



UNIDAD 1

Ingeniería del Software

Introducción



Ingeniería De Software

Algunas definiciones ...

IEEE

La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software.

SEI (Software Engineering Institute)

***Ingeniería** es la aplicación sistemática del conocimiento científico en la creación y construcción de soluciones (“cost-effective”) para resolver problemas prácticos al servicio del hombre.*

***Ingeniería del SW** es aquella parte de la ingeniería que aplica los principios de las ciencias de la computación y las matemática para alcanzar soluciones (“cost-effective”) a problemas de software.*

IEEE

Es la disciplina tecnológica y de administración que se ocupa de la producción sistemática & mantenimiento de productos de software que son desarrollados en tiempo y costo estimados.

Body of Knowledge (BoK)

Un “cuerpo de conocimiento” es un requisito para calificar a cualquier área de la ingeniería como una profesión y describe el conocimiento relevante para una disciplina.

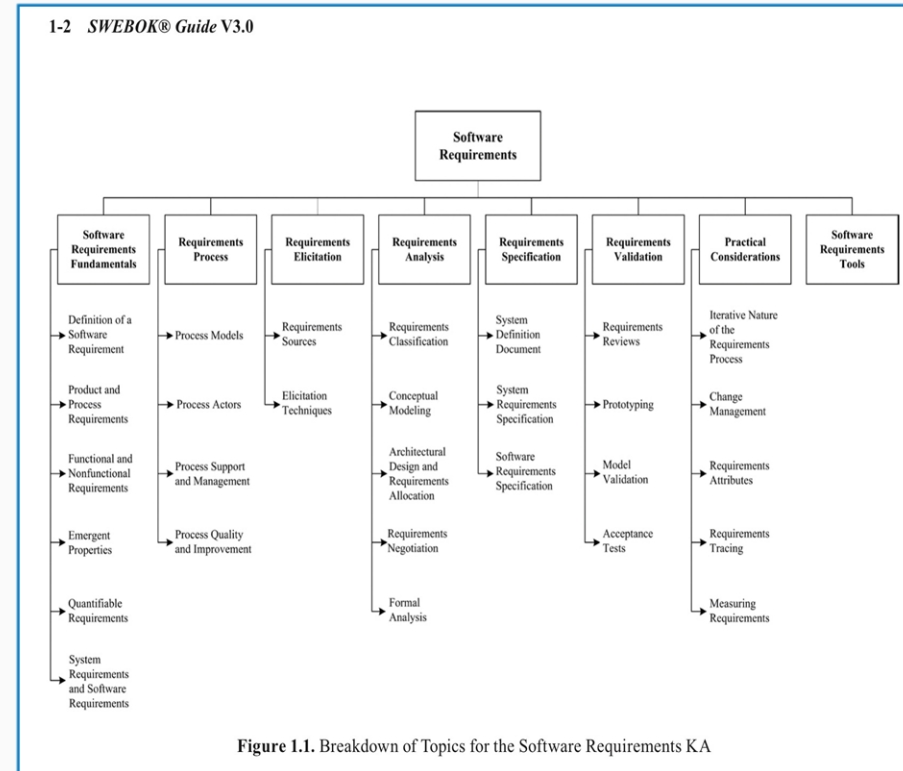
SWEBoK (Software Engineering Body of Knowledge)

- *Describe el conocimiento que existe de la Ingeniería de SW.*
- *Comenzó 1998, liderado por la IEEE Computer Society, para “convertir a la ingeniería del software en una disciplina legítima y una profesión reconocida”.*
- *Está conformado por 15 Áreas de Conocimiento (Knowledge Areas – KA) que sintetizan conceptos básicos y referencian a información mas detallada.*
- *La versión actual es SWEBoK 3.0 (publicada en el 2014).*
- *Tiene reconocimiento internacional como ISO/IEC Technical Report 19759:2015.*

Body of Knowledge (BoK)

The 15 SWEBOK KAs

1. *Software Requirements*
2. *Software Design*
3. *Software Construction*
4. *Software Testing*
5. *Software Maintenance*
6. *Software Configuration Management*
7. *Software Engineering Management*
8. *Software Engineering Process*
9. *Software Engineering Models and Methods*
10. *Software Quality*
11. *Software Engineering Professional Practice*
12. *Software Engineering Economics*
13. *Computing Foundations*
14. *Mathematical Foundations*
15. *Engineering Foundations*





Problemas habituales en el D&M de Software

¿Cuáles son los problemas habituales en el D&M de SW?

