

Diseño de Sistemas

Unidad 1: Diseño y Sistemas - Parte 2

Martín Agüero

Pablo Sabatino

2019



UTN.BA

DPTO. INGENIERÍA EN SISTEMAS DE INFORMACIÓN
CÁTEDRA DISEÑO DE SISTEMAS

Agenda

Atributos de Calidad del Diseño.

Influencias del Proceso de Diseño: Requerimientos Funcionales, No Funcionales, Reglas de Negocio, Ambiente de Desarrollo y Uso del Sistema, Recursos Humanos y las Herramientas Tecnológicas Disponibles.

Calidad

Donde aplicamos el concepto de calidad

Calidad de...

- de un producto.
- vida.
- del aire, del agua.
- calidad de un producto software.
- de diseño.
- de un proyecto.



<https://www.standishgroup.com/>

Calidad - Definiciones

Diccionario: *“Propiedad o conjunto de propiedades inherentes a una cosa, que permiten apreciarla como igual, mejor o peor que las restantes de su especie”.*

Peter Drucker: *“Calidad es lo que el cliente está dispuesto a pagar en función de lo que obtiene y valora”.*

IEEE

- Grado en que un sistema, componente o proceso satisface los requerimientos especificados.
- Grado en el que un sistema, componente o proceso satisface necesidades y expectativas de un cliente o usuario.

Calidad -

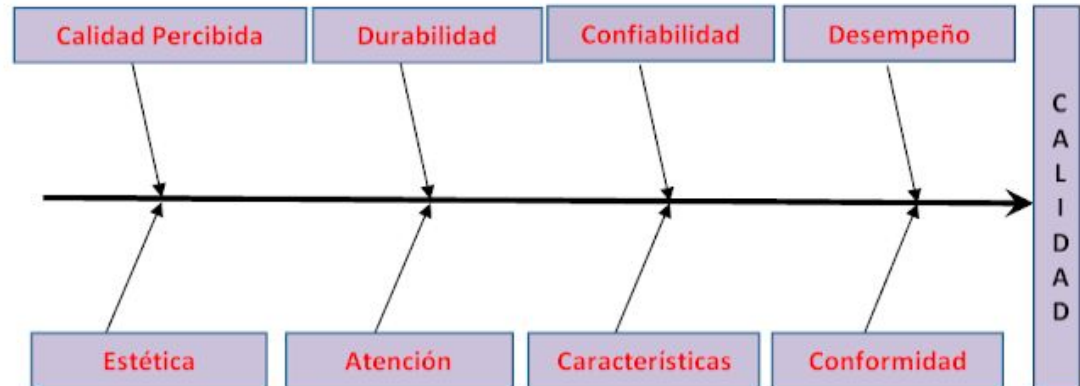
Regla Federal Express

La empresa de mensajería, **Federal Express**, considera que sus gastos anuales por acciones no previstas (errores en los destinos, retrasos aéreos, reclamos de facturas, etc.) superan los 800 millones de dólares, teniendo en cuenta los errores cometidos y la pérdida de clientes. Estos costos se expresan mediante la regla “1-10-100”.

Esta regla indica que si un problema se detecta en el momento que ocurre, su costo es de 1 dólar por hora. Si es detectado más tarde, cuesta 10 dólares. Por último, si es detectado por el cliente, el costo se eleva a unos 100 dólares.

Calidad del Diseño - Las 8 Dimensiones de la Calidad (de Garvin)

En 1987, David A. Garvin publicó en la Harvard Business Review su famoso artículo **“Compitiendo en las Ocho Dimensiones de la Calidad”** en donde propone detalladamente un nuevo modelo para la Calidad: la Gestión Estratégica de la Calidad, convirtiéndose en el más reciente Gurú de la Calidad. Fue en esta publicación en la que él hizo la primera presentación de las Ocho Dimensiones, una estructura conceptual para la reflexión de lo que sería la Calidad de un producto.



+sobre David A. Garvin

https://de.wikipedia.org/wiki/David_A._Garvin

Calidad del Diseño - Las 8 Dimensiones de la Calidad (de Garvin)

(1) Calidad del desempeño.

¿El software entrega y ofrece todo lo relevado en las especificaciones como parte del modelo de requerimientos, dando valor agregado al usuario final?

(2) Calidad de las características.

¿El software tiene características que sorprenden y agradan la primera vez que lo emplean los usuarios finales? ISO 9126

(3) Confiabilidad.

¿El software proporciona todas las características y capacidades sin fallar? ¿Está disponible cuando se necesita? ¿Entrega funcionalidad libre de errores?

Calidad del Diseño - Las 8 Dimensiones de la Calidad (de Garvin)

(4) Conformidad.

¿El software concuerda con los estándares locales y/o internacionales que son relevantes para la aplicación? Por ejemplo, ¿la interfaz de usuario está de acuerdo con las reglas aceptadas del diseño para la selección de menú o para la entrada de datos?

(5) Durabilidad.

¿El software puede recibir mantenimiento (cambiar) o corregirse (actualizarse) una vez en producción? Gestión del cambio.

(6) Servicio. ¿Existe la posibilidad de que el software reciba mantenimiento (cambios) o correcciones en un periodo de tiempo aceptablemente breve? Disponibilidad de servicio.

Calidad del Diseño - Las 8 Dimensiones de la Calidad (de Garvin)

(7) Estética.

UX Design (User Experience Design) o *diseño de experiencia de usuario* es una filosofía de diseño que tiene por objetivo la creación de productos que resuelvan necesidades concretas de sus usuarios finales, consiguiendo la mayor satisfacción y mejor experiencia de uso posible con el mínimo esfuerzo.

