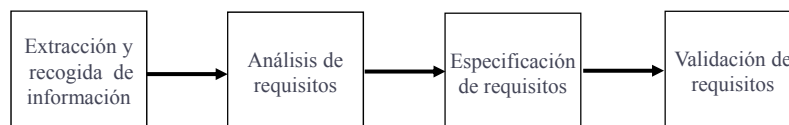


Definición de requisitos

Definición de requisitos

- Definir requisitos del software
 - tarea iterativa, buena ERS (Especificación de Requisitos de Software), técnicas de especificación
- Definir requisitos de interfaces
 - crítico para usuario, interacción, facilidad de uso
- Integrar los requisitos y asignar prioridades

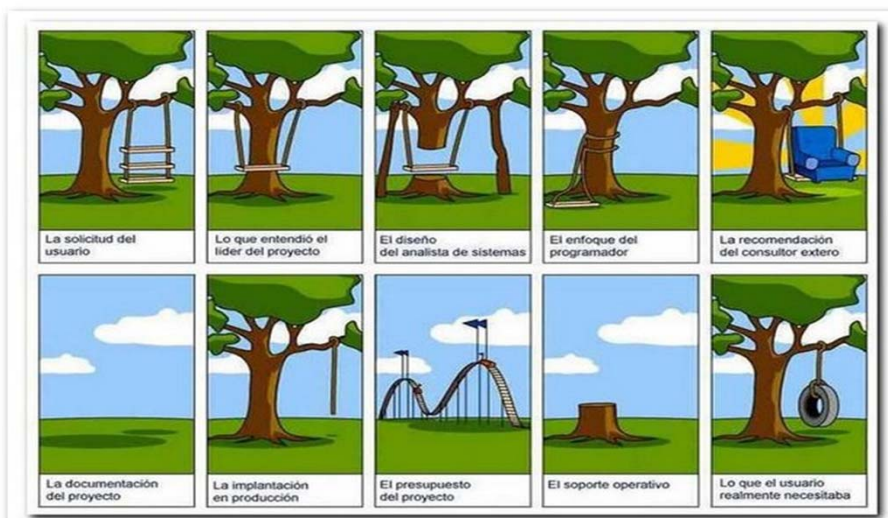


Proceso

- **Extracción o determinación de requisitos.** Proceso mediante el cual los clientes o futuros usuarios del software descubren, revelan, articulan y comprenden los requisitos que desean.
- **Análisis de requisitos.** Proceso de razonamiento sobre los requisitos obtenidos en la etapa anterior, detectando y resolviendo posibles inconsistencias o conflictos, coordinando los requisitos relacionados entre sí, etc.
- **Especificación de requisitos.** Proceso de redacción o registro de los requisitos. Suele recurrirse a un lenguaje natural, lenguajes formales, modelos, gráficos, etc.
- **Validación de los requisitos.** Confirmación, por parte del usuario o el cliente de que los requisitos especificados son válidos, consistentes, completos, etc.

Aunque estas actividades no tienen por qué realizarse en secuencia, ya que hay muchas iteraciones y solapamientos entre ellas, sí marcan un proceso general para la fase de análisis.

¿Una mala captura de requisitos?



Técnicas de recogida de información

METRICA V3 cuenta con guías para algunas de estas técnicas

- **Entrevistas**
- **Desarrollo conjunto de aplicaciones (JAD)**
- **ETHICS**
- **Prototipado**
- **Observación**
- **Cuestionarios**
- **Estudio de documentación**
- **Tormenta de ideas (brainstorming)**



