

<u>Teoría 1: Repaso, comunicación y SRS.....</u>	<u>1</u>
<u>Proceso de software.....</u>	<u>1</u>
<u>Modelo.....</u>	<u>1</u>
<u>Comunicación.....</u>	<u>2</u>
<u>Elicitación de requisitos.....</u>	<u>2</u>
<u>Muestreo de la documentación, los formularios y los datos existentes.....</u>	<u>2</u>
<u>Observación del ambiente de trabajo.....</u>	<u>2</u>
<u>Visitas al sitio.....</u>	<u>3</u>
<u>Cuestionarios.....</u>	<u>3</u>
<u>Entrevista.....</u>	<u>3</u>
<u>JRP.....</u>	<u>4</u>
<u>Brainstorming.....</u>	<u>4</u>
<u>Requerimientos SRS.....</u>	<u>4</u>
<u>Requerimientos.....</u>	<u>4</u>
<u>ConOps (IEEE Std. 1362-1998).....</u>	<u>5</u>
<u>SRS (standard requirements specifications) IEEE Std. 830-1998.....</u>	<u>5</u>
<u>Sección 1 del SRS.....</u>	<u>6</u>
<u>Sección 2 del SRS.....</u>	<u>6</u>
<u>Sección 3 del SRS.....</u>	<u>7</u>
<u>Sección 4 del SRS.....</u>	<u>7</u>
<u>Sección 5 del SRS.....</u>	<u>7</u>

Teoría 1: Repaso, comunicación y SRS

Proceso de software

Un proceso de software es un conjunto de actividades y resultados asociados que producen un producto de software.

Estos fueron vistos el año pasado (la etapa elicitación y especificación de requerimientos). Específicamente nosotros usamos SCRUM.

Modelo

Representación abstracta de un proceso de software.

Los modelos de proceso de software cuentan con actividades fundamentales (se presentan en todos los modelos):

1. Especificación del software.
 - a. Técnicas de elicitación.
 - b. Especificación de requerimientos.
2. Desarrollo del software.
3. Validación del software.
4. Evolución del software.

Comunicación

La comunicación es el proceso más importante de la interacción humana, que se produce de forma verbal y no verbal.

Elicitación de requisitos

Proceso que trata de adquirir toda la información relevante para poder producir un modelo de los requerimientos de un dominio de problema.

Objetivos:

1. Conocer el dominio del problema para facilitar comunicación con clientes y usuarios y entender sus necesidades.
2. Conocer el sistema actual.
3. Identificar necesidades (explícitas e implícitas) de clientes, usuarios y las expectativas que tienen del sistema a desarrollar.

Muestreo de la documentación, los formularios y los datos existentes

Es una técnica de elicitación que recolecta hechos a partir de documentación preexistente. Por ejemplo de:

1. Organigrama (identificación de propietario, usuarios, claves).
2. Memos, notas internas, minutas, registros contables.
3. Solicitudes de proyectos de sistemas informáticos previos.

Permite conocer el historial que origina el proyecto.

Útil cuando el sistema está en funcionamiento y tenemos acceso al mismo. (Por ejemplo si necesitamos crear una transición entre el sistema nuevo y el actual mediante capacitación).

Observación del ambiente de trabajo

Técnica en la cual el analista se convierte en observador de cómo los trabajadores y sistemas interactúan de forma natural.

Lineamientos de la observación:

1. Determinar quién será observado y cuando.
2. Determinar si pido permiso al observado y explicó la necesidad de la observación, o no (avisar puede intimidar al observado, y conviene hacer una sin avisar y otra si).
3. Mantener bajo perfil (evitar intimidar al trabajador puesto que no actuara como suele trabajar en realidad).
4. Tomar nota.
5. Revisar notas con alguien apropiado.
6. No interrumpir.

Visitas al sitio

1. Investigar el dominio del problema.
2. Patrones de soluciones (mismo problema en otra organización).
3. Revistas especializadas.
4. Buscar problemas similares en internet.
5. Consultar otras organizaciones.