(8) Percepción. En ciertas situaciones, existen prejuicios que influyen en la percepción de la calidad por parte del usuario. Por ejemplo, si se introduce un producto de software elaborado por un proveedor que en el pasado ha demostrado mala calidad, muy probablemente se tendrá la percepción de la calidad del producto tendrá influencia negativa (Pressman, R).

Calidad del Diseño

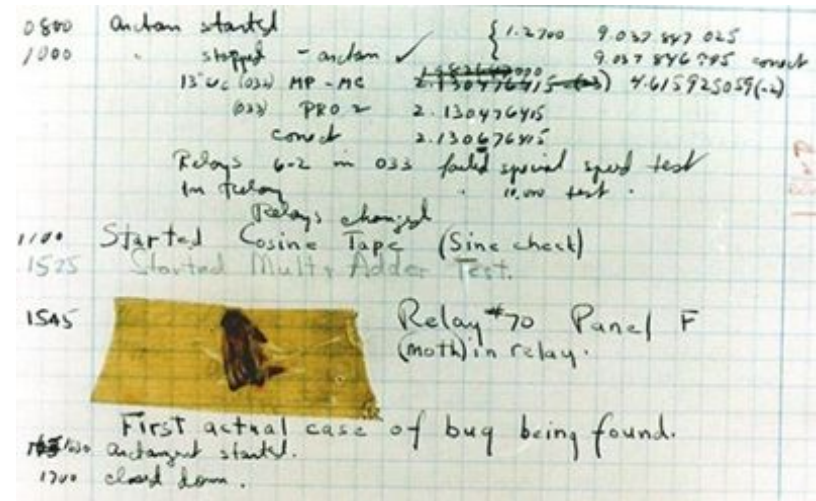
¿Que es un BUG?

Bug es el término que se utiliza coloquialmente para describir un **defecto** en un sistema de información.



Historia del primer Bug:

http://en.wikipedia.org/wiki/Software_bug



Calidad del Diseño

Error / Defecto / Falla

Error: Una equivocación de una persona al desarrollar alguna actividad de desarrollo de software.

Defecto: Se produce cuando una persona comete un error. (por ej: cuando un diseñador comprende mal un requerimiento y crea un diseño inadecuado).

Falla: Es un desvío respecto del comportamiento esperado del sistema. Puede producirse en cualquier etapa.

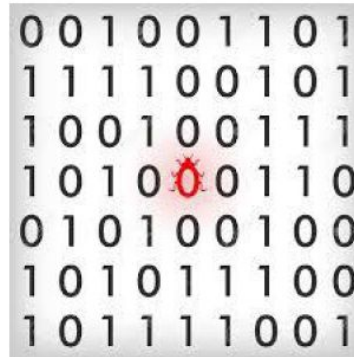
Calidad del Diseño

Error / Defecto / Falla



Error humano

*Puede
derivar
en*



Defecto

*Puede
derivar
en*



Falla

Calidad del Diseño

Razones de las fallas:

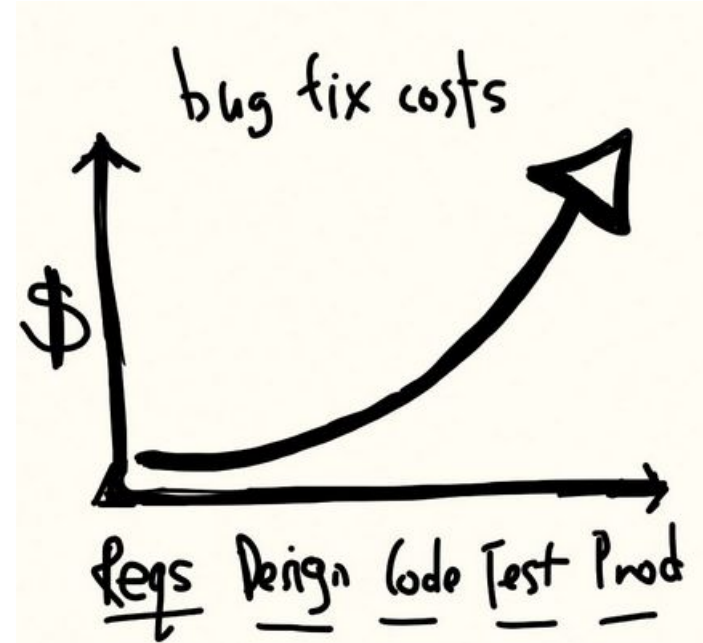
- ▼ Especificación errónea (incompleta o incorrecta)
- ▼ La especificación puede tener un requerimiento imposible de implementar con la infraestructura disponible ([epic user story](#)).
- ▼ Defecto en el diseño del sistema.
- ▼ Defecto en el diseño del alguno de los módulos.
- ▼ Código fuente de un programa erróneo (un bug).

Calidad del Diseño

Costo de la Calidad

Solucionar defectos compite -en prioridad- con desarrollar nuevas funcionalidades y avanzar en el proyecto.

Postergar la solución de bugs, puede llevar a elevadísimos costos (tanto de tiempo, costo y esfuerzo) cuando se lo quiera tratar.



Calidad del Diseño

Costo de la Calidad

Costos de Prevención

Costo de todos aquellos esfuerzos para asegurar la calidad del software y prevenir defectos **en todas** las fases del desarrollo de software. Ejemplo: Actividades de QA, definición de procesos, mejora de procesos, etc.

Costos de evaluación

Costo del esfuerzo para descubrir la condición de la calidad del software (evaluaciones planificadas). Ejemplo: Auditorías, testing, estabilización, etc.

Calidad del diseño - Ejercicio grupal 1:

Para discutir basado en los temas presentados:

¿Cuál de los siguientes paradigmas de la gestión de la calidad corresponde a un enfoque moderno?

