Revisión de Aspectos Relevantes de la Ingeniería de Software

Juan Carlos Cabanillas Noris

19 de Marzo de 2016

¿Qué es la Ingeniería de Software?

- La *IEEE Computer Society* define a la ingeniería de software como:

  "La aplicación de una sistemática, disciplinada de una aproximación cuantificable para el desarrollo, operación y mantenimiento del software…"

- Área importante dentro de la teoría y práctica de la computación.

SWEBOK

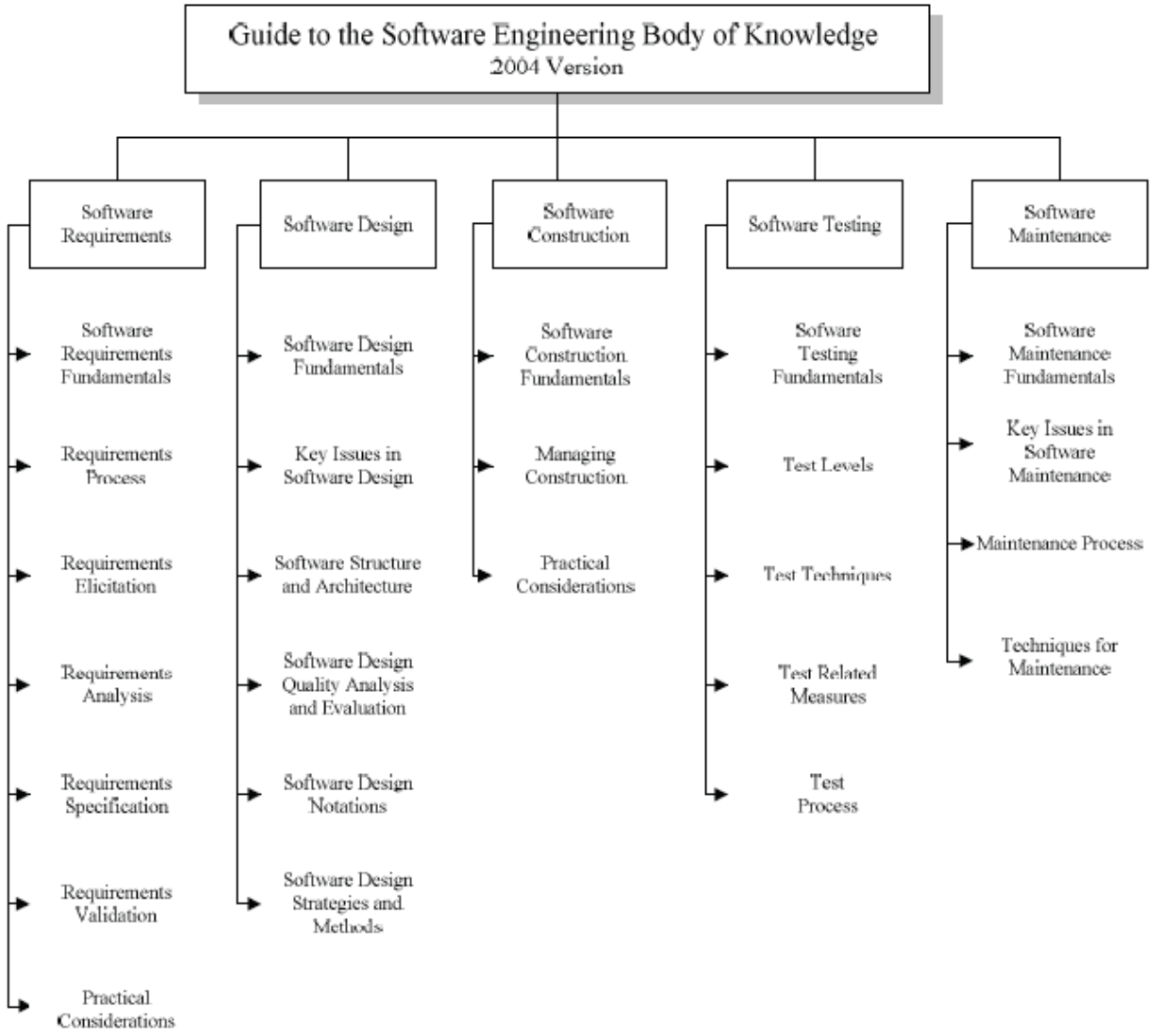
(Software Engineering Body of Knowledge)

