



Técnicas de Documentación y Validación

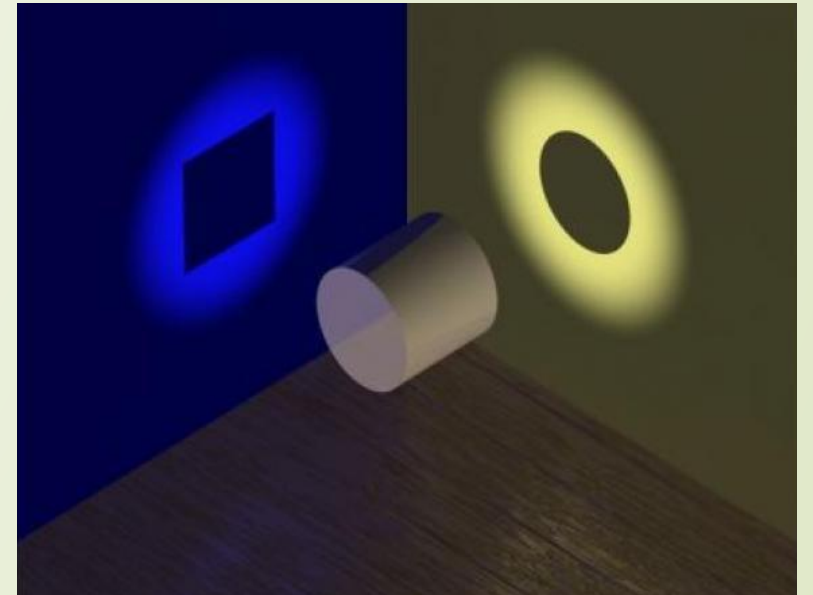
Unidad 1 – V & D de Requerimientos

Objetivos

- Diferentes para usuario y desarrollador

Para el usuario o cliente :

“El objetivo es definir la necesidad de su organización (lo que se pretende y lo que no se pretende). “



Objetivos (cont.)


- Comunicación entre clientes, usuarios y desarrolladores
- Permitir iniciar la actividad de diseño.
- Soportar las actividades de pruebas del sistema
- Controlar la evolución del sistema





Especificación de requisitos

- ¿Qué contiene?
 - Información del problema
 - Interfaz externa del sistema con su entorno.
 - Propiedades y comportamiento del sistema
 - Restricciones de diseño y fabricación del producto
 - Contribución del sistema
 - Restricciones acerca de las propiedades emergentes

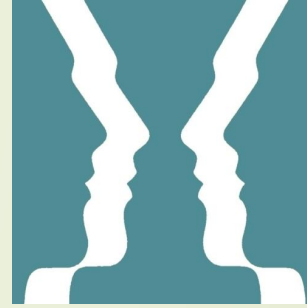


Especificación de requisitos

- ¿Qué NO contiene?
 - Requisitos del proyecto: planificación, costes, fases, hitos,...
 - Diseño
 - Planes de garantía del producto

DRU vs. ERS

(documento de requerimientos de usuario vs especificación requerimientos de software)



DRU

- Punto de vista del usuario/cliente/interesado.
- Sin demasiado nivel de detalle.
- Se incluye la descripción del problema actual (razones por las que el sistema de trabajo actual es insatisfactorio) y las metas que se espera lograr con la construcción del nuevo sistema

ERS

- Punto de vista del desarrollador
- Requisitos del software más detallados.
- Descripción de características del sistema y contribución al logro de los objetivos.



Pero que son los requerimientos?

- “... es una condición o capacidad a la que el sistema (siendo construido) debe cumplir..”
- “... una capacidad del software necesaria por el usuario para resolver un problema o alcanzar un objetivo...”
- “...una capacidad del software que debe ser reunida o poseída por un sistema o componente del sistema para satisfacer un contrato, especificación, estándar, u otra documentación formal...”



Características...

Necesario: Un requerimiento es necesario si su omisión provoca una deficiencia en el sistema a construir, y además su capacidad, características físicas o factor de calidad no pueden ser reemplazados por otras capacidades del producto o del proceso.