- ☐ Cumplir los estándares.
- ☐ Aprovechar el tiempo.
- ☐ Detectar errores.
- ☐ Encontrar bugs.

Calidad del diseño - Ejercicio grupal 2:

Para discutir basado en los temas presentados:

Los costes de prevención son para evitar:

- ☐ Los errores de producción.
- ☐ Los gastos en la inspección del producto.
- ☐ Los errores en cualquier función del programa.
- ☐ Los gastos de las auditorías.

Calidad del Diseño

Cómo evaluar la calidad del diseño

Calidad del Diseño

Medición de calidad

El principal objetivo de Calidad del Software es minimizar la ausencia de **bugs** (**en sentido amplio**) en el software.

Este es el sentido más básico de conformidad con los requerimientos. Se puede expresar cuantitativamente (medir):

- tasa de defectos:
 número de defectos / LOC, FP, y otros
- confiabilidad:
 número de fallas / N horas de operación



Calidad del Diseño

McGlaughlin sugiere tres características para **evaluar un buen diseño**:

- ✓ Debe implementar todos los **requerimientos** explícitos contenidos en el modelo de requerimientos e incluir a todos los requerimientos implícitos que desean los participantes.
- ✓ La **documentación** debe ser una guía legible y comprensible para quienes generan el código y para los que lo prueban y dan el posterior soporte o mantenimiento.
- ✓ Debe abordar tanto el dominio de los datos, como las funcionalidades y el comportamiento desde el punto de vista de la **implementación**.

Calidad del Diseño

Pressman propone las siguientes características para **evaluar un buen diseño**:

- ✓ Debe poseer una **arquitectura** que:
 - Se haya creado con el empleo de estilos o patrones arquitectónicos reconocibles.
 - Esté formada por componentes con buenas características de diseño.
 - Se desarrolle en forma evolutiva, de modo que faciliten la implementación y las pruebas.
- ✓ Debe ser **modular**, es decir, el software debe estar dividido de manera lógica en elementos o subsistemas.
- ✓ Debe contener distintas representaciones de datos, arquitectura, interfaces y componentes.

Calidad del Diseño

Características para **evaluar un buen diseño** según **Pressman** (cont.)

- ✓ Debe conducir a **estructuras de datos** apropiadas para las clases que se van a implementar y que surjan de patrones reconocibles de datos.
- ✓ Debe llevar a **componentes** que tengan características funcionales independientes.
- ✓ Debe conducir a **interfaces** que reduzcan la complejidad de las conexiones entre los componentes y el ambiente externo.
- ✓ Debe obtenerse con el empleo de un **método** repetible motivado por la información obtenida durante el análisis de los requerimientos del software.
- ✓ Debe **representarse** con una notación que comunique con eficacia su significado.

Calidad del Diseño

Atributos de Calidad

Calidad del Diseño - Modelo Furps

Hewlett-Packard en el año 1987 desarrolló un conjunto de atributos de la calidad del software a los que se dio el acrónimo **FURPS**: Funcionalidad, Usabilidad, Confiabilidad, Rendimiento y Mantenibilidad.

Functionality

Usability

Reliability

Performance

Serviceability



Calidad del Diseño - Modelo Furps

Los atributos de calidad FURPS representan el objetivo de todo diseño de software:

- La **funcionalidad** se califica de acuerdo con el conjunto de características y capacidades del programa, la generalidad de las funciones que se entregan y la seguridad general del sistema.
- La **usabilidad** considera que el software debe ser fácil de utilizar, sin esfuerzo adicional, por el usuario para quién está diseñado. Esto significa que debe tener una interfaz de usuario apropiada y una documentación adecuada.

Calidad del Diseño - Modelo Furps

- La **confiabilidad** se evalúa con la medición de la frecuencia y gravedad de las fallas, la exactitud de los resultados que salen, el tiempo medio para que ocurra una falla (TMPF), la capacidad de recuperación ante ésta y lo predecible del programa.
- El **rendimiento** se mide con base en la velocidad de procesamiento, el tiempo de respuesta, el uso de recursos, el conjunto y la eficiencia.
- La **mantenibilidad** combina la capacidad del programa para ser ampliable (extensibilidad), adaptable y servicial (estos tres atributos se denotan con un término más común: mantenibilidad), y además que pueda probarse, ser compatible y configurable y que cuente con la facilidad para instalarse en el sistema y para que se detecten los problemas.

Calidad del Diseño

Jean Kaiser sugiere un conjunto de metas para el diseño:

Simplicidad: Existe una tendencia entre ciertos diseñadores a dar al usuario final “demasiado”: contenido exhaustivo, extremos visuales, animaciones intrusas, páginas web enormes... Es mejor moderación y simplicidad.

Robustez: Es frecuente que una aplicación de software haga una “promesa” implícita al usuario. Éste espera contenido y funciones robustas que sean relevantes para sus necesidades.

Calidad del Diseño

Otros atributos son:

Flexibilidad: La facilidad con que un sistema o componente puede ser modificado para ser utilizado en ambientes o aplicaciones diferentes a los cuales fue específicamente diseñado*.

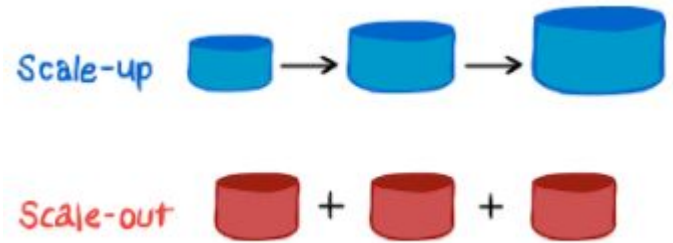
Eficiencia: El software no debe hacer que se malgasten los recursos del sistema, como la memoria y los ciclos de procesamiento. Por lo tanto, la eficiencia incluye tiempos de respuesta y procesamiento, utilización de la memoria y otros.