- Documento creado por la Software Engineering Coordinating Committee (IEEE Computer Society).
- Guía al conocimiento presente en el área de la Ingeniería del Software.
- Existe un amplio consenso respecto a los contenidos de la disciplina.
- El material está organizado en Áreas de Conocimiento (Knowledge Areas, KAs)

Áreas del Conocimiento de SWEBOK

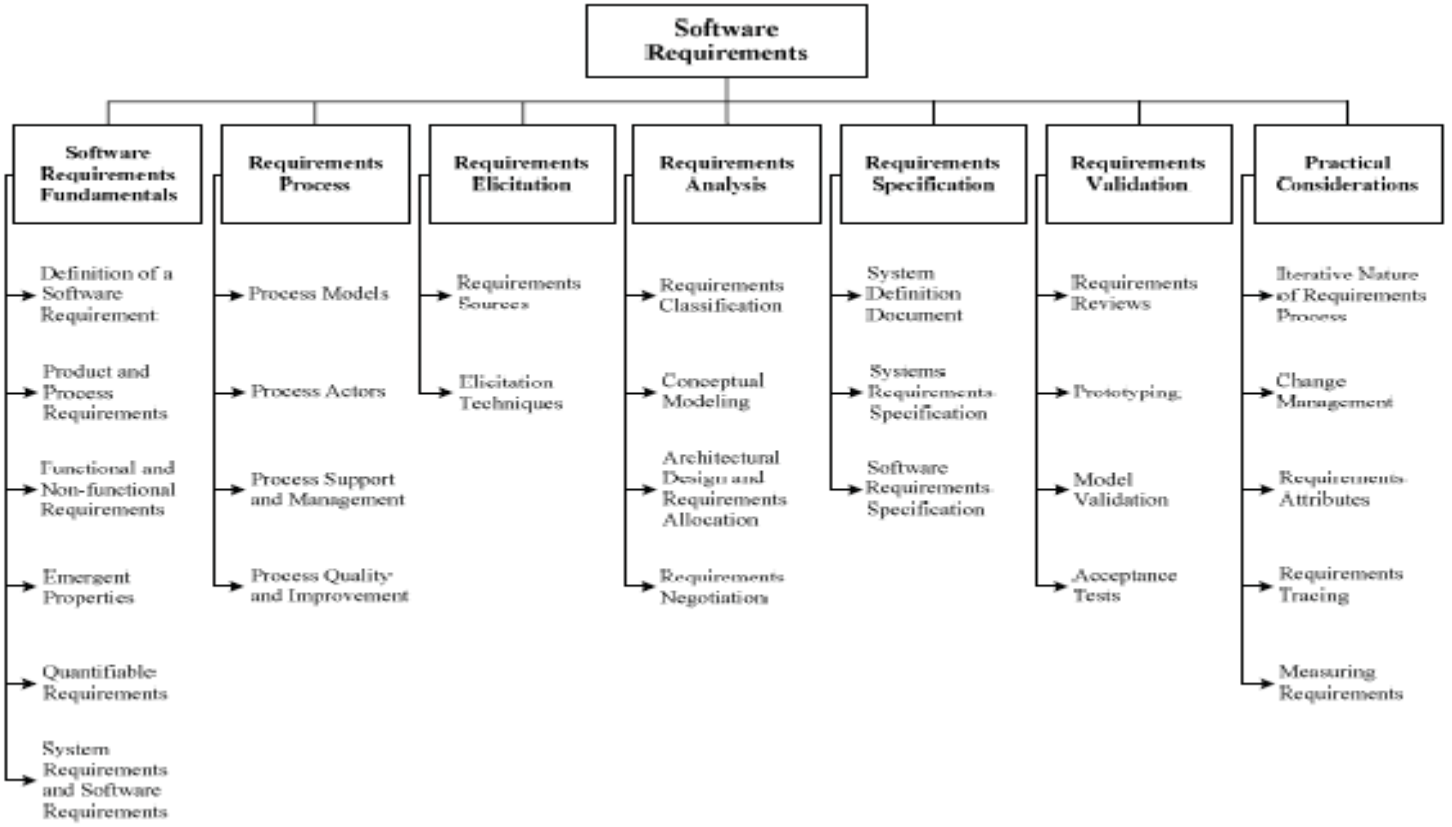
a) *Requerimientos de Software*
b) *Diseño de Software*
c) *Construcción de Software*
d) *Pruebas de Software*
e) *Mantenimiento de Software*
f) *Administración de la Configuración de Software*
g) *Administración de la Ingeniería de Software*
h) *Procesos de Ingeniería de Software*
i) *Herramientas y métodos de la Ingeniería de Software*
j) *Calidad de Software*

