

Ingeniería de Requisitos

Tema 2. Proceso de Ingeniería de Requisitos

*Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante*

Materiales cedidos para su uso docente por el prof. Emilio Insfrán Pelozo. Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia

Contenido

- Procesos
 - Entradas y salidas
 - Variabilidad
- Proceso de IDR
 - Fases
 - Elicitación
 - Definición
 - Especificación
 - Validación
 - Gestión de requisitos
- ERS (Documentación – Estándares)
- Herramientas de soporte
- Mejora de procesos
 - Modelo de Madurez
 - Modelo de Madurez para la IDR

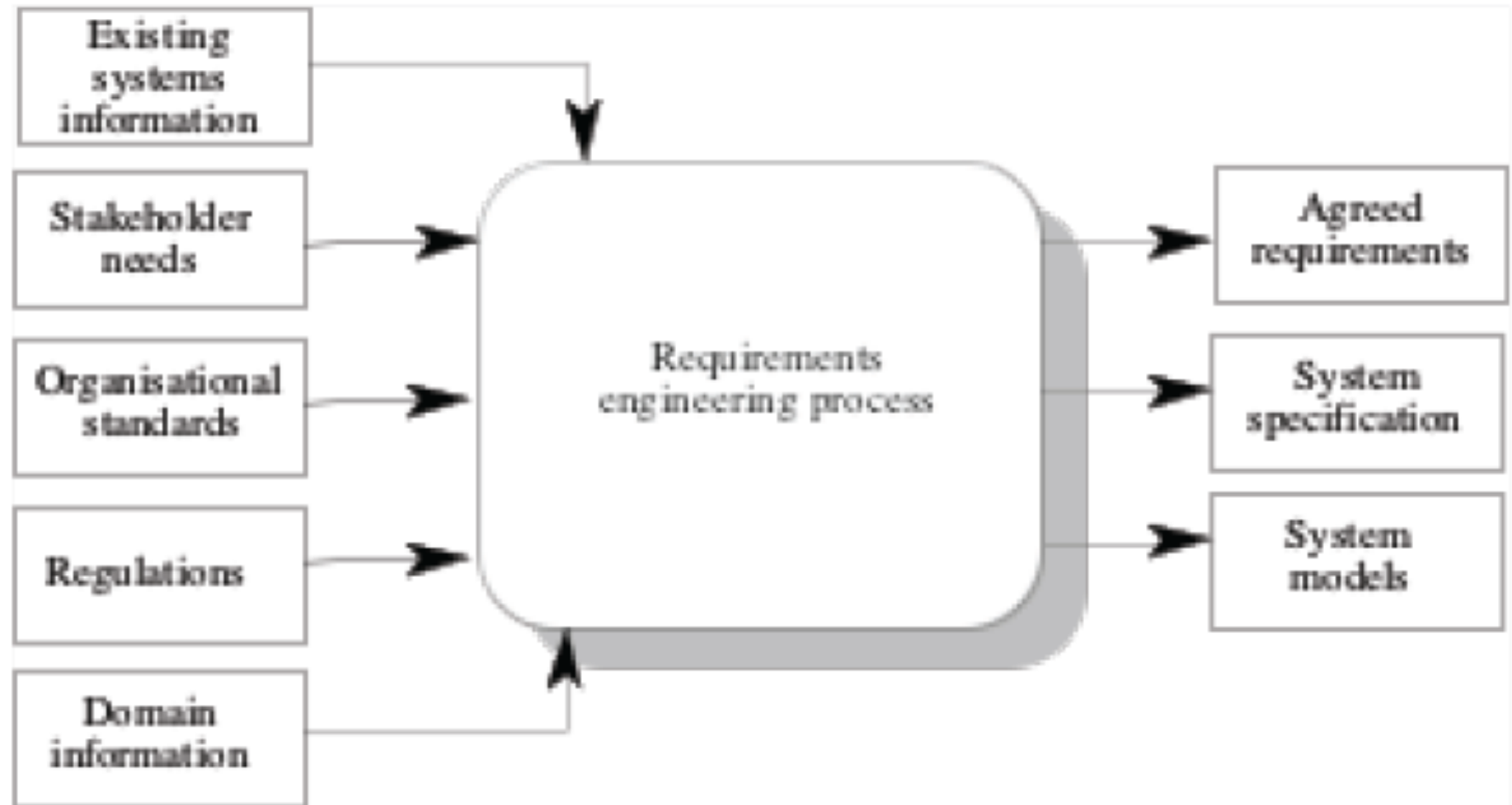
Procesos

- Un proceso es un ***conjunto organizado de actividades*** que transforman entradas en salidas
- Las descripciones de procesos ***encapsulan conocimiento*** contrastado y permiten que éste sea reutilizado
- Ejemplos de descripciones de procesos:
 - Manual de instrucciones para una lavadora
 - Manual de procedimientos de una entidad bancaria (préstamos, transferencias entre distintas entidades,...)
 - Metodología para el desarrollo de software (RUP,...)

Procesos

- En algunos casos, las descripciones de procesos están definidas a un nivel de detalle muy fino
 - los pasos a ejecutar deben realizarse exactamente como se indican en el proceso
 - pero esto no se aplica a procesos complejos...
- Procesos de diseño (en general, no sólo en el ámbito del software)
 - Son procesos que involucran creatividad, interacciones con un amplio rango de personas, decisiones de ingeniería, conocimiento previo y experiencia
 - Las entradas para este tipo de procesos no están definidas de forma clara y explícita (hay que *identificarlas* y *extraerlas*)

Proceso de IDR: entradas y salidas



Proceso de IDR: entradas y salidas

Input or output	Type	Description
Existing system information	Input	Information about the functionality of systems to be replaced or other systems which interact with the system being specified
Stakeholder needs	Input	Descriptions of what system stakeholders need from the system to support their work
Organisational standards	Input	Standards used in an organisation regarding system development practice, quality management, etc.
Regulations	Input	External regulations such as health and safety regulations which apply to the system.
Domain information	Input	General information about the application domain of the system
Agreed requirements	Output	A description of the system requirements which is understandable by stakeholders and which has been agreed by them
System specification	Output	This is a more detailed specification of the system functionality which may be produced in some cases
System models	Output	A set of models such as a data-flow model, an object model, a process model, etc. which describes the system from different perspectives

Proceso de IDR: Variabilidad

- Los procesos aplicados en la IDR varían radicalmente de una organización a otra
- Los **factores** que influyen en esta variabilidad incluyen:
 - Madurez técnica
 - Cultura organizacional
 - Dominio de aplicación
 - Otras disciplinas de soporte involucradas