(*) IEEE Standard glossary of software engineering terminology 610.12-1990 (IEEE Press, Los Alamitos, 1999)

Calidad del Diseño

Escalabilidad: Los sistemas son **escalables** cuando sus capacidades pueden aumentarse al agregar nuevos recursos para enfrentar nuevas demandas del sistema.

En términos de tamaño, hay una distinción entre expandir (scaling up) y ampliar (scaling out). Lo primero significa sustituir recursos en el sistema con recursos más poderosos. Por ejemplo, es posible expandir o aumentar la memoria de un servidor de 16 GB a 64 GB. En cambio, ampliar significa agregar recursos adicionales al sistema (por ejemplo, un servidor Web adicional para trabajar junto a un servidor existente).



Calidad del Diseño - ISO 9126

ISO 9126

El modelo de calidad ISO 9126 es un estándar internacional para la evaluación del software. Se basó (principalmente) en el trabajo de los autores McCall (1977), Boehm (1978) y el modelo FURPS.


Identifica 6 atributos principales que caracterizan la calidad del software:

- Funcionalidad (Functionality)
- Confiabilidad (Reliability)
- Usabilidad (Usability)
- Eficiencia (Efficiency)
- Mantenibilidad (Maintainability)
- Portabilidad (Portability)

Calidad del Diseño - ISO 9126 (características)

Funcionalidad (Functionality)


Capacidad del software de proveer los servicios necesarios para cumplir con los requisitos funcionales.

	Idoneidad (si realiza lo esperado)	Suitability
	Preciso (correctitud de las operaciones)	Accurateness
	Interoperabilidad (interacción con otros sistemas)	Interoperability
	Seguridad (impide accesos no autorizados)	Security
	Conformidad (a normas, reglas, leyes, etc.)	Compliance

Calidad del Diseño - ISO 9126 (características)

Confiabilidad (Reliability)


Capacidad del software de mantener las prestaciones requeridas del sistema, durante un tiempo establecido y bajo un conjunto de condiciones definidas.

	Madurez (frecuencia de fallas)	Maturity
	Tolerancia a fallas	Fault tolerance
	Recuperabilidad	Recoverability

Calidad del Diseño - ISO 9126 (características)

Usabilidad (Usability)


la capacidad de un producto software de ser comprendido, aprendido, usado y de ser atractivo para el usuario, en condiciones específicas de uso. Esta definición es pone el énfasis en los atributos internos y externos del producto, los cuales contribuyen a su usabilidad.

	Comprensibilidad	Understandability
	Facilidad de aprendizaje	Learnability
	Operabilidad (facilidad de operación)	Operability

Calidad del Diseño - ISO 9126 (características)

Eficiencia (Efficiency)

Esta característica permite evaluar la relación entre el nivel de funcionamiento del software y la cantidad de recursos usados.

	Comportamiento en el tiempo	Time behavior
	Comportamiento de los recursos	Resource behavior

Calidad del Diseño - ISO 9126 (características)

Mantenibilidad (Maintainability)

Se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya sea por la corrección de errores o por el incremento de funcionalidad.



Analizabilidad (encontrar causas de fallas)

Analyzability

Posibilidad de cambio

Changeability

Estabilidad (sensibilidad al cambio)

Stability

Capacidad de prueba

Testability

Calidad del Diseño - ISO 9126 (características)

Portabilidad (Portability)

Conjunto de atributos relacionados con la capacidad de un sistema SW para ser transferido desde una plataforma a otra.



Adaptabilidad (cambio de contexto)

Adaptability

Instalabilidad (esfuerzo para inst.)

Installability

Conformidad

Conformance

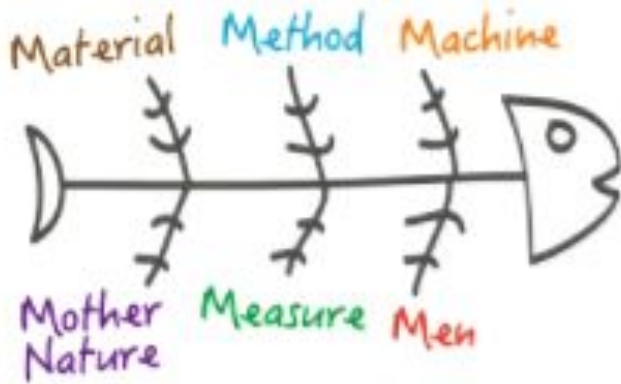
Reemplazabilidad (plug and play?)

Replaceability

Calidad del Diseño

Buscando soluciones ágiles a los problemas

Diagrama de Ishikawa o de espina de pez (Causa Efecto)
Los 5 Porqués



¿POR QUÉ?



¿POR QUÉ?



¿POR QUÉ?



¿POR QUÉ?



¿POR QUÉ?

La Solución Real Se
Encuentra Aquí

Calidad del Diseño

Ejercicio grupal:

Partiendo de un caso hipotético donde un Comité debe definir el presupuesto para un proyecto de software. Elija el tipo de software e indique cómo distribuiría los \$100.000 asignados de acuerdo a los atributos definidos en la norma ISO 9126 (en porcentajes). Justifique.

