Cláusula 6.2.2)

Proceso de Gestión del
Portafolio del Proyecto (Cláusula
6.2.3)

Proceso de Gestión de los
Recursos Humanos
(Cláusula 6.2.4)

Proceso de Gestión de la
Calidad (Cláusula 6.2.5)

Procesos del Proyecto

Proceso de Planificación del
Proyecto
(Cláusula 6.3.1)

Proceso de Evaluación y
Control del Proyecto
(Cláusula 6.3.2)

Proceso de Gestión de
Decisiones
(Cláusula 6.3.3)

Proceso de Gestión del
Riesgo (Cláusula 6.3.4)

Proceso de Gestión de la
Configuración (Cláusula
6.3.5)

Proceso de Gestión de la
Información (Cláusula 6.3.6)

Proceso de Medición
(Cláusula 6.3.7)

Procesos Técnicos

Proceso de Definición de Requisitos
de las Partes Interesadas (Cláusula
6.4.1)

Proceso de Análisis de Requisitos del
Sistema (Cláusula 6.4.2)

Proceso de Diseño Arquitectural del
Sistema (Cláusula 6.4.3)

Proceso de Implementación (Cláusula
6.4.4)

Proceso de Integración del Sistema
(Cláusula 6.4.5)

Proceso de Pruebas de Calificación
del Sistema (Cláusula 6.4.6)

Proceso de Instalación del Software
(Cláusula 6.4.7)

Proceso de Soporte de la Aceptación
del Software (Cláusula 6.4.8)

Proceso de Operación del Software
(Cláusula 6.4.9)

Proceso de Mantenimiento del
Software (Cláusula 6.4.10)

Proceso de Retiro del Software
(Cláusula 6.4.11)

Procesos Específicos del Software

Procesos de Implementación del Software

Proceso de Implementación del
Software (Cláusula 7.1.1)

Proceso de Análisis de Requisitos
del Software (Cláusula 7.1.2)

Proceso de Diseño Arquitectural
del Software (Cláusula 7.1.3)

Proceso de Diseño Detallado del
Software (Cláusula 7.1.4)

Proceso de Construcción del
Software (Cláusula 7.1.5)

Proceso de Integración del
Software (Cláusula 7.1.6)

Proceso de Pruebas de
Calificación del Software
(Cláusula 7.1.7)

Procesos de Soporte del Software

Proceso de Gestión de la
Documentación del Software
(Cláusula 7.2.1)

Proceso de Gestión de la
Configuración del Software
(Cláusula 7.2.2)

Proceso de Aseguramiento de la
Calidad del Software (Cláusula
7.2.3)

Proceso de Verificación del
Software (Cláusula 7.2.4)

Proceso de Validación del Software
(Cláusula 7.2.5)

Proceso de Revisión del Software
(Cláusula 7.2.6)

Proceso de Auditoría del Software
(Cláusula 7.2.7)

Proceso de Resolución de
Problemas del Software (Cláusula
7.2.8)

Procesos de Reutilización del Software

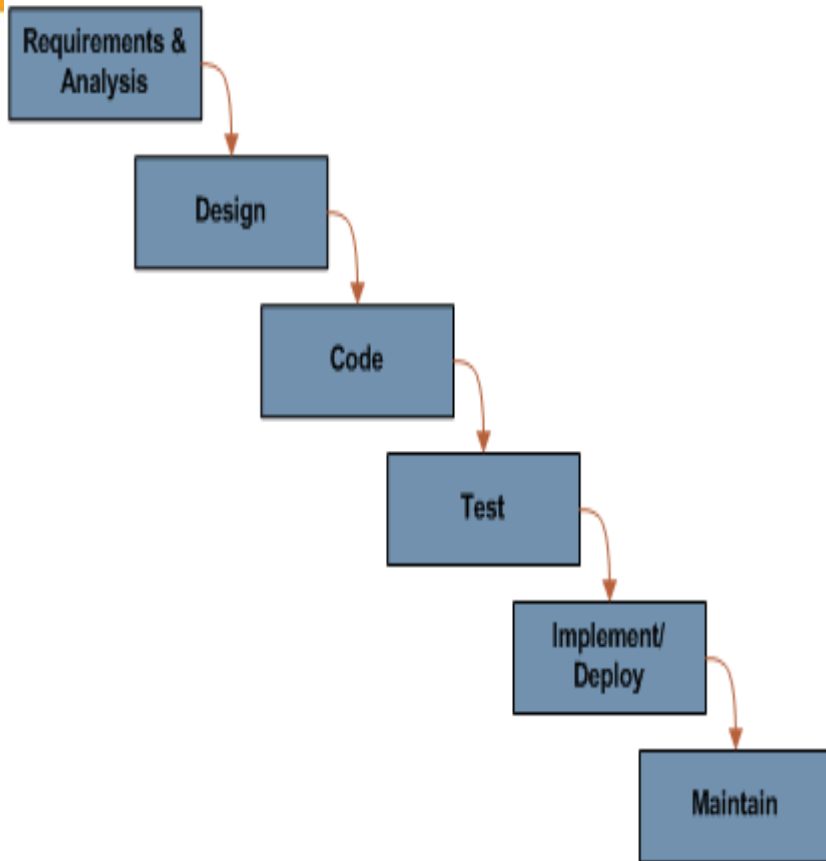
Proceso de Ingeniería de Dominio
(Cláusula 7.3.1)