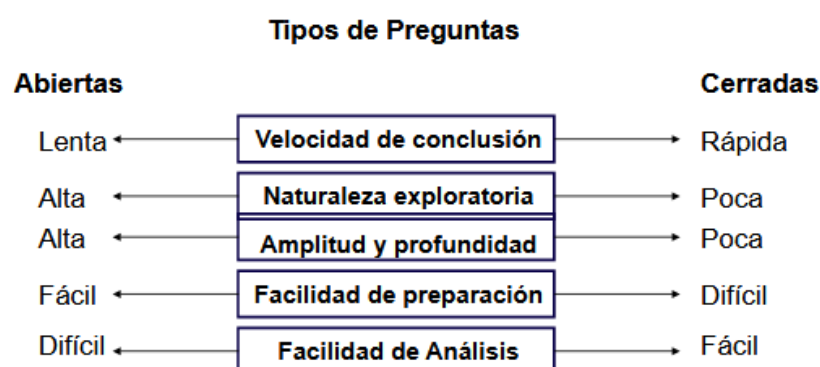
Extracto del video que explica para qué sirve esta técnica:

"Supóngase que es el software para una fabrica, si, yo pienso y desarrollo el software pero después ¿las máquinas donde van a estar?, ¿donde va a estar el operador para cargar los datos? ... ¿puedo poner la computadora en cualquier lado? ¿Voy a estar cerca del lugar que necesito?"

Cuestionarios

2 tipos de preguntas, abiertas y cerradas

Cerradas son las más fáciles de concluir y analizar pero no tienen mucha naturaleza exploratoria ni son amplias o profundas. Las abiertas son lo contrario.



Está bueno tener ambas, lo que está cerrado y categorizado lo hago cerrado y sin lo hago abierto para la próxima.

Sirven para cuando queremos preguntar a muchas personas y también cuando están separadas geográficamente, por lo que en ambos casos sería costoso encontrarse en persona con todos.

Entrevista

Para preparar una entrevista debemos:

1. Leer los **antecedentes**.
2. Establecer los **objetivos** de la misma.
3. **Seleccionar los entrevistados**.
4. **Planificar entrevista** (fecha, hora lugar, duración)
5. **Selección de tipo de preguntas** y su **estructura**. Definir guión, con lenguaje claro y sin opiniones en las preguntas, que sean concisas, y evitar que sean largas y complejas.
Claramente: Si sabemos poco vamos a tener que explorar un poco más, sino alcanza una sola entrevista habrá que solicitar otra.

3 tipos de organización de entrevistas:

1. Piramidal (inductiva, cerradas a abiertas).
2. Embudo(deductiva, abiertas a cerradas).
3. Diamante(combinación, cerradas a abiertas a cerradas).

El profe recomienda diamante si falta calle.

JRP

Requiere de una instalación física, más compleja de llevar a cabo al tener que reunir mucha gente. Tiene de bueno que al tener todos los actores involucrados en un mismo lugar se genera un ambiente que permite resolver muchas dudas y discrepancias entre los requerimientos de una.

Brainstorming

Consiste en que todos los participantes digan todas las ideas que tengan sobre un cierto tema para luego revisarlas para quedarse con las mejores. Promueve el desarrollo de ideas creativas.

Requerimientos SRS

Requerimientos

Los requerimientos pasan por 4 etapas:

1. Solicitud
2. Definición
3. Análisis
4. Especificación (acá usamos HU u otro visto por ejemplo).

Tipos de requerimientos:

- Funcionales: Definen el comportamiento del sistema, describen las tareas que el sistema debe realizar, es importante que se defina

manteniendo un equilibrio entre generalidad y exceso de detalle al pedo.

- No funcionales: Cosas fuera del sistema, definen aspectos, deseables desde el punto de vista del usuario. O también restricciones como tiempos de respuesta, usabilidad, mantenimiento, etc.

ConOps (IEEE Std. 1362-1998)

Estándar de documento para describir características del sistema escrito desde el punto de vista del usuario, dirigido para el usuario.

Esta descripción es la forma en que se puede comunicar la visión general, cualitativa y cuantitativa de las características del sistema entre cliente y desarrollador.

Ofrece formato específico para ser completado, no especifica técnica exacta sino que proporciona guías para poder hacerlo, identifica elementos que al menos deben estar.

SRS (standard requirements specifications) IEEE Std. 830-1998

Un SRS especifica funciones de un producto de software, programa o conjuntos de programa en un entorno. El documento de especificación de requisitos puede desarrollarlo personal desarrollador o de la parte cliente (conviene ambos).