Conciso: Un requerimiento es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.

Completo: Un requerimiento está completo si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.



Características... (cont.)

Consistente: Un requerimiento es consistente si no es contradictorio con otro requerimiento.

No ambiguo: Un requerimiento no es ambiguo cuando tiene una sola interpretación. El lenguaje usado en su definición, no debe causar confusiones al lector.

Verificable: Un requerimiento es verificable cuando puede ser cuantificado de manera que permita hacer uso de los siguientes métodos de verificación: inspección, análisis, demostración o pruebas.

Clasificación según RUP

Factor de Calidad	Atributos
Funcionalidad	<ul style="list-style-type: none">✓ Características y capacidades del programa✓ Generalidad de las funciones✓ Seguridad del sistema
Facilidad de uso	<ul style="list-style-type: none">✓ Factores humanos✓ Factores estéticos✓ Consistencia de la interfaz✓ Documentación
Confiabilidad	<ul style="list-style-type: none">✓ Frecuencia y severidad de las fallas✓ Exactitud de las salidas✓ Tiempo medio de fallos✓ Capacidad de recuperación ante fallas✓ Capacidad de predicción
Rendimiento	<ul style="list-style-type: none">✓ Velocidad del procesamiento✓ Tiempo de respuesta✓ Consumo de recursos✓ Rendimiento efectivo total✓ Eficacia
Capacidad de Soporte	<ul style="list-style-type: none">✓ Extensibilidad✓ Adaptabilidad✓ Capacidad de pruebas✓ Capacidad de configuración✓ Compatibilidad✓ Requisitos de instalación



Clasificación general - Funcionales

Requerimientos funcionales:

- Declaraciones de los servicios que proveerá el sistema, de la manera en que éste reaccionará ante entradas particulares y de cómo se comportará en situaciones particulares.
- Pueden declarar lo que el sistema no debe hacer. Una falla de un requerimiento funcional degrada un sistema.



Requerimientos funcionales - Ejemplos

Deben responder a preguntas como:

1. ¿Qué entrada es necesaria para que un cheque se imprima?
2. ¿Bajo qué condiciones puede cambiar el monto del pago?
3. ¿Qué provoca la remoción de un empleado de la nómina de pago?
4. El usuario deberá tener la posibilidad de buscar en el conjunto inicial de la base de Datos o seleccionar un subconjunto de ella.
- 5.. El sistema deberá proveer visores adecuados para que el usuario lea documentos en el almacén de documentos.
6. A cada pedido se le deberá asignar un identificador único que el usuario podrá copiar al área de almacenamiento permanente de la cuenta



Requerimientos NO funcionales

Son aquellos requerimientos que no se refieren directamente a las funciones específicas que entregará el sistema, sino a las propiedades emergentes de éste tales como la fiabilidad, el tiempo de respuesta, la capacidad de almacenamiento, etc.

Definen las restricciones del sistema como la capacidad de dispositivos de E/S y la representación de datos que se utiliza en las interfaces del sistema. En general se refieren al sistema como un todo y no a rasgos particulares, por lo tanto, una falla de un requerimiento no funcional, podría inutilizar un sistema.



Requerimientos NF - Ejemplos

1. El sistema debe ser desarrollado sobre un computador X
2. Los estándares de desarrollo deben ser los que la organización tiene establecidos.
3. El sistema deberá ejecutarse en un cloud privado
4. ...



Práctica N°1A

Desarrollo: #personas /3

Analizar el documento

Describir el glosario de términos

Reescribir el documento de manera ordenada si fuese necesario

Identificar los requerimientos funcionales

Identificar los requerimientos NO funcionales

Identificar atributos de calidad

Identificar los participantes/roles del sistema

Realizar revisión con los pares

Exposición



Ágiles - User stories

¿Que son?

Son partes de un enfoque orientado a negociar los requerimientos más que a escribirlos. Toda User story incluyen una oración o más, pero además contiene la conversación que describe esa funcionalidad.

¿Cómo se escriben?

Como <rol de usuario>,
quiero <función del sistema>
para poder <valor de negocio>

(Idea: anotar caso de test al reverso!!!)