Los proyectos se terminan con mucho atraso

A veces ni se terminan ...

- *Se congelan / suspenden / postergan*
- *Se postergan*
- *Quedan ahí ...mueren...*
- *Aparición del “Proyecto Fase II” (incompletitud) o “Reingeniería de ...” (deuda técnica)*

Los proyectos se exceden en el costo estimado

El producto final del proyecto no cumple con las expectativas del cliente

No hace lo que se supone debería hacer

El producto final no cumple con los reqs. de calidad mínimos esperados

Además de lo funcional, temas relacionados con performance, seguridad, escalabilidad, etc....

¿Cuáles son las causas comunes?

No hay administración y control de proyectos

- *No se puede controlar lo que no se mide*

El éxito depende de los gurúes

Se confunde la solución con los requerimientos

- *La tecnología no es la solución en si misma*

Las estimaciones son empíricas y no hay historia sobre proyectos realizados

La calidad es una noción netamente subjetiva que no se puede medir

No se consideran los riesgos

Se asumen compromisos imposibles de cumplir

Las actividades de mantenimiento van degradando el software

Se comienzan los proyectos con requerimientos no claros ni estables

- *Existe una tendencia a ser optimista (simplificando, subestimando, ...)*

Y la lista sigue

Y las reacciones mas frecuentes ...

Buscando al gurú que nos saque del problema

- *A veces funciona*

Agregando gente a un proyecto que está atrasado

Recortando / eliminando cualquier actividad que genere documentación

Recortando / eliminando la etapa de Testing

Recortando / eliminando cualquier actividad que no sea codificar

Asumiendo definiciones en lugar del usuario

Recurriendo a una “bala de plata”

- *Un nuevo paradigma / Una nueva herramienta / Un nuevo lenguaje*

Justificando con frases típicas como:

- *“ ... Administrar SW es diferente ... ”*
- *“ ... En definitiva, el usuario no sabe lo que quiere ... ”*
- *“ ... Lo que ocurre que la tecnología tiene que madurar ... ”*
- *“ ... En realidad ya casi está, está en un 90% ... ”*

.....

Y las reacciones mas frecuentes ...

Buscando al gurú que nos saque del problema

- *A veces funciona*

Agregando gente a un proyecto que está atrasado

Recortando / eliminando cualquier actividad que genere documentación

Recortando / eliminando la etapa de Testing

Recortando / eliminando cualquier actividad que no sea codificar

Asumiendo definiciones en lugar del usuario

Recurriendo a una “bala de plata”

- *Un nuevo paradigma / Una nueva herramienta / Un nuevo lenguaje*

Justificando con frases típicas como:

- *“ ... Administrar SW es diferente ... ”*
- *“ ... En definitiva, el usuario no sabe lo que quiere ... ”*
- *“ ... Lo que ocurre que la tecnología tiene que madurar ... ”*
- *“ ... En realidad ya casi está, está en un 90% ... ”*

.....

Análisis de la Situación

¿ Por qué seguimos padeciendo los mismos problemas y cometiendo los mismos errores?

¿ Será porque ...

- ... *nos cuesta reconocer que tenemos un problema ?*

Se piensa que después de haber tenido (relativo) éxito con algunos proyectos, se puede en consecuencia llevar adelante cualquier proyecto

- ... *no compartimos conocimientos ?*

- ... *rechazamos la complejidad ?*

- ... *no admitimos que tenemos que aprender ?*

No se puede conocer todo !!!!

- ... *no nos gusta que nos “critiquen” (constructivamente) ?*

Relación Proyecto SW <-> Persona

- ... *menospreciamos a las tareas no técnicas ?*

Solo se respeta a aquellos que conocen mucho de Tecnología ??

- ... *pensamos que bajo presión existen soluciones mágicas que por el solo hecho de presionar hace que las cosas sucedan ?*

- ... *asumimos que producir SW es simplemente así ... ?*



Diagrama Causa-Efecto

¿Qué es? ¿Para que se usa?