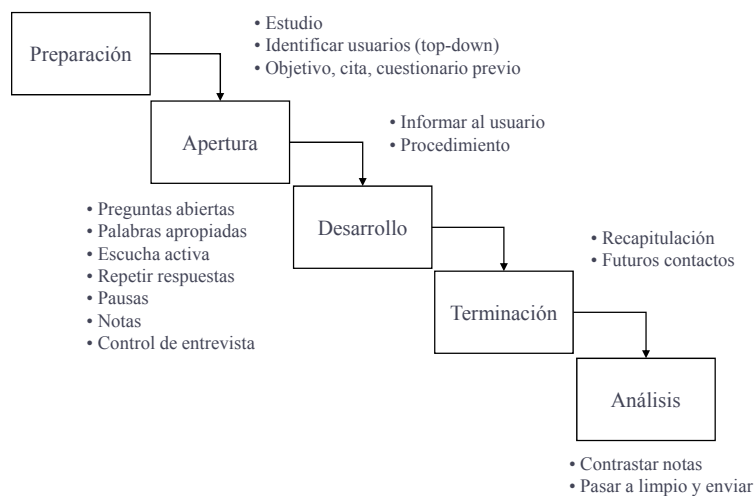
Cuestionarios

- Para usuarios dispersos, numerosos, poco disponibles, etc.
 - Distinto de encuesta: no hay entrevistador presente
- Algunas indicaciones de diseño:
 - Tener claros objetivos de información que deseamos obtener
 - Son limitados por la naturaleza de la técnica
 - Preguntas muy claras y precisas (no presentes para aclarar)
 - Una sola cuestión en cada pregunta
 - Cortos y sencillos (evitar aburrimiento)
 - Preguntas cerradas con unas pocas abiertas
 - Escalas Likert para opinión: diseñar bien la escala
 - Introducción con propósito y que agradezca colaboración
 - Alguna pregunta final de sugerencias, olvidos, etc.

Entrevistas - Resumen

- Entrevistas
 - Obtener información de forma individual
 - Preparar un guion previo que se remite al entrevistado
 - Obtener la máxima información sin provocar rechazo
 - Al final resumir las conclusiones para aclarar malentendidos
 - Enviar el acta al entrevistado para fijar ideas por escrito

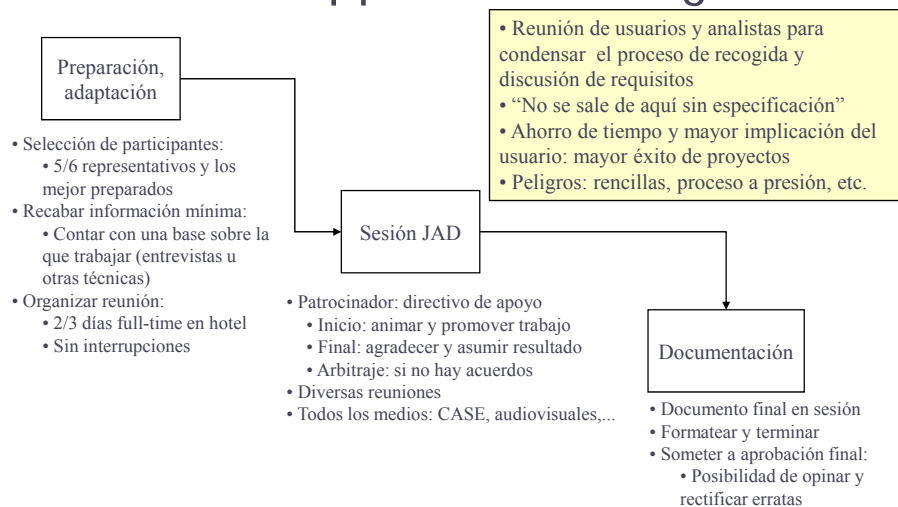
Entrevistas



Reuniones

- Procedimiento
 - Obtener información dispersa entre varios usuarios, comunicar información, tomar decisiones
 - Preparar el orden del día y convocar la reunión
 - Al inicio resumir objetivos y método de trabajo
 - Importancia de la persona que dirige la reunión
 - Al final resumir las conclusiones para aclarar malentendidos, destacar puntos pendientes y fijar siguiente reunión o actividades/responsables
 - Enviar el acta a los participantes para fijar ideas por escrito
- Diversos tipos

JAD: Joint Application Design



JAD – Proceso detallado (I)

- JAD (Joint Application Design)
 - Reducir el tiempo de desarrollo de un sistema, manteniendo su calidad, involucrando a los usuarios a lo largo de todo el desarrollo del sistemas
 - Pocas reuniones, de larga duración y bien preparadas
 - Durante la reunión elaborar y aprobar modelos (puede ser directamente en una herramienta CASE)
 - Actividades:
 - Inicio: se define el ámbito y la estructura del proyecto, los productos a obtener, el material necesario y los participantes.
 - Desarrollo: se identifican los modelos a obtener, se obtienen y se consensuan.
 - Finalización: se validan, completan e integran los modelos para obtener los productos de salida.