Primeras Cinco KAs



a) Requerimientos de Software

"Los requerimientos para un sistema son descripciones de lo que el sistema debe hacer: el servicio que ofrece y las restricciones de su operación".



Aspectos relevantes de los *Requerimientos de Software*

- Requerimientos de usuario: Representan los requisitos abstractos de alto nivel.
  - Lenguaje natural
  - Diagramas
  - Servicios que esperan los usuarios
  - Restricciones de operación
- Requerimientos del sistema: Descripción detallada de lo que el sistema debe hacer.
  - Servicios
  - Restricciones de operación del sistema
  - Documento (especificaciones funcionales)

Aspectos relevantes de los *Requerimientos de Software*

Los requerimientos del sistema se suelen clasificar en:

- Requerimientos funcionales del sistema:
  - Servicios que debe proveer
  - Cómo debe reaccionar a entradas particulares
  - Comportamiento en situaciones especificas

- Requerimientos No funcionales del sistema
  - Limitaciones sobre servicios o funciones
  - Restricciones impuestas por estándares
  - Se aplican al sistema como un todo

Aspectos relevantes de los *Requerimientos de Software*

Chapter	Description
Preface	This should define the expected readership of the document and describe its version history, including a rationale for the creation of a new version and a summary of the changes made in each version.
Introduction	This should describe the need for the system. It should briefly describe the system's functions and explain how it will work with other systems. It should also describe how the system fits into the overall business or strategic objectives of the organization commissioning the software.
Glossary	This should define the technical terms used in the document. You should not make assumptions about the experience or expertise of the reader.
User requirements definition	Here, you describe the services provided for the user. The nonfunctional system requirements should also be described in this section. This description may use natural language, diagrams, or other notations that are understandable to customers. Product and process standards that must be followed should be specified.
System architecture	This chapter should present a high-level overview of the anticipated system architecture, showing the distribution of functions across system modules. Architectural components that are reused should be highlighted.
System requirements specification	This should describe the functional and nonfunctional requirements in more detail. If necessary, further detail may also be added to the nonfunctional requirements. Interfaces to other systems may be defined.
System models	This might include graphical system models showing the relationships between the system components and the system and its environment. Examples of possible models are object models, data-flow models, or semantic data models.
System evolution	This should describe the fundamental assumptions on which the system is based, and any anticipated changes due to hardware evolution, changing user needs, and so on. This section is useful for system designers as it may help them avoid design decisions that would constrain likely future changes to the system.
Appendices	These should provide detailed, specific information that is related to the application being developed; for example, hardware and database descriptions. Hardware requirements define the minimal and optimal configurations for the system. Database requirements define the logical organization of the data used by the system and the relationships between data.
Index	Several indexes to the document may be included. As well as a normal alphabetic index, there may be an index of diagrams, an index of functions, and so on.