- ☐ Web App (CRUD)
- ☐ Aplicación Mobile (CRUD)
- ☐ Procesamiento por lotes (batch)
- ☐ Interfaz con otro sistema
- ☐ Streaming de video (web)
- ☐ Robótica (sist. embebido)
- ☐ CRM (gest. relación con los clientes)
- ☐ Facturación y control de stock (web)
- ☐ Repositorio de documentos públicos
- ☐ Gestión de redes de comunic. (consola)

Funcionalidad	%
Usabilidad	%
Confiabilidad	%
Eficiencia	%
Portabilidad	%
Mantenibilidad	%
Otro:	%

Calidad del diseño

Ejercicio grupal:

Responsabilidad de la calidad final del proyecto presentado en el ejercicio anterior:

- ☐ Full Stack Developer
- ☐ Analista de QA
- ☐ Configuration Manager
- ☐ DBA
- ☐ Usuarios o clientes del sistema
- ☐ Analistas de negocio
- ☐ Scrum Master
- ☐ Product Owner (dueño de producto).
- ☐ Project Manager
- ☐ Ninguno



Influencias del Proceso de Diseño

Influencias del Proceso de Diseño: Stakeholders

Influencias del Proceso de Diseño

Stakeholders (parte interesada)

Son los individuos u organismos que *ganan* o *pierden* con algún cambio, son impactados positiva o negativamente independientemente o no de su voluntad. Tienen interés en el resultado.



Influencias del Proceso de Diseño

Stakeholders

Clasificación



Interesados en su construcción

Responsabilidad por el diseño y desarrollo
(Project Managers, Diseñadores, Expertos)

Interés financiero

Responsabilidades sobre la compra o venta del desarrollo
(Analistas de negocios, gerente de compras, ventas)



Interesados en el despliegue y operación

Responsables de implementación y mantenimiento
(Equipo de capacitación, soporte, infraestructura)

Interesados en el uso

Todos los usuarios (directos e indirectos)



Influencias del Proceso de Diseño

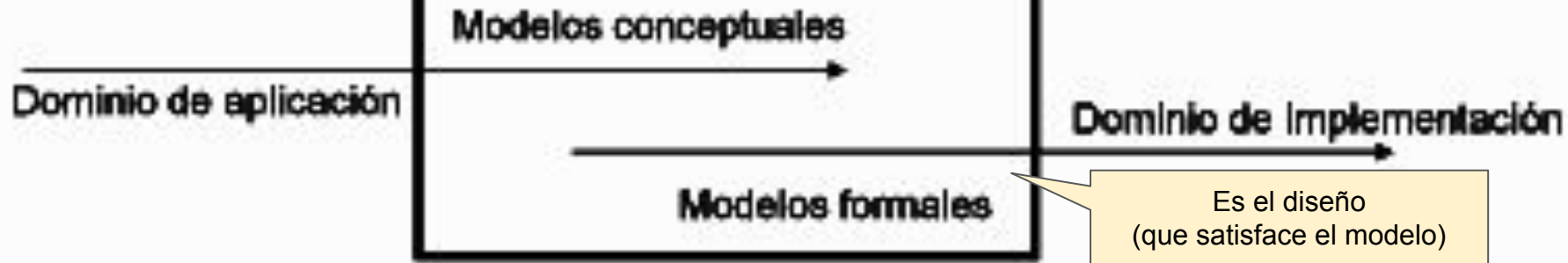
Ambiente de desarrollo y uso del sistema

Influencias del Proceso de Diseño

Ambiente de Desarrollo y Uso del Sistema

Describen una solución al problema

Diseño y Desarrollo



Influencias del Proceso de Diseño

Ambiente de Desarrollo y Uso del Sistema

"El proceso de software siempre involucra transformaciones desde modelos mentales informales hasta una implementación formal (que detalla cómo operará la solución provista)" Blum, B., Software Engineering. A holistic View, Oxford University Press, New York 1992.

Se necesita de la **abstracción** para tratar con la complejidad:

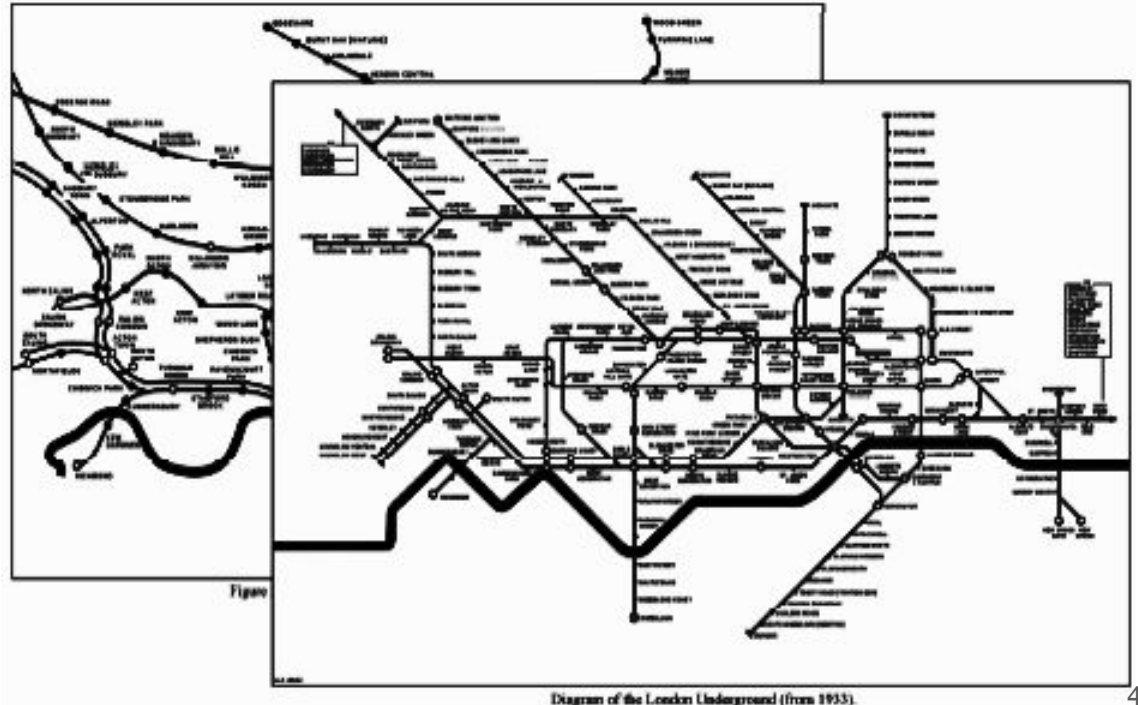
- Es un examen selectivo de ciertos aspectos
- Su objetivo es destacar los aspectos importantes y suprimir los no importantes