JAD – Proceso detallado (II)

- Fijar los componentes del equipo, su perfil y responsabilidades:
 - Moderador (líder Jad) con amplios conocimientos de la metodología de trabajo, dinámica de grupos, psicología del comportamiento, así como de los procesos de la organización objeto del estudio.
 - Promotor, persona que ha impulsado el desarrollo.
 - Jefe de proyecto, responsable de la implantación del proyecto.
 - Especialista en modelización, responsable de la elaboración de los modelos en el transcurso de la sesión.
 - Desarrolladores, aseguran que los modelos son correctos y responden a los requisitos especificados.
 - Usuarios, responsables de definir los requisitos del sistema y validarlos.

JRP (Joint Requirements Planning)

- Potenciar la participación activa de la alta dirección. Muy útil en el desarrollo del Plan Estratégico de SI
- Las características de las sesiones JRP y JAD son comunes en cuanto a la dinámica del desarrollo de las sesiones y la obtención de los modelos con el soporte de las herramientas adecuadas.
- Diferencias: nivel más alto en la organización en cuanto a visión global del negocio y capacidad de decisión. Tipo de información de salida :
 - Modelos de procesos de la organización.
 - Modelo de información.
 - Modelo de sistemas de información, etc.

JRP – Proceso detallado (I)

- **Perfiles:**
 - Moderador (líder JRP), gran capacidad de relación, habilidades de negociación y de gestión de dinámica de grupos y conocimiento de los procesos de la organización.
 - Promotor, persona que ha impulsado el proyecto.
 - Director de proyecto, responsable del proyecto.
 - Consultores, responsable de traducir los requisitos, de tal forma, que los usuarios puedan entender y discutir los resultados.
 - Especialista en modelización, responsable de la elaboración de los modelos .
 - Usuarios de alto nivel, responsables de definir los procesos de la organización y las prioridades de implantación.

JRP – Proceso detallado (II)

▫ Actividades:

- Iniciación, se establece el alcance del proyecto, los procesos de negocio implicados, las unidades organizativas afectadas y los perfiles del equipo JRP.
- Búsqueda, se identifican los objetivos, se estudia la situación actual y se busca la información relevante, que pueda ser útil.
- Preparación, se seleccionan los participantes, se prepara el material necesario y la agenda de JRP.
- Realización: se introduce la reunión, los objetivos y se elaboran los productos objeto de la sesión.
- Finalización: se completan los productos y se presenta a los participantes que corresponda.

Prototipado

- Creación de “maquetas” de sistemas
 - para evaluación de usuario y/o desarrolladores
 - según objetivo, requiere herramientas distintas (Office, entornos visuales, programación, etc.)
 - coste/esfuerzo alto (¿10% de proyecto?)
- Aplicación:
 - área poco definida: dificultad o sin tradición
 - coste alto de rechazo
 - evaluar impacto previamente
- Tipos:
 - interfaz de usuario: más habitual
 - modelos de rendimiento (no en análisis de requisitos)
 - prototipado funcional, evolutivo
 - reaprovecha código (ciclo de vida)



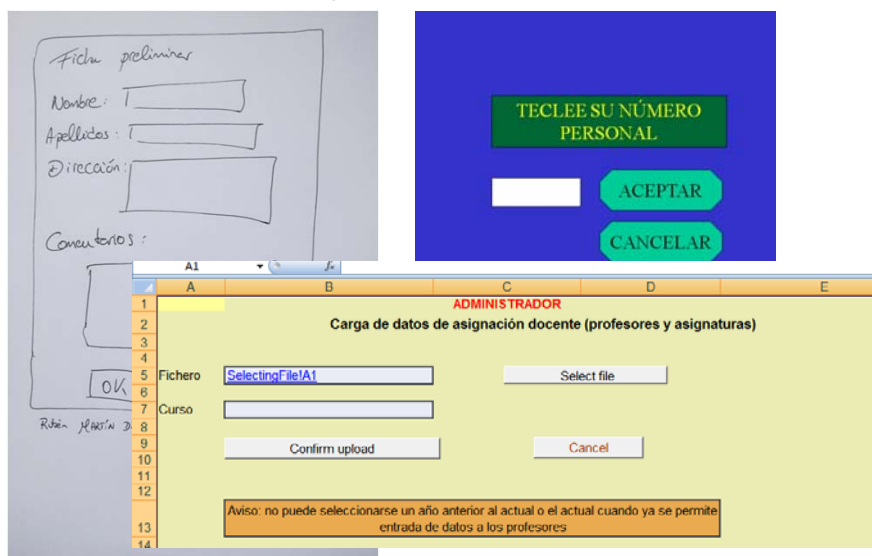
Prototipado y métodos afines

- Maquetas (Mock Ups)
 - Visión simplificada de la aplicación a desarrollar
 - Muestra para el usuario de como quedará y a que se parecerá la interfaz del software sin haber programado aplicación ni funcionalidad
 - Representar la interfaz gráfica del sistema con trazos sobre papel de la interfaz o interfaces huecas con herramienta visual.
- Storyboards*
 - Se encargan de presentar secuencia de interfaz gráfica con las actividades del sistema
 - Papel o herramientas como Visio, Excel, PowerPoint,...