Ágiles - User stories

Independiente -> Permite una mejor priorización

Negociable -> Colaboración y negociación entre las partes

Valiosa -> Aportarle valor al usuario o cliente

Estimable -> En tamaño

Pequeña (Small) -> Para poder estimarla, priorizarla y planearla.

Verificable -> Prueba y confirmación de que cumple con lo esperado.



Algunas User stories de por ahi...

Como **gerente** quiero poder **seleccionar un proveedor** para **atender un pedido**.

Criterio de aceptación:

Cuando ingrese debo poder ver una grilla con los diferentes pedidos para autorizar, al seleccionar uno debo ver las 3 opciones recomendadas por el sector de compras (al hacer clic sobre alguna de ellas se me debe mostrar el detalle del presupuesto) con una casilla de selección por cada una, el sistema solo debe permitirme seleccionar una opción.

Al tener una opción seleccionada se debe habilitar el botón de autorizar pedido.



Algunas User stories de por ahi...

Como **personal de compras** quiero poder **generar una orden de compra** para poder **enviársela al proveedor**

Criterio de aceptación:

Debo poder visualizar todos los pedidos autorizados, al hacer clic sobre alguno de ellos los debo poder visualizar en profundidad y tener un botón “generar orden de compra” que registre dicha orden.



Algunas User stories de por ahi...

Como **personal de compras** quiero poder **registrar una factura** para luego **asociarla a un compromiso de pago**

Criterio de aceptación:

Al ingresar debo poder ver un formulario en donde ingresar los datos de la factura: empresa, fecha, cuil, monto total. Y luego de ingresar tener la opción para confirmar dicha factura



Práctica N°1B

Escribir al menos 5 User stories para 3 roles diferentes.
Definir pruebas de aceptación para cada una.





Cambios en los requerimientos

- Documento ordenado
- Deben minimizarse referencias externas
- Falencias de la documentación electrónica.
- Los requerimientos evolucionan.
- El sistema evoluciona
- Esencial utilizar control de cambios



Validación de requerimientos

- ¿Que significa validar?
 - Demostrar que los requerimientos definidos representan las intenciones/necesidades del cliente.
- ¿Por qué?
 - Los costos de errores en los requerimientos son altos, por lo cual, la validación es muy importante.
 - Fijar un error de requerimiento después del desarrollo puede resultar en un costo 100 veces mayor que fijar un error en la implementación.

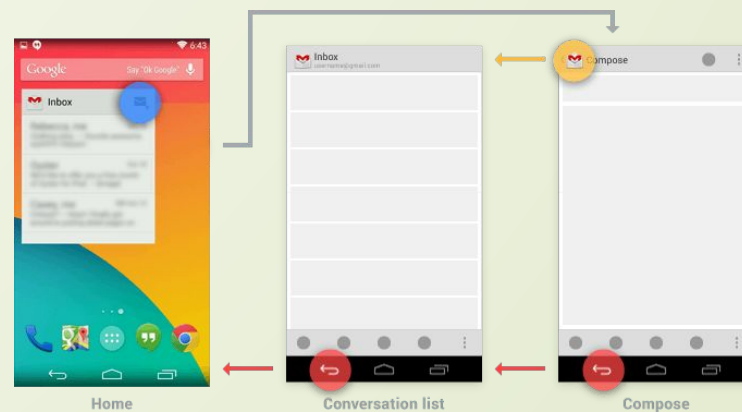


Validación de requerimientos - Prototipado

- **Prototipado de interfaz de usuario**
- Es una técnica de representación aproximada de la interfaz de usuario de un sistema software que permite a clientes y usuarios entender más fácilmente la propuesta.
- **Desechables:** se utilizan sólo para la validación de los requisitos y posteriormente se desechan. Pueden ser prototipos en papel o en software.
- **Evolutivos:** una vez utilizados para la validación de los requisitos, se mejora su calidad y se convierten progresivamente en el producto final.

Validación de requerimientos – Prototipado (cont.)