Proceso de Gestión de Activos de
Reutilización (Cláusula 7.3.2)

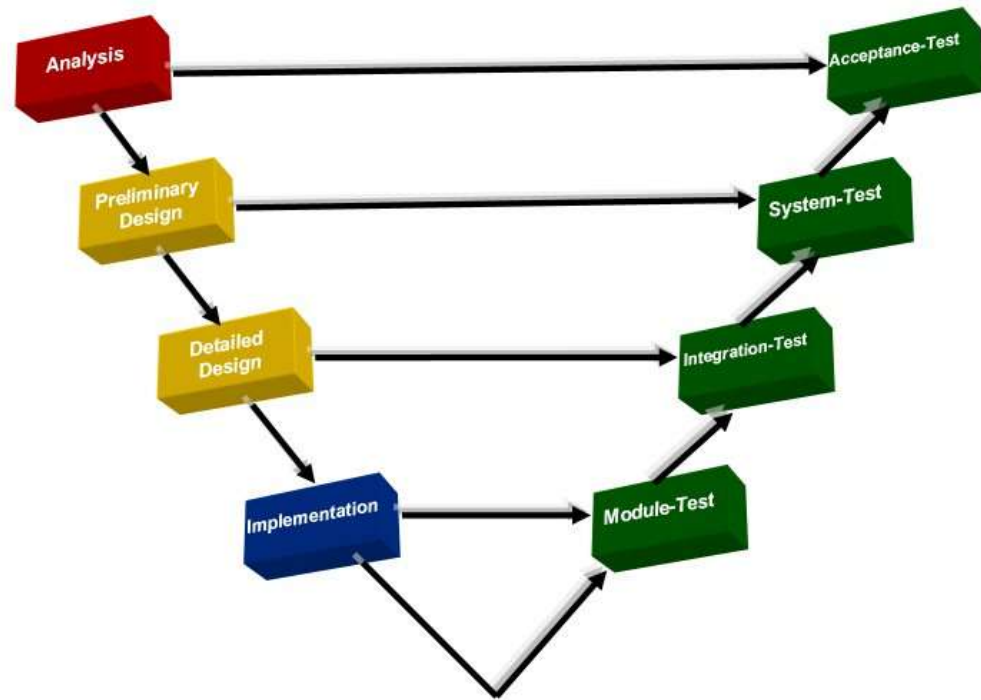
Proceso de Gestión de Programas
de Reutilización (Cláusula 7.3.3)

Figura 1. Grupos de procesos del ciclo de vida

Métodos tradicionales



Waterfall Model –
Modelo cascada (Royce, 1970)
<http://www.waterfall-model.com/>



V- Waterfall Model –
Modelo V (Brook, 1986)
<http://www.waterfall-model.com/>

Paradigmas de la orientación a objetos

Problemas

- Pocos sistemas lograban terminarse
- Pocos se terminaban con los requisitos iniciales
- No todos los que se terminaban y cumplían con los requisitos, se usaban según lo planificado

“Adaptar el software a nuevos requerimientos imposibles de haber sido planificados inicialmente, era imposible”

La **orientación a objetos** brinda estos métodos de experimentación, no exige la planificación de un proyecto por completo antes de escribir la primera línea de código

¿Qué es la Orientación a Objetos?