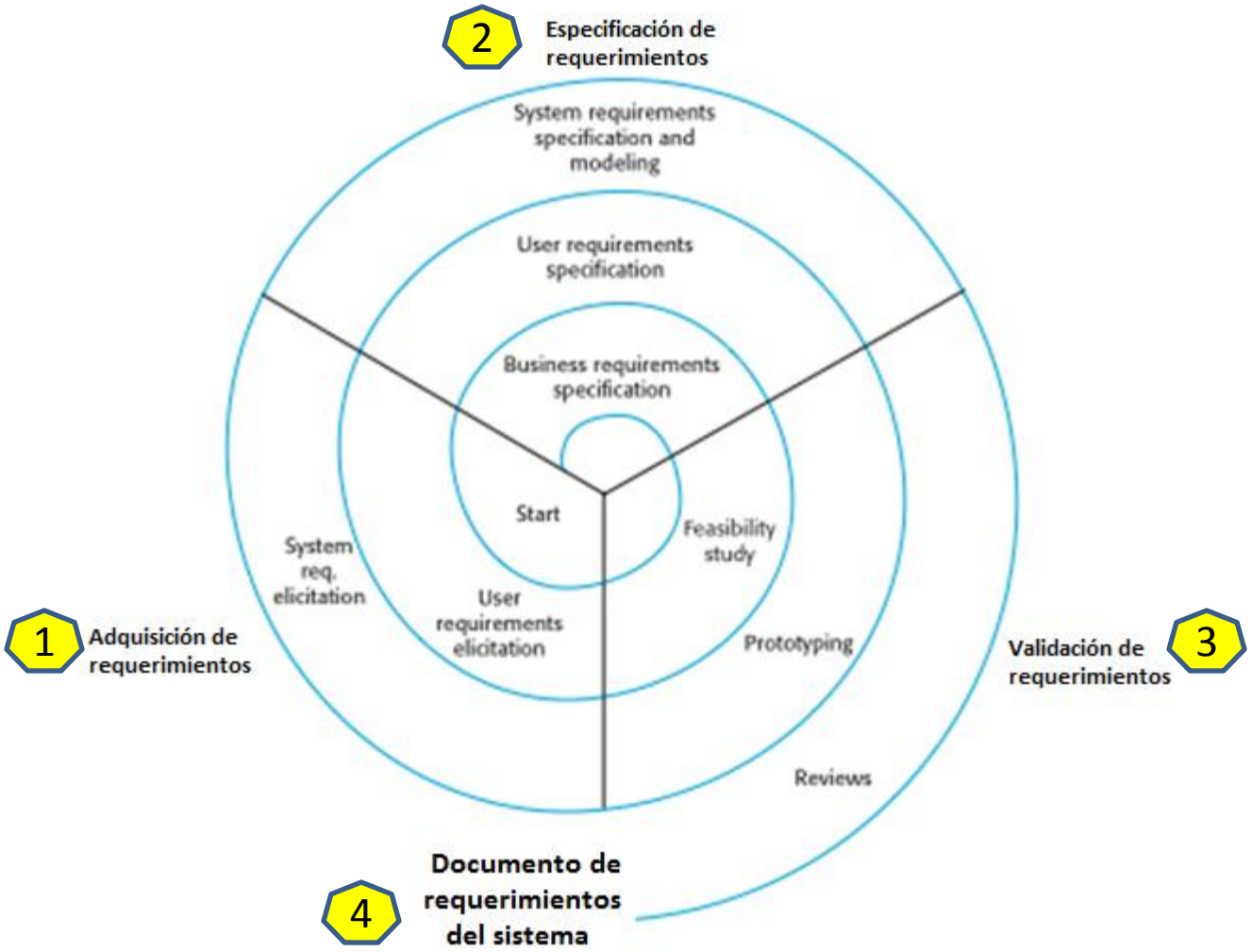
Estructura del documento de requerimientos