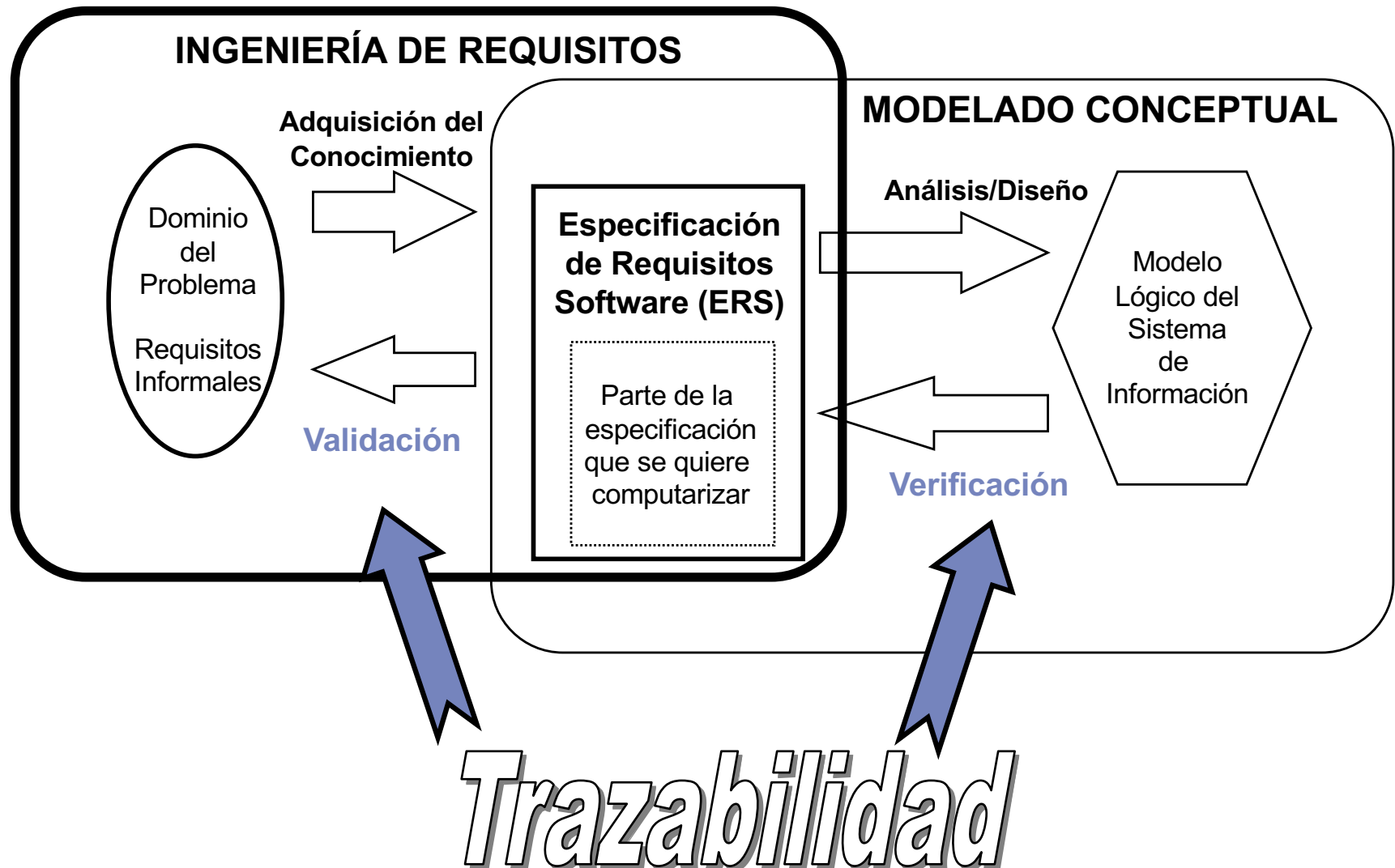
Contenido

- Procesos
 - Entradas y salidas
 - Variabilidad
- Proceso de IDR
 - Fases
 - Elicitación
 - Definición
 - Especificación
 - Validación
 - Gestión de requisitos
- ERS (Documentación – Estándares)
- Herramientas de soporte
- Mejora de procesos
 - Modelo de Madurez
 - Modelo de Madurez para la IDR