Influencias del Proceso de Diseño

Ambiente de Desarrollo y Uso del Sistema

Ejemplo de abstracción

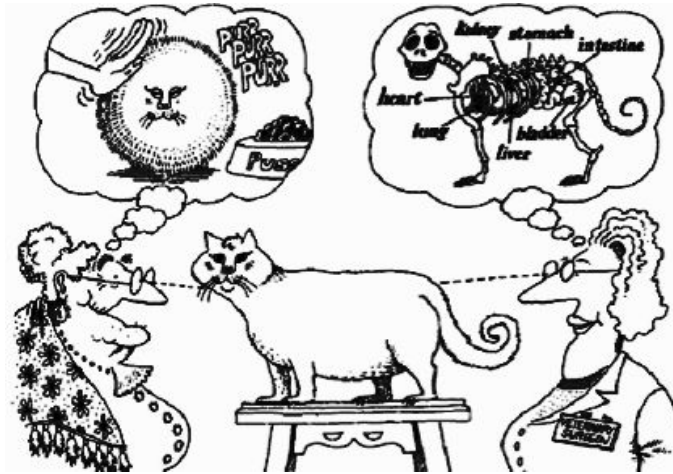
En este caso, el diagrama esquemático, permite un entendimiento más claro de cómo moverse por la red de subte.



Influencias del Proceso de Diseño

Ambiente de Desarrollo y Uso del Sistema

Abstracción



“La abstracción enfoca las características esenciales de un objeto relativas a la perspectiva del observador”. Booch, G., Object Oriented Design with Applications, 1991.

Influencias del Proceso de Diseño

Requerimientos

Influencias del Proceso de Diseño

Requerimientos

“La fase más compleja de construir un sistema de software es decidir con precisión qué construir. Ninguna otra fase del trabajo conceptual es más difícil que la de establecer requerimientos detallados, incluyendo las interfaces con los usuarios, con el hardware y otros sistemas. Ninguna otra fase puede causar tantos perjuicios al sistema si se realiza mal”. Brooks, No Silver Bullets, 1987.

Influencias del Proceso de Diseño

Requerimientos del Usuario y del Sistema

Requerimientos del usuario: Son declaraciones, en lenguaje natural y en diagramas, de los servicios que se espera que el sistema proporcione y de las restricciones bajo las cuales debe funcionar.

Requerimientos del sistema: Establecen con detalle las funciones, servicios y restricciones operativas del sistema. El documento de requerimientos del sistema (algunas veces denominado especificación funcional) debe ser preciso. Debe definir exactamente qué es lo que se va a implementar.

Influencias del Proceso de Diseño

Requerimientos Funcionales y No Funcionales

A su vez, los *requerimientos del sistema* se clasifican en funcionales y no funcionales:

Requerimientos Funcionales: Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales también pueden declarar explícitamente lo que el sistema no debe hacer.

Requerimientos No Funcionales: Son restricciones de los servicios ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad.

Influencias del Proceso de Diseño

Requerimientos

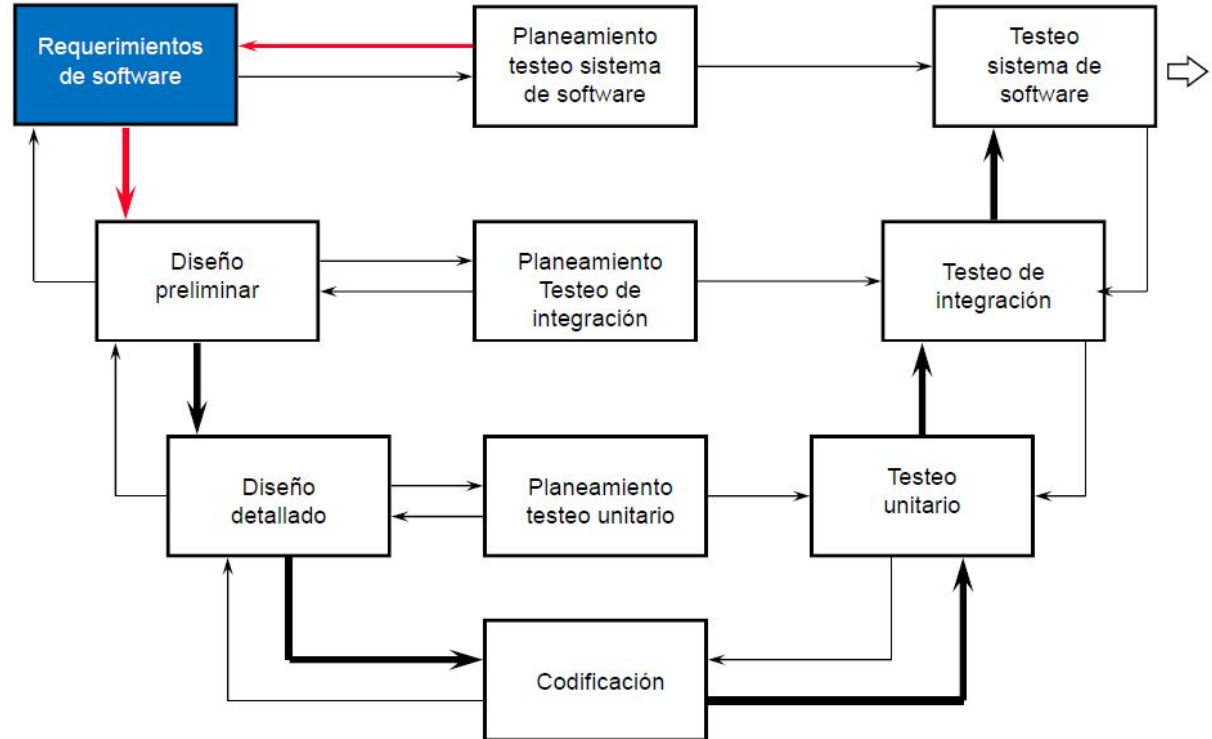
Definición IEEE-Std-610 (1990)