- El concepto surge en los **lenguajes de programación**
 - Se organiza el software como una colección de objetos discretos que encapsulan
 - » Estructuras de Datos y
 - » Comportamiento.
 - Un sistema OO funciona mediante la colaboración entre los objetos que se comunican entre sí.
- El concepto se extiende a los **métodos de análisis y diseño**
 - Se utilizan los objetos del mundo real como base para construir modelos
 - Los elementos que forman los sistemas del mundo real se corresponden con objetos software

Recordemos conceptos básicos

- **Clase y Objeto**

- Atributos
- Operaciones
- Comportamiento
- Identidad

- **Interfaz**

- **Asociación**

- Agregación
- Herencia

- **Polimorfismo**

- **Paso de mensajes**

- **Encapsulación**

Proceso de Desarrollo de Software



Software Requirement Specification v1 – SRS y SS

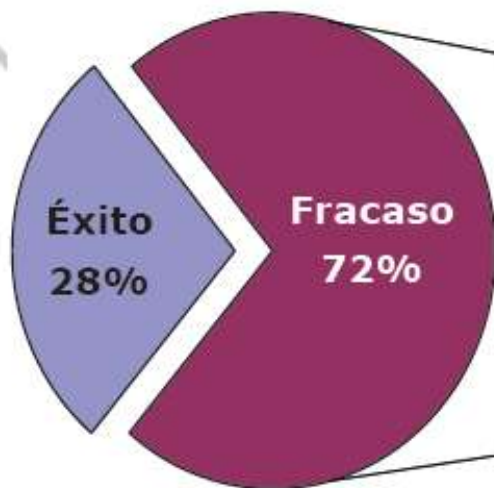
Documento de Análisis de Software – DAS O SRS v2