Es la representación de varios elementos (causas) de un sistema que pueden contribuir a un problema (efecto)

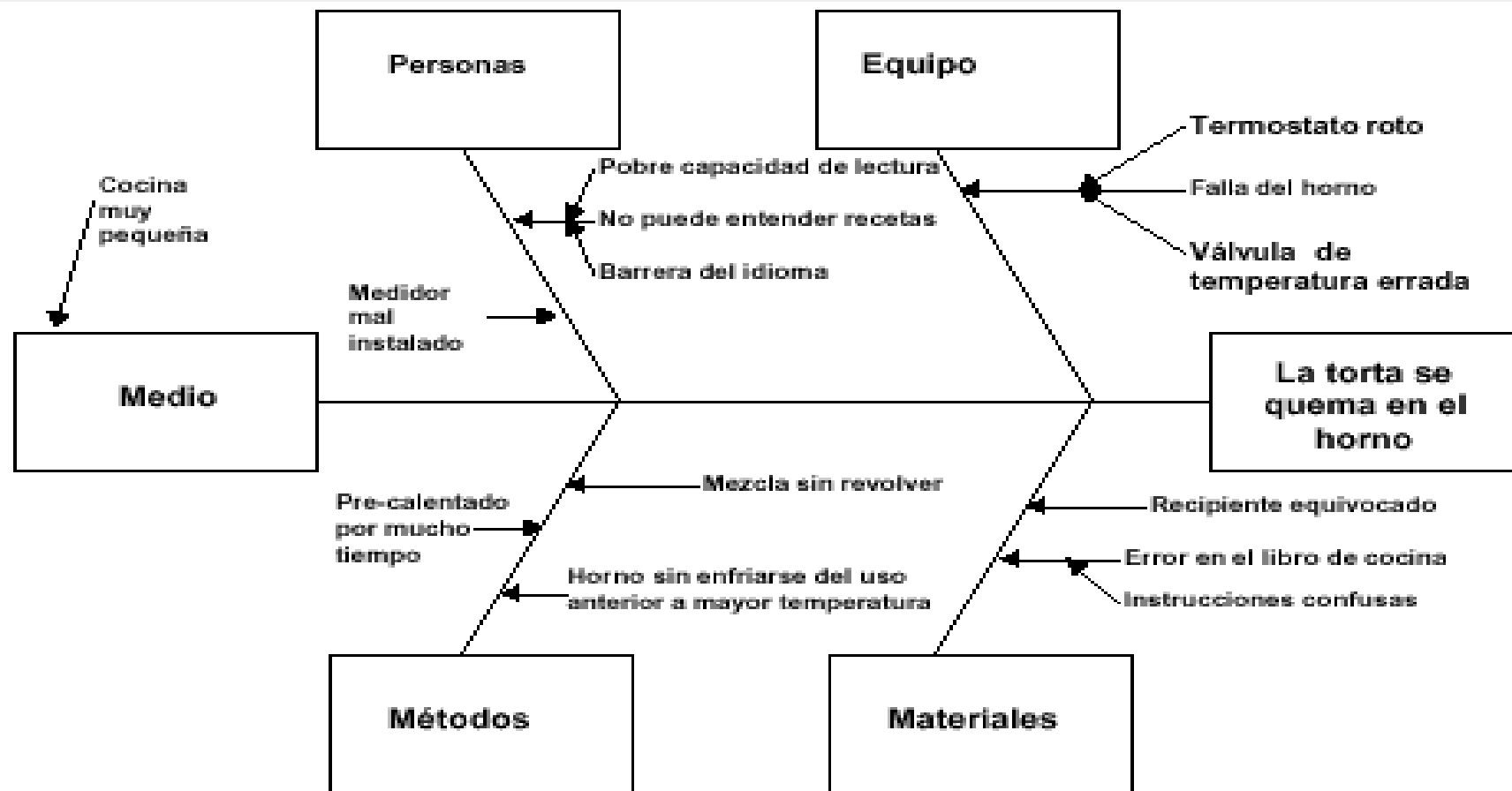
Es útil para la recolección de datos y es efectivo para estudiar los procesos y situaciones

- *Es utilizado para identificar las posibles causas de un problema específico. La naturaleza gráfica del diagrama permite que los grupos organicen grandes cantidades de información sobre el problema y determinar exactamente las posibles causas. Aumenta la probabilidad de identificar las causas principales*

Un diagrama causa y efecto bien preparado es un vehículo para ayudar a los equipos de mejora continua a tener una concepción común de un problema complejo

También conocido por Diag. De Ishikawa o Espina de Pescado

¿Qué es? ¿Para qué se usa?





Preguntas