1. *Condición o capacidad que necesita el usuario para resolver un problema o alcanzar un objetivo.*
2. *Condición o capacidad que debe satisfacer o poseer un sistema o un componente de un sistema para satisfacer un contrato, un estándar, una especificación u otro documento formalmente impuesto.*
3. *Representación documentada de una condición o capacidad.*

Influencias del Proceso de Diseño

Requerimientos

Pueden ser vistos como el punto de partida y de llegada del proceso de desarrollo.



Influencias del Proceso de Diseño

Requerimientos

El rol de los Requerimientos:

- ✓ Es un acuerdo entre desarrolladores-clientes-usuarios finales.
- ✓ Base para el diseño del sistema.
- ✓ Soporte para verificación y validación.
- ✓ Soporte para la evolución del sistema.

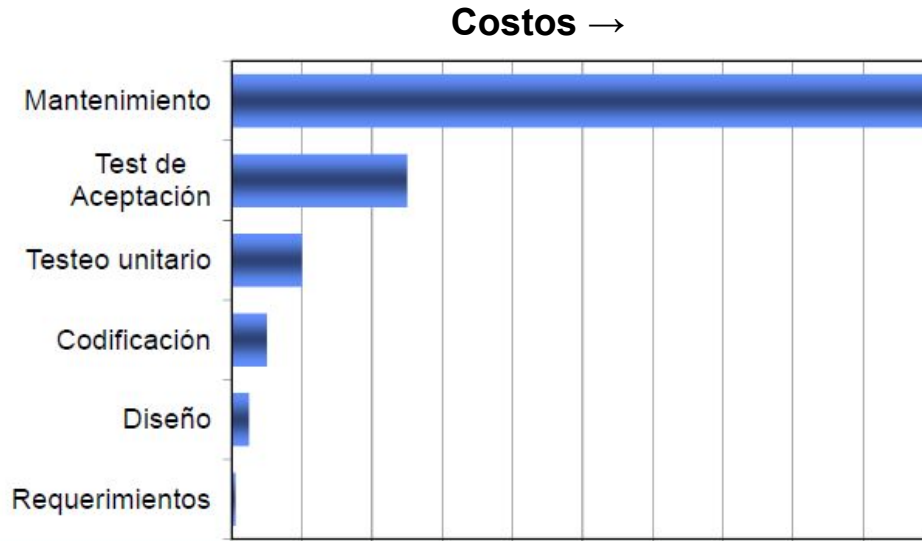
Influencias del Proceso de Diseño

Requerimientos



Influencias del Proceso de Diseño

Requerimientos

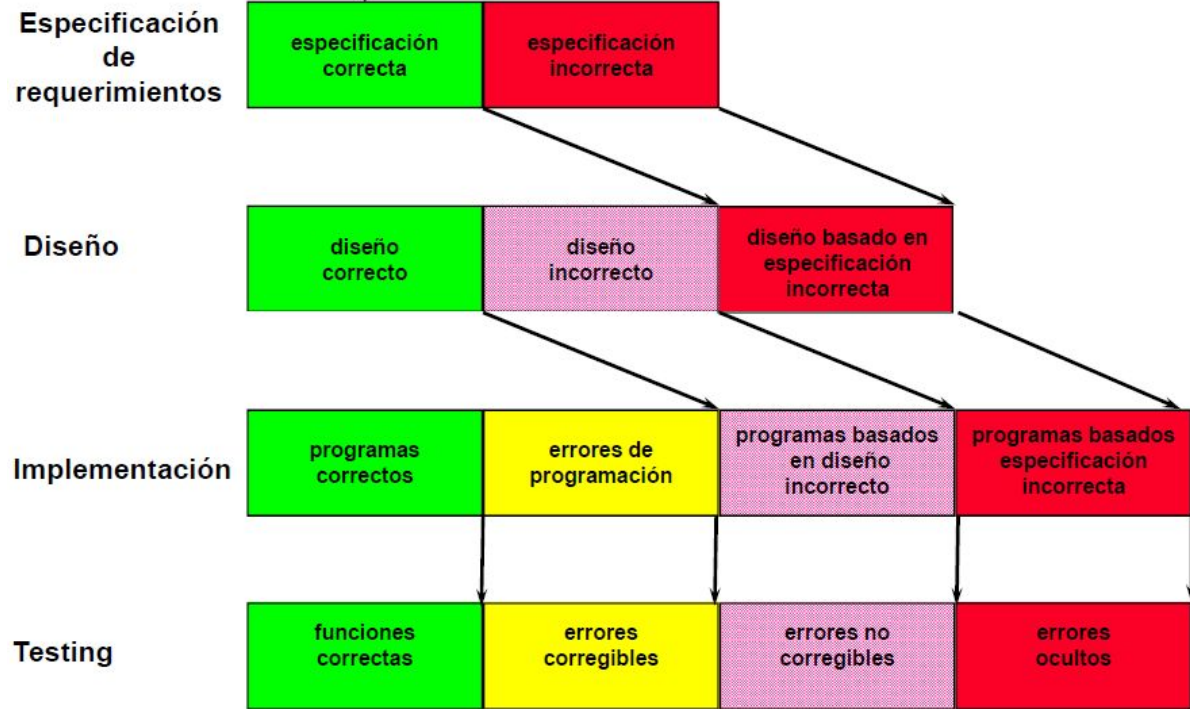


Reparar un error en un producto liberado cuesta entre 100 y 200 veces más que hacerlo en la fase de requerimientos