Documento de Diseño de Software - DDS

Componentes y Producto Software

IEEE-STD 830

Ingeniería de Requerimientos

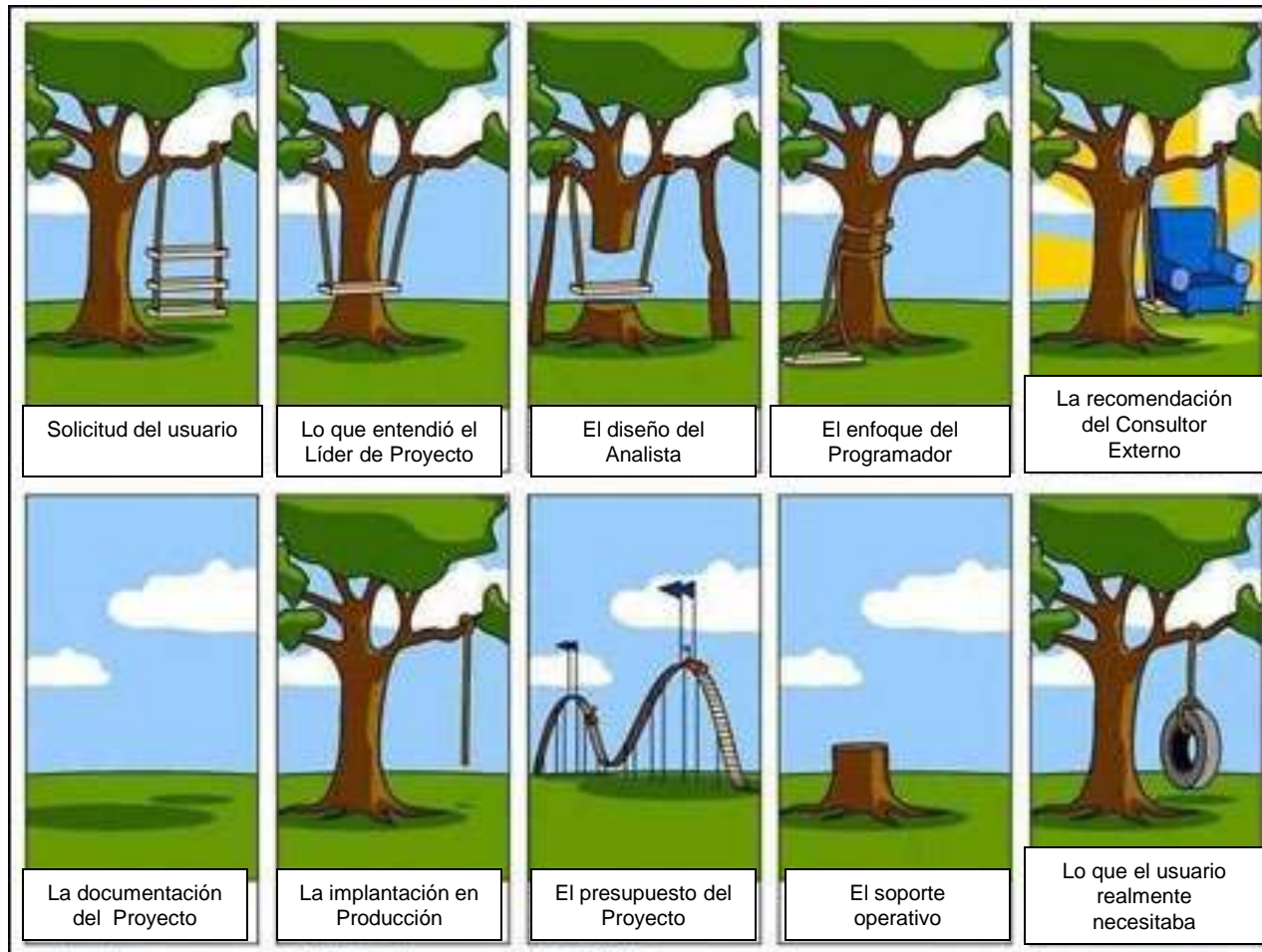


- Requisitos incompletos: 13%
- Poca participación de los usuarios: 12%
- Expectativas poco realistas: 10%
- Cambios a los requisitos: 9%
- Requisitos innecesarios: 8%

Fuente: Standish Group, CHAOS CHRONICLES

http://intesysconsulting.com/yahoo_site_admin/assets/docs/Porque-fracasan-los-proyectos-de-TI.169152608.pdf página 13

Ingeniería de Requerimientos



Interpretación de los requerimientos

<http://www.youtube.com/watch?v=gInrQ2fymSg&playnext=1&list=PL73C633E7F3AB6C93>

“No hay viento favorable para el que no sabe a dónde va”
Lucio Anneo Séneca, filósofo
(4 a.C. – 65 d.C.)