*

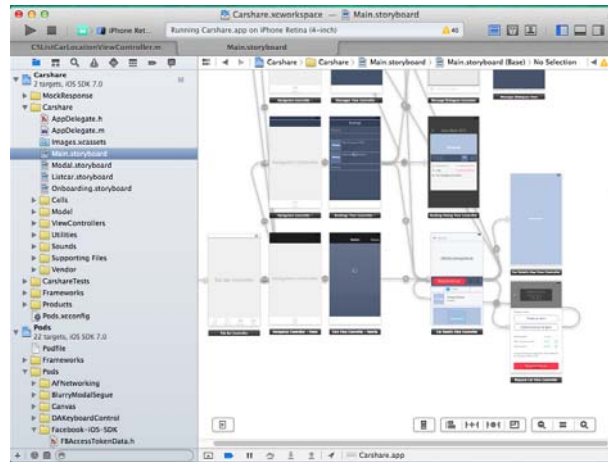
[Guía para la creación de storyboards en el diseño de aplicaciones:](#)

Ejemplos (I)



Ejemplos (II)

Tutorial para crear interfaces usando storyboard para desarrolladores iOS (Apple)



Otras técnicas (I)

- Observación:
 - Captar mejor la realidad del trabajo diario
 - “Pasar una mañana en el puesto”, “pasar por cliente”, *mystery shopping*, etc.
- Estudio de documentación:
 - Siempre necesario: no depende de tener interlocutor
 - Imprescindible como fase previa de otras técnicas
 - Análisis: procesos formales/documentados, normas, impresos, formularios, legislación, etc.
 - Mejor tener siempre muestras de documentos rellenos: apreciar uso real



Otras técnicas (II)

- **ETHICS y enfoques globales similares:**
 - **ETHICS: *Effective Technical and Human Implementation of Computer-based Systems*** (Enid Mumford). Un sistema no tiene éxito si no se ajusta a factores de organización, sociales y ergonómicos.
 - El trabajo ideal se compara con el trabajo hecho actualmente y el nivel de satisfacción del staff. Esta comparación lleva a identificar las necesidades a ser cambiadas y su porqué.
 - Satisfacción de empleados: mayor productividad
 - También BPR (Business Process Reengineering) y otras técnicas de procesos y métodos de trabajo
 - Enfoque global: análisis de organización → req. de software
- **Brainstorming:**
 - Crear nuevas ideas cuando no hay tradición del problema o de software existente
 - Técnica creativa muy usada en publicidad, etc.
 - Ronda de ideas en grupo sin evaluar su bondad
 - Al parar, leer lo dicho previamente y empezar otra ronda

Dificultad de comunicación



Resistencia al cambio

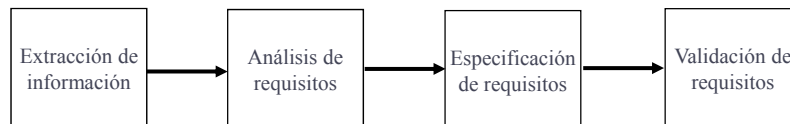
- **Ser humano: animal de costumbres**
 - Se resiste incluso a cambios buenos
- **Causas:**
 - Resistencia al personal informático
 - Equipos conjuntos, coordinación de directivos, comprensión de informáticos, formación de usuarios
 - Percepción de que “el proyecto no es bueno”
 - Involucrar en proyecto, análisis de coste/beneficio
 - No percibir necesidad de cambio (de forma de trabajo, de sistema, etc.)
 - “Si no se rompe, no lo toques”, involucrar, análisis coste/beneficio
 - Miedo a perder poder o puesto. Computerfobia: miedo tecnológico
 - Formación, integración en proyecto
 - Sistemas mal diseñados para usuarios (falta de “usabilidad”)

Resistencia a los informáticos



Definición de requisitos

- Definir requisitos del software
 - tarea iterativa, buena ERS, técnicas de especificación
- Definir requisitos de interfaces
 - crítico para usuario, interacción, facilidad de uso
- Integrar los requisitos y asignar prioridades