Aspectos relevantes de los *Requerimientos de Software*

Notation	Description
Natural language	The requirements are written using numbered sentences in natural language. Each sentence should express one requirement.
Structured natural language	The requirements are written in natural language on a standard form or template. Each field provides information about an aspect of the requirement.
Design description languages	This approach uses a language like a programming language, but with more abstract features to specify the requirements by defining an operational model of the system. This approach is now rarely used although it can be useful for interface specifications.
Graphical notations	Graphical models, supplemented by text annotations, are used to define the functional requirements for the system; UML use case and sequence diagrams are commonly used.
Mathematical specifications	These notations are based on mathematical concepts such as finite-state machines or sets. Although these unambiguous specifications can reduce the ambiguity in a requirements document, most customers don't understand a formal specification. They cannot check that it represents what they want and are reluctant to accept it as a system contract

Forma de escribir una especificación de requerimientos del sistema

Aspectos relevantes de los *Requerimientos de Software*



Vista en Espiral del proceso de ingeniería de requerimientos

Aspectos relevantes de los *Requerimientos de Software*

ADQUISICIÓN DE REQUERIMIENTOS	VALIDACIÓN DE REQUERIMIENTOS	ADMINISTRACIÓN DE REQUERIMIENTOS
Descubrimiento de Requerimientos -Descubrimiento -Entrevistas -Escenarios -Casos de uso -Etnografía	Comprobaciones de validez	Planeación de la Admón. de requerimientos
Clasificación y organización de requerimientos	Comprobación de consistencia	Administración del cambio
Priorización y negociación de requerimientos	Comprobación de totalidad	
Especificación de requerimientos	Comprobación de realismo	
	Verificabilidad	

b) Diseño de Software

"El proceso de la definición de la arquitectura, componentes, interfaces, y otras características del sistema o componentes".

- Las sub-áreas en que se divide son:
  - *Fundamentos de Diseño de Software*
  - *Aspectos clave en el Diseño de Software*
  - *Estructura de Software y Arquitectura*
  - *Análisis y Evaluación de la Calidad del Diseño*
  - *Notaciones del Diseño de Software*
  - *Estrategias del Diseño de Software*

c) Construcción de Software

"Se refiere a la creación detallada del trabajo, validación de software a través de una combinación de codificación, verificación, pruebas de unidad, pruebas de integración, y depuración".

- Las sub-áreas son:
  - Fundamentos de la Construcción de Software
  - Gestión de la Construcción
  - Consideraciones Practicas

d) Pruebas de Software

"Consiste de la verificación dinámica del comportamiento de un programa en un conjunto finito de casos de pruebas, apropiadamente seleccionado del dominio de ejecuciones finitas usuales, contra el comportamiento esperado".