- **Mock-ups:** Pantallas, típicamente dibujadas a mano en papel. Soporte limitado de verificación. Representan un aspecto concreto del sistema.
- **Storyboards:** Evolución de los mock-ups. Agregan secuencia de acciones, o escenarios, que se deben realizar con el programa.
- **Maquetas:** Versión simplificada del sistema software deseado. Únicamente se implementan pantallas e interconexión entre estas. Puede incluirse implementación sencilla para producir algunos datos reales. (prototipo funcional).



Validación de requerimientos - Inspección o recorrido

- Walkthrough asociada a la inspección de código fuente.
- Objetivos:
 - Detectar conflictos
 - Aumentar el conocimiento
- Autor revisa el producto en detalle, y los participantes manifiestan sus opiniones sobre el mismo.
- Permite a clientes y usuarios comprender el significado de cada requisito y manifestar su acuerdo o desacuerdo con los mismos cuando se aplica a ERS.
- Permite validar de manera natural la secuencia de pasos de un caso de uso.





Validación de requerimientos - Generación de casos de prueba


- Se basa en que los requerimientos deben ser verificables
- Los casos de pruebas son artefactos bien definidos para la prueba de software
- Deben conocerse los datos de entrada así como las tareas a realizar y los resultados esperados
- Ejemplo:
“El sistema permitirá exportar información”



Práctica N°1C

Armar Mock-ups para validar los principales requerimientos

Describir al menos 5 casos de pruebas





Generación de casos de prueba

“El sistema permitirá exportar información de un cliente especificado. Los datos a exportar así como el formato de archivo utilizado se encuentran especificados en el Anexo X”

El caso de prueba debería ser:

1. *Se especifica el identificador del cliente*
2. *Se invoca la función de exportar datos para el cliente especificado*
3. *Se obtiene un archivo conteniendo la información del cliente en el formato determinado*




Modelo de calidad




- Define una serie de *características de calidad* que deben cumplirse en un determinado producto. En el caso de los requisitos de un sistema software, normalmente se definen dos modelos de calidad, uno para la Especificación de Requisitos del Sistema (ERS) completa y otro para los requisitos individuales.
- http://www.buenosaires.gob.ar/sites/gcaba/files/estandar_desarrollo_asi.pdf




Modelo de calidad para ERS

- **Correcta:** La ERS es acorde a la estructura y contenidos especificados en la organización, es decir, contiene todas las secciones obligatorias, el contenido de las secciones es acorde a lo especificado y se usan las plantillas oportunas de forma correcta.
 - **Completa:** La ERS contempla TODO lo que se conoce que debe hacer el sistema software a desarrollar y no contiene secciones obligatorias sin realizar.
 - **Consistente internamente:** La ERS no contiene requisitos u otra información inconsistente entre sí, es decir, no contiene contradicciones internas ni redundancias innecesarias.
 - **Consistente externamente:** La ERS no contiene requisitos u otra información inconsistente con otros documentos del proyecto.
- 



Modelo de calidad para requerimientos específicos

- **Comprensible:** Redactado de manera tal que lo entienden todos los participantes en el proyecto. Se utiliza terminología del dominio presente en el glosario de términos.
- **No ambiguo:** Redactado de manera tal que su significado no depende de la subjetividad del lector.
- **Gramática/Ortográficamente correcto:** Está redactado según la normas gramaticales y ortográficas del idioma correspondiente.
- **Realizable:** Puede implementarse con tecnología existente y a un coste asumible dentro del presupuesto del proyecto.
- **Comprobable:** Está redactado de forma que se pueda definir una o más pruebas, de costo y tiempo razonables, para comprobar que el sistema software a desarrollar cumple el requisito en cuestión.
- **Trazado:** Las dependencias del requisito están trazadas hacia otros requisitos u objetos de documentación de nivel superior de los que depende



Practica N°1D - IEEE 830

- “Presenta el contenido y características de una Buena captura de requerimientos de software así como lineamientos para conseguirlos. Esta práctica está orientada a software a desarrollar pero también puede ser utilizada para la selección de productos comerciales...”
- [ieee830Explicado.pdf](#)
- [practica1_ejemplo.pdf](#)