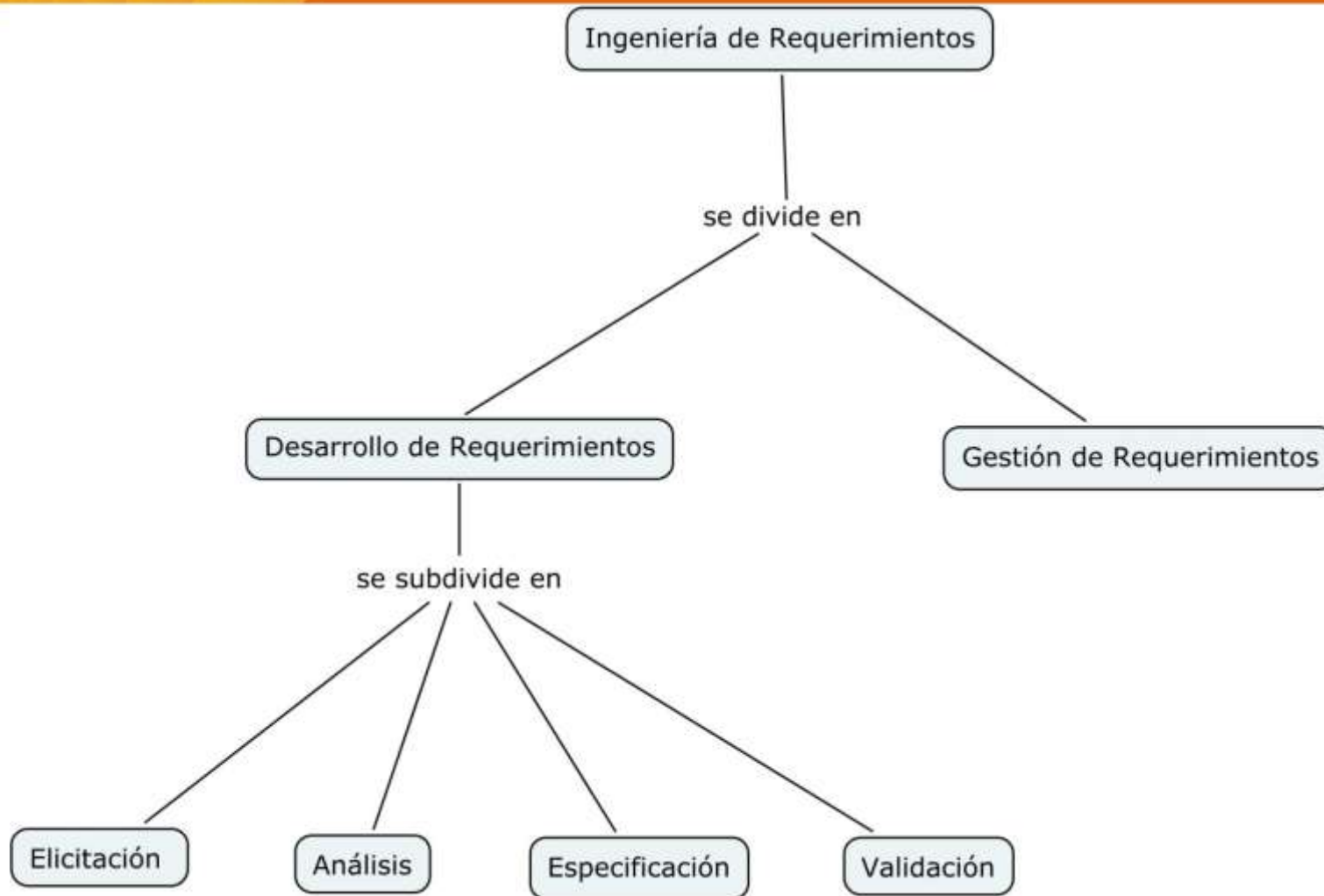
Ingeniería de Requerimientos

Denota el proceso de especificar los requerimientos a través del **estudio de las necesidades de los involucrados** (stakeholders), analizar sistemáticamente y refinar dichas especificaciones.

Objetivos de la Ingeniería de Requerimientos