Análisis de requisitos

- Completitud.
 - No hay omisiones: no faltan requisitos (propiedad global) y no faltan detalles en la especificación de cada requisito (propiedad individual).
 - Difícil de determinar: contrastar con el cliente, comparar con proyectos semejantes, buscar la visión de conjunto, detectar huecos o partes infra-especificadas,...
- Detección de Conflictos e Inconsistencias
 - Los requisitos se agrupan por categorías y se organizan en sub-conjuntos, se estudia cada requisito en relación con el resto y se clasifican en base a las necesidades de los clientes/usuarios.
 - Es corriente en clientes y usuarios solicitar más de lo que puede realizarse o proponer requisitos contradictorios, argumentando que su versión es esencial por necesidades especiales.

Especificación de requisitos del software

- Especificación de Requisitos del Software (ERS)
 - documentación de los requisitos esenciales del software y sus interfaces externas
- Características:
 - Información veraz y completa
 - Comunicación eficaz, ser útil para técnicos y usuarios
- ERS debe definir el qué, no el cómo ni el cuándo
 - Describir todos los requisitos pero no más
 - No describir detalles de diseño, proyecto, verificación, aunque hay que ser precisos en los no funcionales

Preguntas planteadas en una ERS

- La funcionalidad.
 - ¿Qué tiene que hacer el software ?
- Las interfaces externas.
 - ¿Cómo el software actúa recíprocamente con las personas, el hardware y otros sistemas hardware o software?
- La actuación.
 - ¿Cuál es la velocidad, la disponibilidad, tiempo de respuesta, tiempo de la recuperación, etc.?
- Los atributos de calidad.
 - ¿Que fiabilidad, mantenibilidad, portabilidad, seguridad, etc. necesita el sistema?
- Las restricciones del diseño.
 - ¿Hay alguna restricción de idioma, hardware, recursos, etc.?

Características de una buena ERS

- IEEE Std. 830:
 - No ambigua
 - Completa
 - Fácil de verificar
 - Consistente
 - Fácil de modificar
 - Ordenación por prioridades
 - Facilidad para identificar fuente y efectos de cada requisito (trazabilidad)
 - Facilidad de uso en explotación y mantenimiento



Requisitos funcionales y no funcionales

- Requisito funcional (IEEE std. 610):
 - Requisito que especifica una función que un sistema o componente debe ser capaz de realizar
 - Ejemplos de un sistema de gestión de notas:
 - Obtener estadísticas de notas
 - Introducir nota de prácticas
 - Gestionar datos anagráficos del alumno
- Requisito no funcional:
 - Requisito que especifica una característica de funcionamiento
 - Características de ejecución de las funciones
 - Seguridad, rendimiento, facilidad de uso, capacidad, etc.
 - Ejemplos de un sistema de gestión de notas:
 - Capaz de gestionar 100 alumnos y hasta 10 exámenes y 5 prácticas por alumno
 - Seguridad de acceso a las notas basada en clave de 128 bits

Categorías requisitos no funcionales



IEEE std. 830

- **Introducción**
 - Objetivo, ámbito, definiciones y siglas, referencias
- **Descripción general**
 - Visión general de producto, funciones, usuarios, limitaciones generales, supuestos y dependencias
- **Requisitos específicos numerados**
 - Funcionales (Entrada, Proceso, Salida)
 - Interfaz
 - Rendimiento, restricciones de diseño, calidad, otros
- **Apéndice**
 - No obligatorio, por ejemplo: ejemplos de formato de entrada/salida

Técnicas de especificación

- Formales:
 - Basadas en lógica formal: ej., notación Z
 - $usados \cap libres = \emptyset$
- Semiformales:
 - Gráficas: notación definida
 - DFD (diagrama de flujo de datos), diagrama de estados
 - Textual: gramática semiformal
 - Pseudocódigo de proceso, BNF
 - Plantilla: formulario
 - DD (descripción de datos): nombre, utilidad y estructura
 - Matriciales: para comprobar más que definir
 - Matriz de trazabilidad (requisito/quien lo genera, quien lo aprueba)

Test

La utilidad de los prototipos rápidos es

- a) Servir de alternativa por si el proyecto falla.
- b) No tienen utilidad puesto que se desechan.
- c) Se usan para establecer una comunicación más precisa y directa con el usuario.
- d) Ninguna de las anteriores