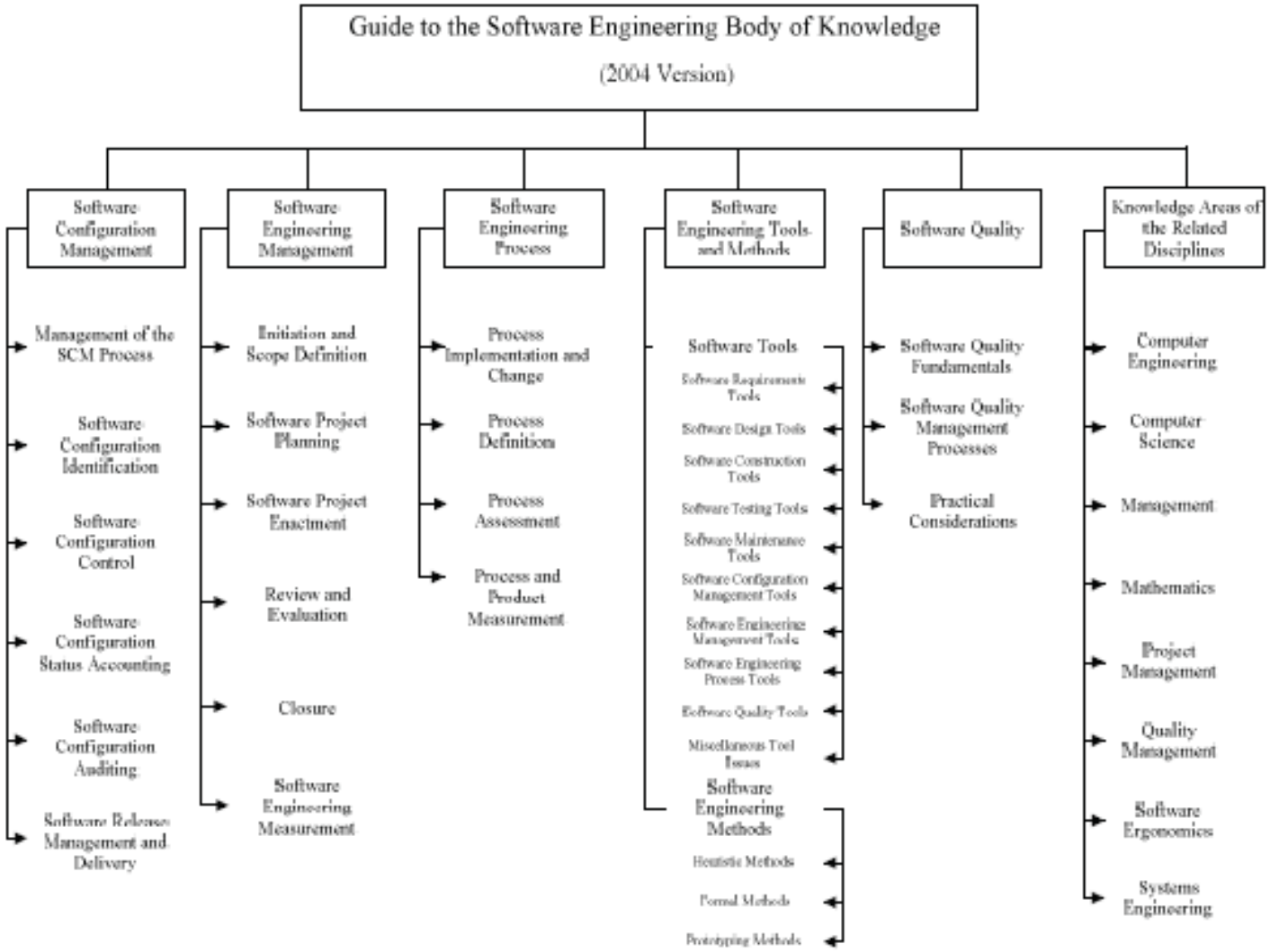
- Incluye cinco sub-áreas:
  - *Fundamentos de Pruebas de Software*
  - *Niveles de Prueba*
  - *Técnicas de Prueba*
  - *Mediciones relacionadas a las Pruebas*
  - *Proceso de Pruebas*

e) Mantenimiento de Software

"Una vez en operación, puede ocurrir que existan ciertas anomalías sin cubrir, cambios en los ambientes de operación, y surgimiento de requerimientos de nuevos usuarios".

- Se divide en cuatro sub-áreas:
  - *Fundamentos*
  - *Aspectos Clave*
  - *Proceso de Mantenimiento*
  - *Técnicas para el Mantenimiento*

Últimas Seis KAs



Procesos de Software

- Serie de actividades relacionadas que conducen a la elaboración de un producto de software.

- Las actividades de los procesos de software son:

  1.- Especificación del software

  2.- Diseño e Implementación del software

  3.- Validación del software

  4.- Evolución del Software

Procesos de Software

- Estos procesos pueden mejorarse con la estandarización de los mismos.

- Las ventajas de la estandarización son:
  - Reduce el tiempo de capacitación
  - Reduce costos
  - Mejora la comunicación

Modelado del Sistema

- Proceso para desarrollar modelos abstractos de un sistema.
- En cada modelo se presenta una visión o perspectiva diferente de dicho sistema.
- Representa un sistema usando algún tipo de notación gráfica (entre ellas el Lenguaje de Modelado Unificado, UML).