Influencias del Proceso de Diseño

Requerimientos

Modelo de propagación de defectos



Influencias del Proceso de Diseño

Requerimientos

Importancia de los Requerimientos

- ✓ Cuanto más tarde en el ciclo de vida se detecta un error, más cuesta repararlo.
- ✓ Muchos errores permanecen latentes y no son detectados hasta bastante después de la etapa en que se cometieron.
- ✓ Los errores de requerimientos son típicamente: hechos incorrectos, omisiones, inconsistencias y ambigüedades.
- ✓ Los errores en los requerimientos pueden detectarse.

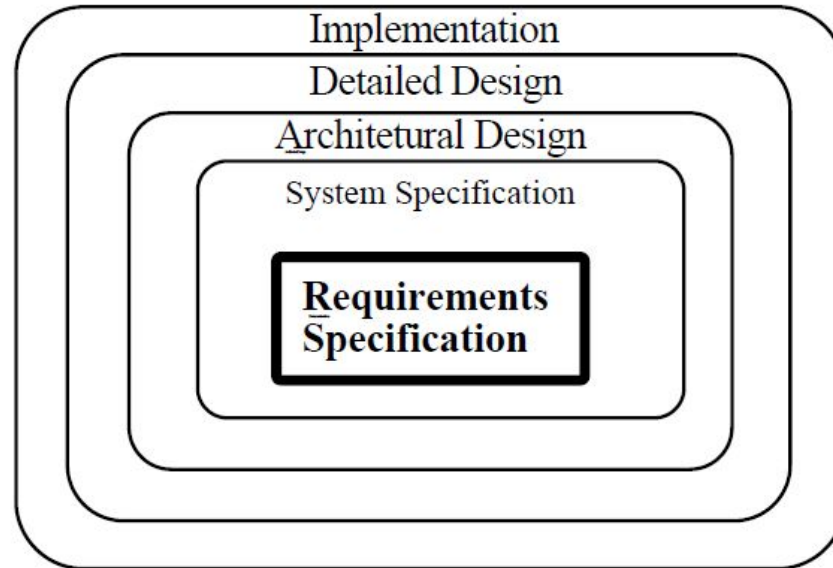
Influencias del Proceso de Diseño

Especificación de Requerimientos de Software (SRS)

Influencias del Proceso de Diseño

Requerimientos

Especificación de Requerimientos



Influencias del Proceso de Diseño

Requerimientos

Especificación de Requerimientos de Software (SRS)

Una especificación de requerimientos de software (Software Requirements Specification) es un documento formal que se crea cuando debe especificarse una descripción detallada de todos los aspectos del software que se va a elaborar. Su contenido incluye: Requerimientos Funcionales, No Funcionales, Interfaces, Atributos especiales y otros.

IEEE SRS: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=278253>

Influencias del Proceso de Diseño

Especificación de Requerimientos de Software

Una especificación de requerimientos de software (Software Requirements Specification) es un documento que se crea para establecer una descripción detallada de todos los aspectos del software que se va a elaborar, antes comenzar el proyecto. El contenido habitual de una SRS es:

1. Introducción

- 1.1 Propósito
- 1.2 Convenciones del documento
- 1.3 Audiencia objetivo y sugerencias de lectura
- 1.4 Alcance del proyecto
- 1.5 Referencias

2. Descripción general

- 2.1 Perspectiva del producto
- 2.2 Características del producto
- 2.3 Clases y características del usuario

Influencias del Proceso de Diseño

2. Descripción general (cont.)

- 2.4 Ambiente de operación
- 2.5 Restricciones de diseño e implementación
- 2.6 Documentación para el usuario
- 2.7 Supuestos y dependencias

3. Características del sistema

- 3.1 Característica 1 del sistema (pueden ser casos de uso)
- ...
- 3.n Característica n del sistema

4. Requerimientos de la interfaz externa

- 4.1 Interfaces de usuario
- 4.2 Interfaces del hardware
- 4.3 Interfaces del software
- 4.4 Interfaces de comunicaciones

Influencias del Proceso de Diseño

Especificación de Requerimientos de Software

5. Otros requerimientos no funcionales

- 5.1 Requerimientos de desempeño
- 5.2 Requerimientos de seguridad
- 5.3 Requerimientos de estabilidad
- 5.4 Atributos de calidad del software

6. Otros requerimientos

- Apéndice A: Glosario
- Apéndice B: Modelos de análisis
- Apéndice C: Lista de conceptos

Fuente: Karl Wieggers http://www.processimpact.com/process_assets/srs_preview.pdf

Otra plantilla: ISO/IEC/IEEE 29148:2011 <https://standards.ieee.org/findstds/standard/29148-2011.html>

Referencias

Pressman, R., Ingeniería del Software.
Mc Graw Hill. 2010.

Sommerville, I. Ingeniería de Software.
Pearson. 2011.

Oliveros, Alejandro. Calidad del Software.
Postgrado Universidad Nacional de La Plata. 2011.

The ISO 9126 Standard.
<http://www.issco.unige.ch/en/research/projects/ewg96/node13.html>

IEEE Standard Glossary of Software Engineering Terminology. 610.12-1990
<http://ieeexplore.ieee.org/servlet/opac?punumber=2238>

Mizuno, Y., Software Quality Improvement.
IEEE Computer. 1983.