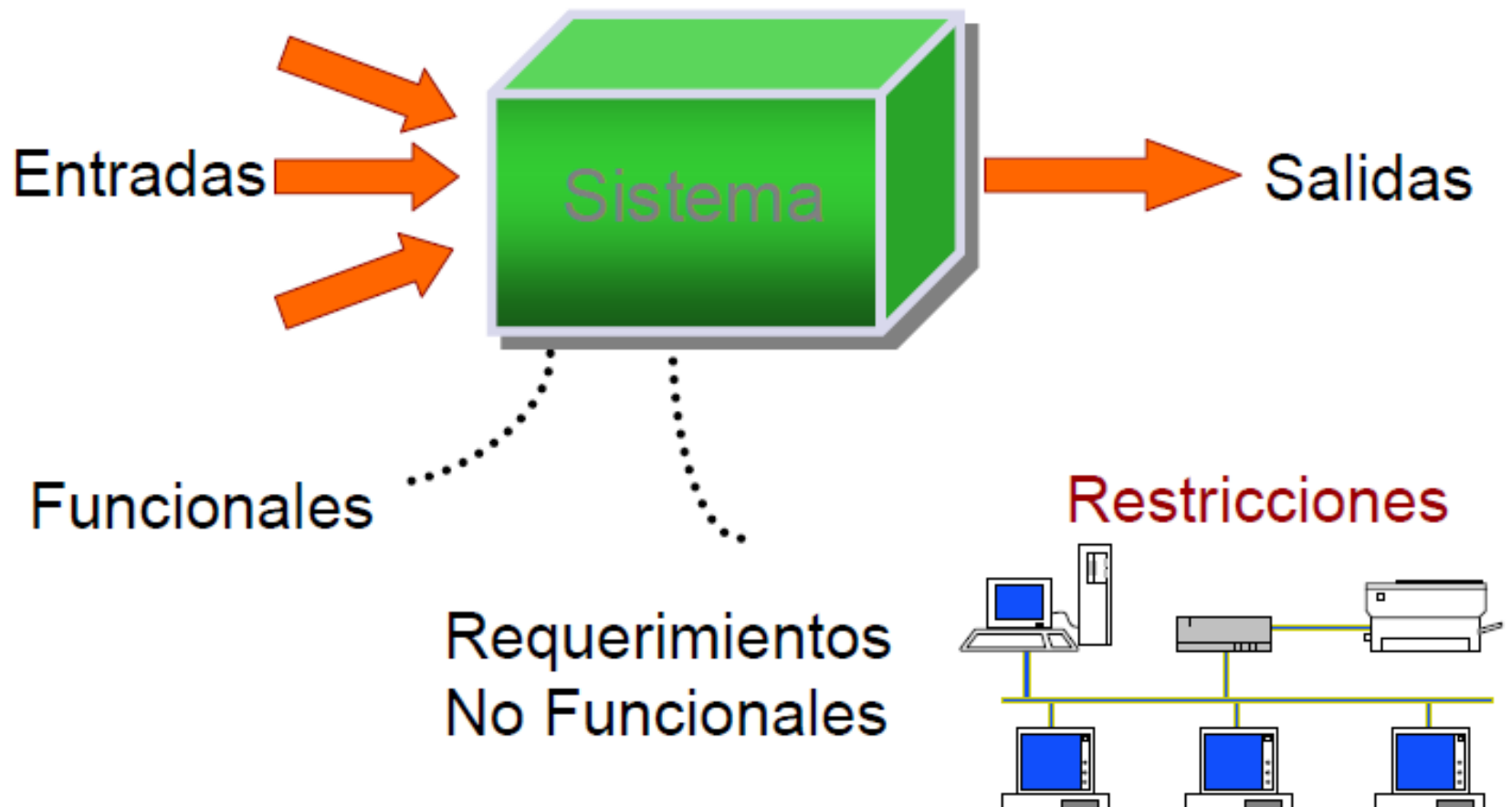
- Asegurar que el **producto final** cumpla con los **requerimientos propuestos** por el usuario.
- Establecer y mantener acuerdos entre los usuarios y el equipo de desarrollo acerca del tratamiento a los cambios a los requerimientos.