Características de SRS:

1. Alcance: Brindar una colección de buenas prácticas para escribir especificaciones. Se describen los contenidos y las cualidades de una buena especificación de requerimientos.
2. Naturaleza del SRS: El SRS es una especificación de un producto de software en particular, escrito por uno o más personas (desarrolladores o clientes).
3. Ambiente del SRS: El software puede contener toda la funcionalidad del proyecto o ser parte de un sistema más grande. En el último caso habrá un SRS que declarará las interfaces entre el sistema y su software desarrollado y pondrá qué función externa y requerimientos de funcionalidad tiene con el software desarrollado.
4. Correcto: Un SRS es correcto si cada requisito declarado se encuentra en el software.
5. No ambiguo: Un SRS es inequívoco si y sólo si cada requisito declarado tiene solo una interpretación.
6. Completo: Reconoce todos los requisitos externos impuestos por especificaciones del sistema
7. Consistente: Tiene que estar de acuerdo con los documentos de nivel superior (como una especificación de requerimientos).

8. Priorizado: Es priorizado por la importancia de sus requerimientos particulares.
9. Comprobable: Es comprobable si sus requisitos declarados son comprobables (no ambiguos).
 - a. Requisito comprobable: Es comprobable si hay algún proceso para verificar que el producto de software cumple el requisito.
10. Modificable: Su estructura y estilo permite cambios a los requerimientos fácilmente y consistentemente conservando la estructura y estilo.
11. Trazabilidad: Claridad del origen de cada requerimiento y su trazabilidad hacia los requerimientos futuros.
12. Preparación conjunta del SRS: Preparar en conjunto con las partes involucradas y así formar un buen acuerdo.
13. Evolución de SRS: Debe evolucionar junto al software, registrar cambios y responsables y aceptar los mismos.
14. Prototipos: Usa prototipos para la definición de requerimientos.
15. Diseño incorporado en el SRS: Puede incorporar atributos o funciones externas como diseños para interactuar entre subsistemas.
16. Requerimientos incorporados en el SRS: Los detalles particulares son anexados como documentos externos.

Sección 1 del SRS

- Propósito: Define el propósito del documento y se especifica a quien va dirigido.
- Alcance o ámbito del sistema: Nombra al sistema, explica que hace y no hace el sistema, describe los beneficios, objetivos y metas futuras.
- Referencias: Lista completa de referencias de los documentos usados para escribir el SRS. (título, número de reporte, fecha y publicación de cada documento y las fuentes de donde se obtuvieron).

Sección 2 del SRS

Factores generales que afectan al producto y sus requerimientos

- Perspectiva del producto: Se declara si el producto es independiente y totalmente autónomo. Si es parte de un sistema más grande se relacionan los requerimientos de este a la funcionalidad del software e identificar las interfaces entre ese sistema y el software.
- Funciones del sistema: Resumen de las funciones del sistema organizadas y puede usar gráficos para las relaciones entre las mismas.
- Características del usuario: Características generales de los usuarios del producto (nivel educativo, experiencia y especialización técnica).

- Evoluciones previsibles del sistema: Requerimientos a implementar en el futuro.

Sección 3 del SRS

Requerimientos no funcionales del software a detalle para el diseño del sistema y que se pueda probar que satisface estos requerimientos

- Requerimientos de rendimiento: Relacionados a la carga esperada para el sistema. Debe ser medible. Por ejemplo, tiempo esperado por transacción (por ejemplo, que todas tarden un 1 segundo)
- Seguridad: Relacionados a la protección de accesos, usos, sabotajes, modificaciones y destrucciones maliciosas o accidentales. Por ejemplo, logs de actividad
- Portabilidad: Relacionados a la facilitación de traslado a otras plataformas o entornos. Por ejemplo, el uso de un determinado lenguaje por su portabilidad.

Sección 4 del SRS

Identifica el tipo de mantenimiento necesario para el sistema y quien realiza estas tareas junto a cada cuanto se deben de hacer. Por ejemplo, generación de estadísticas de acceso semanal por el desarrollador.

Sección 5 del SRS

Información relevante que no forma parte del SRS. Por ejemplo, casos de uso.