Modelado del Sistema

Diferentes perspectivas de los modelos para representar los modelos:

TIPO DE PERSPECTIVA	DESCRIPCIÓN
Externa	Modela el contexto o entorno del sistema
Interacción	Entre un sistema y su entorno, o entre otros componentes del sistema
Estructural	Modela la organización del sistema o la estructura de datos que procese el sistema
Comportamiento	Modela el comportamiento dinámico del sistema y cómo responde a ciertos eventos

Modelado del Sistema

- El estándar UML considera cinco tipos de diagramas para representar los esencial del sistema:

TIPO DE DIAGRAMA	DESCRIPCIÓN
Actividad	Actividades incluidas en un proceso o en el procesamiento de datos
Caso de uso	Interacciones entre un sistema y su entorno
Secuencias	Interacciones entre los actores y el sistema, y entre los componentes del sistema
Clase	Revela las clases de objeto entre el sistema y las asociaciones entre estas clases
Estado	Explica cómo reacciona el sistema frente a eventos internos y externos

Modelado del Sistema

Modelos de Contexto

- En esta primera etapa se decide sobre las fronteras del sistema.
- Determinar cuál funcionalidad se incluirá en el sistema y en el entorno del mismo.
- *Modelo arquitectónico simple.*
- Los sistemas externos pueden generar o consumen datos del sistema.

Modelado del Sistema

Modelos de interacción

- Implican entradas y salidas del usuario
- Interviene interacciones de algún tipo.
  - Entre el sistema a desarrollar y otros sistemas
  - Entre los componentes del sistema
- Diagrama y descripción tabular

Tipos de Modelos de Interacción	Descripción
Modelado de caso de uso	Modela interacciones entre un sistema y actores externos (usuarios u otros sistemas).
Diagramas de secuencia	Modela interacciones entre componentes del sistema e incluso agentes externos.

Modelado del Sistema

Modelos de estructurales

- Muestran la organización del sistema
- En términos de los componentes que constituyen dicho sistema y sus relaciones.
- *Estáticos*: Estructura del diseño del sistema
- *Dinámicos*: Organización del sistema cuando se ejecuta

Tipos de Modelos Estructurales	Descripción
Diagramas de clase	Modelos de sistema orientado a objetos para mostrar las clases de un sistema y sus asociaciones entre dichas clases.
Generalización	Gestiona la complejidad, generalizando los atributos .
Agregación	Define que un objeto (el todo) se compone de otros objetos (las partes).

Modelado del Sistema

Modelos de comportamiento

- Modelos dinámicos del sistema conforme se ejecuta (diagrama y tabular).
- Muestra lo que sucede o lo que se supone que pasa con el sistema cuando responde ante un estímulo de su entorno (datos/eventos).

Tipos de Modelos de Comportamiento	Descripción
Dirigido por datos	Exhiben la secuencia de acciones que ocurren desde una entrada a procesar hasta la salida correspondiente (respuesta al sistema).
Dirigido por eventos	Muestra cómo responde un sistema a eventos externos o internos.

Diseño arquitectónico

- Se interesa por entender cómo debe organizarse un sistema.
- Conocer cómo se tiene que diseñarse la estructura global de ese sistema.
- Modelo arquitectónico que describe la forma en que se organiza el sistema como un conjunto de componentes de comunicación.
- Vistas arquitectónicas
- Patrones arquitectónicos

Procesos de Software

- Serie de actividades relacionadas que conducen a la elaboración de un producto de software.

- Las actividades de los procesos de software son:

  1.- Especificación del software

  2.- Diseño e Implementación del software

  3.- Validación del software

  4.- Evolución del Software

Procesos de Software

- Estos procesos pueden mejorarse con la estandarización de los mismos.

- Las ventajas de la estandarización son:
  - Reduce el tiempo de capacitación
  - Reduce costos
  - Mejora la comunicación