Dominio de la Ingeniería de Requerimientos



<http://www.youtube.com/watch?v=t4PRpQ5xAC4>

Requerimientos de Software



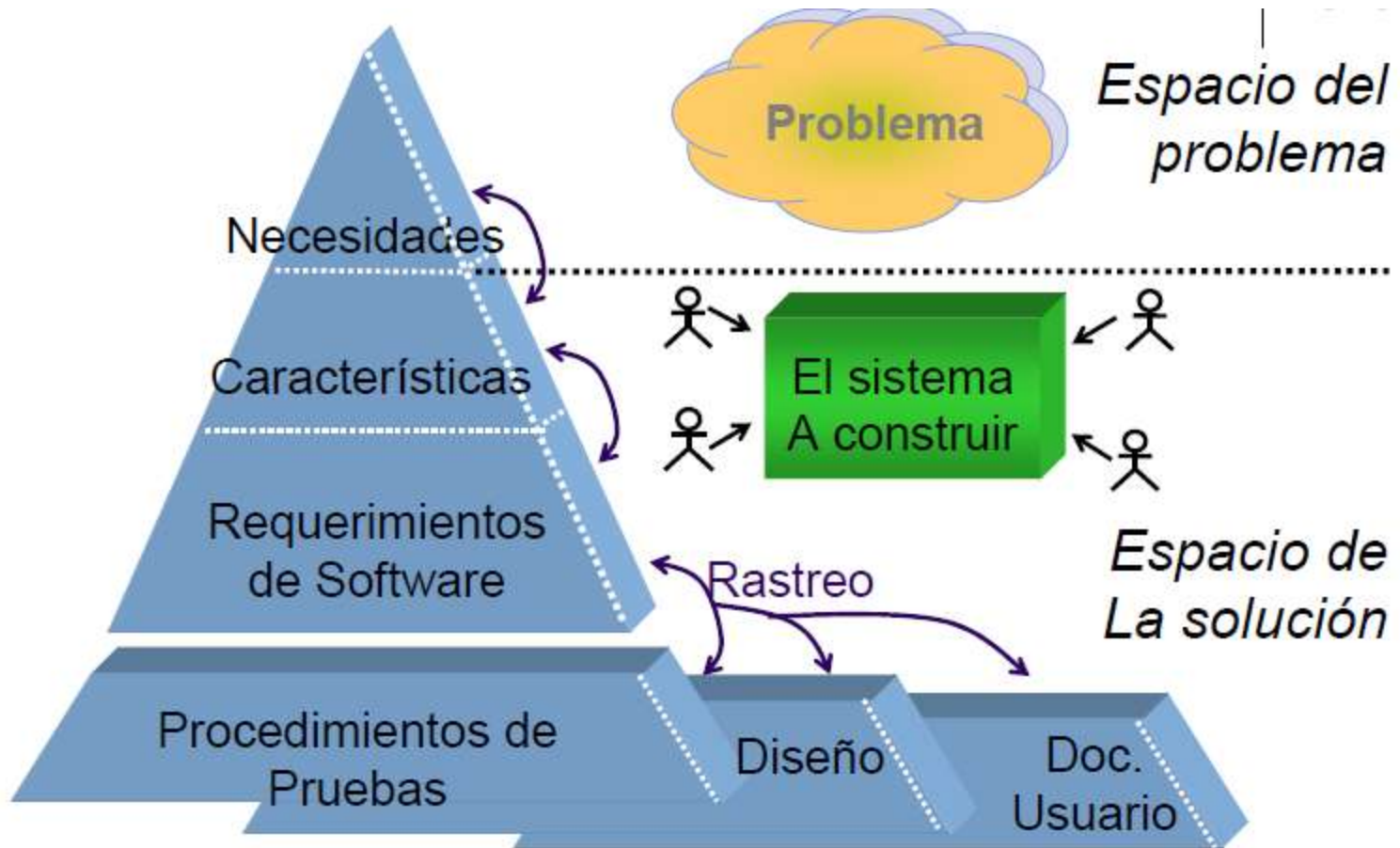
Qué no son requerimientos

- Las especificaciones de requerimientos **no incluyen detalles de diseño o implementación** (restricciones), información de la planificación del proyecto o información de las pruebas.

Entonces:

- Hay que separar dichas cosas de los requerimientos de tal manera que las actividades de requerimientos se centren en **comprender lo que el equipo intenta construir.**

Introducción a los requerimientos

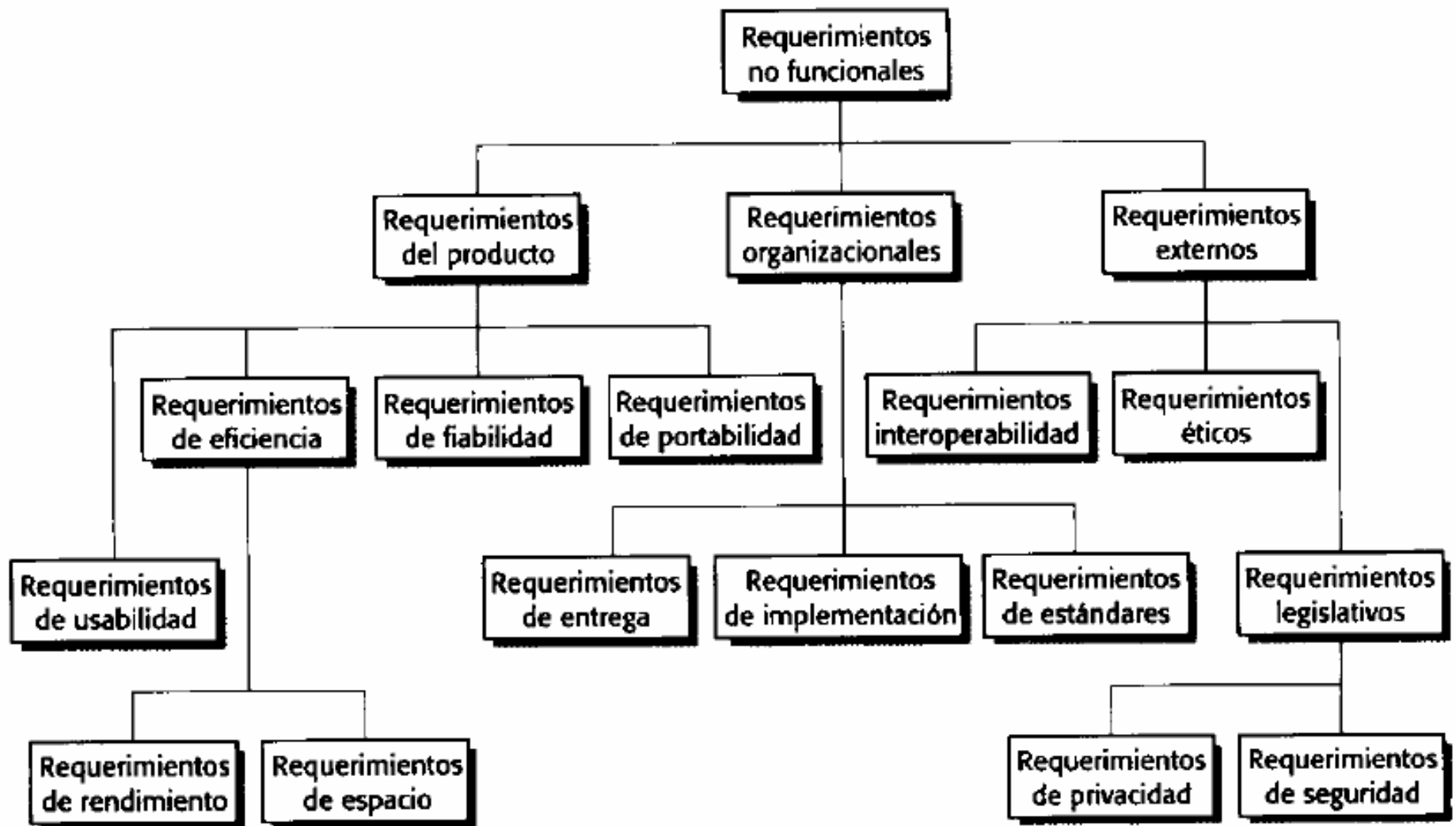


Definiciones

Definición	Descripción
Necesidades de los Stakeholders	Una expresión, formulada por un <i>stakeholder</i> , independiente de la solución sobre el estado deseado en la solución
Características	Describe los servicios, observables externamente, que el sistema deberá proporcionar para satisfacer las necesidades de los <i>stakeholders</i>
Requerimientos de Software	Capacidad o condición que deberá ser alcanzada por el producto software
<ul style="list-style-type: none">• Funcionales	Un requerimiento, desde una perspectiva de caja negra, que especifica como la solución interactúa con el mundo externo
<ul style="list-style-type: none">• No Funcionales	Un requerimiento que expresa, desde una perspectiva de caja negra, los atributos de calidad de la solución
Restricción	Una limitación en el diseño del software o el proceso usado para construir el sistema.

Clasificación

Requerimientos NO Funcionales



Ejemplo: Sistema de Solicitudes de Cambios a Software/Aplicaciones

Ejemplos

Necesidades de los *Stakeholders*

- Help Desk y Testers necesitan notificar al administrador de proyectos cuando una “solicitud de cambio” es creada o cambia de estado.

Características

- Software orientado a Workflow. La solicitud de cambio pasará a través de una serie de estados y asignaciones.

Requerimientos de Software

- Funcionales

El caso de uso se inicia cuando el tester elige “Reportar Defecto”. El sistema muestra una lista de proyectos ...

- No Funcionales

El sistema debe soportar una carga máxima de 500 usuarios, utilizando el sistema concurrentemente.

Restricción

- Para construir el sistema se deberá utilizar C#, ASP .NET, framework .NET 2.0 y SQL Server 2005.

Los requerimientos existen a muchos niveles



1. IEEE-STD 830: Sección 1

Introducción

Tabla de Contenido

Sección 1 del SRS

1. Introducción
 - 1.1 Propósito
 - 1.2 Alcance
 - 1.3 Definiciones, siglas, y abreviaciones
 - 1.4 Referencias
 - 1.5 Apreciación global

Sección 2 del SRS

2. Descripción global
 - 2.1 Perspectiva del producto
 - 2.2 Funciones del producto
 - 2.3 Características del usuario
 - 2.4 Restricciones
 - 2.5 Supuestos
 - 2.6 Priorizar los requisitos.

Sección 3 del SRS

3. Los requisitos específicos
 - 3.1 Interfaces externas
 - 3.2 Requisitos Funcionales
 - 3.3 Requisitos de desarrollo
 - 3.4 Requisitos lógicos de datos
 - 3.5 Restricciones de diseño
 - 3.6 Atributos del software

Apéndices
Índice

1. ISO/IEC/IEEE 29148



1. Introduction

- 1.1 Purpose
- 1.2 Scope
- 1.3 Product overview
 - 1.3.1 Product perspective
 - 1.3.2 Product functions
 - 1.3.3 User characteristics
 - 1.3.4 Limitations
- 1.4 Definitions

2. References

3. Specific requirements

- 3.1 External interfaces
- 3.2 Functions
- 3.3 Usability Requirements
- 3.4 Performance requirements
- 3.5 Logical database requirements
- 3.6 Design constraints
- 3.7 Software system attributes
- 3.8 Supporting information

4. Verification

(parallel to subsections in Section 3)

5. Appendices

- 5.1 Assumptions and dependencies
- 5.2 Acronyms and abbreviations

Figure 8 — Example SRS Outline

EJERCICIO 1

-Identificar requerimientos funcionales y no funcionales de WAZE APP

EJERCICIO 2

1. Instalar CMAP Tools
2. Construir mapa conceptual de:
IEEE 830

¿Consultas?