Modelo de proceso

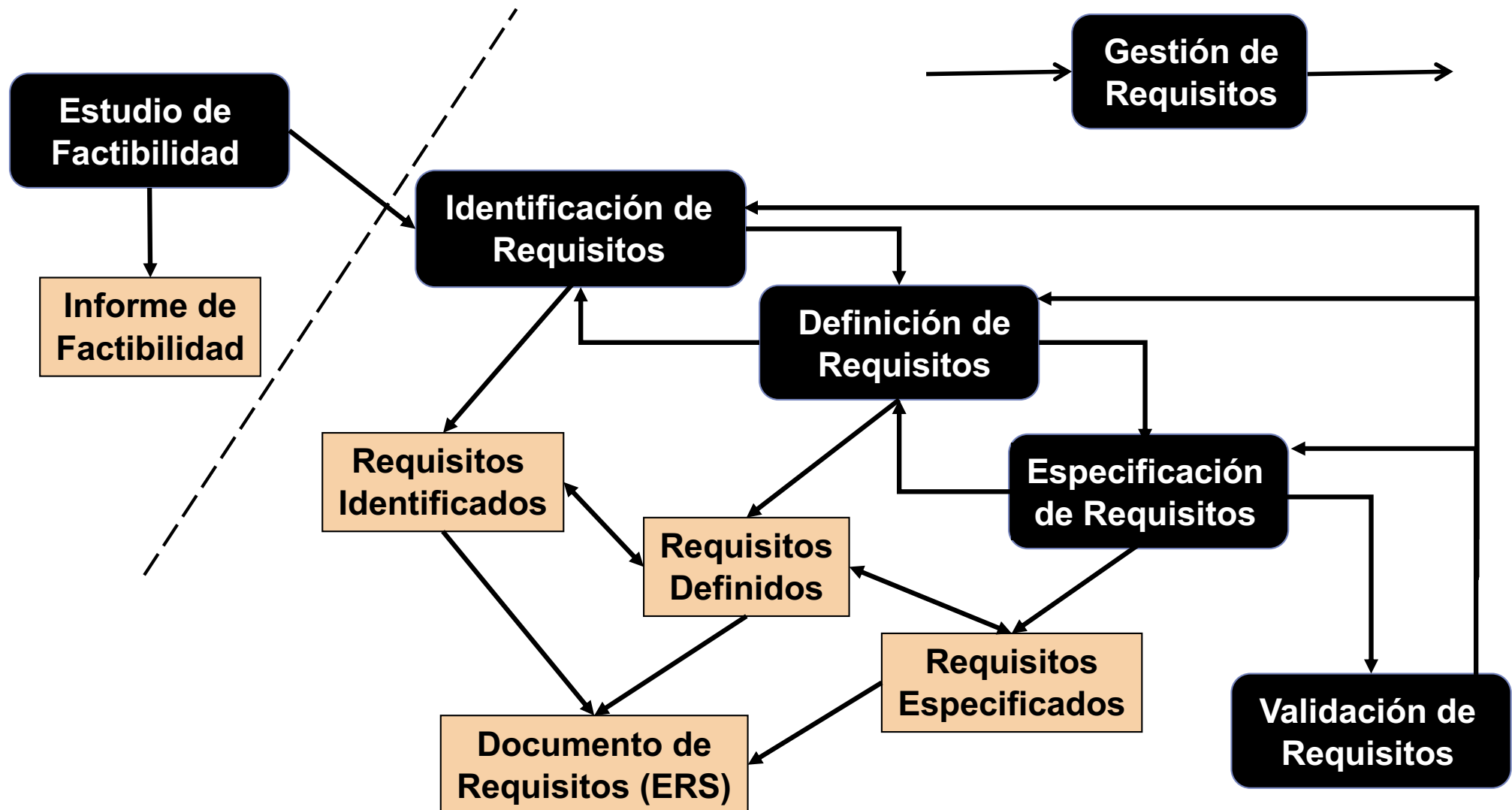
- Un modelo de proceso es una ***descripción simplificada*** de un proceso
- Normalmente se obtiene por la ***generalización y abstracción*** de ese proceso aplicado en diferentes contextos, por diferentes participantes, etc.

Modelo de proceso para la IDR



Ingeniería de Requisitos

Proceso de IDR. Fases y Documentos



Ingeniería de Requisitos

Proceso de IDR. Fases

- **Estudio de Factibilidad**

- Averiguar si las necesidades del usuario pueden ser satisfechas dadas la tecnología y presupuesto disponibles. No forma parte del proceso de IDR.

IDR

- **Identificación/Elicitación de Requisitos**

- Identificar las necesidades concretas del usuario

- **Definición de Requisitos**

- Definir los requisitos en una forma entendible por el usuario

- **Especificación de Requisitos**

- Definir los requisitos con detalle para los desarrolladores

- **Validación de Requisitos**

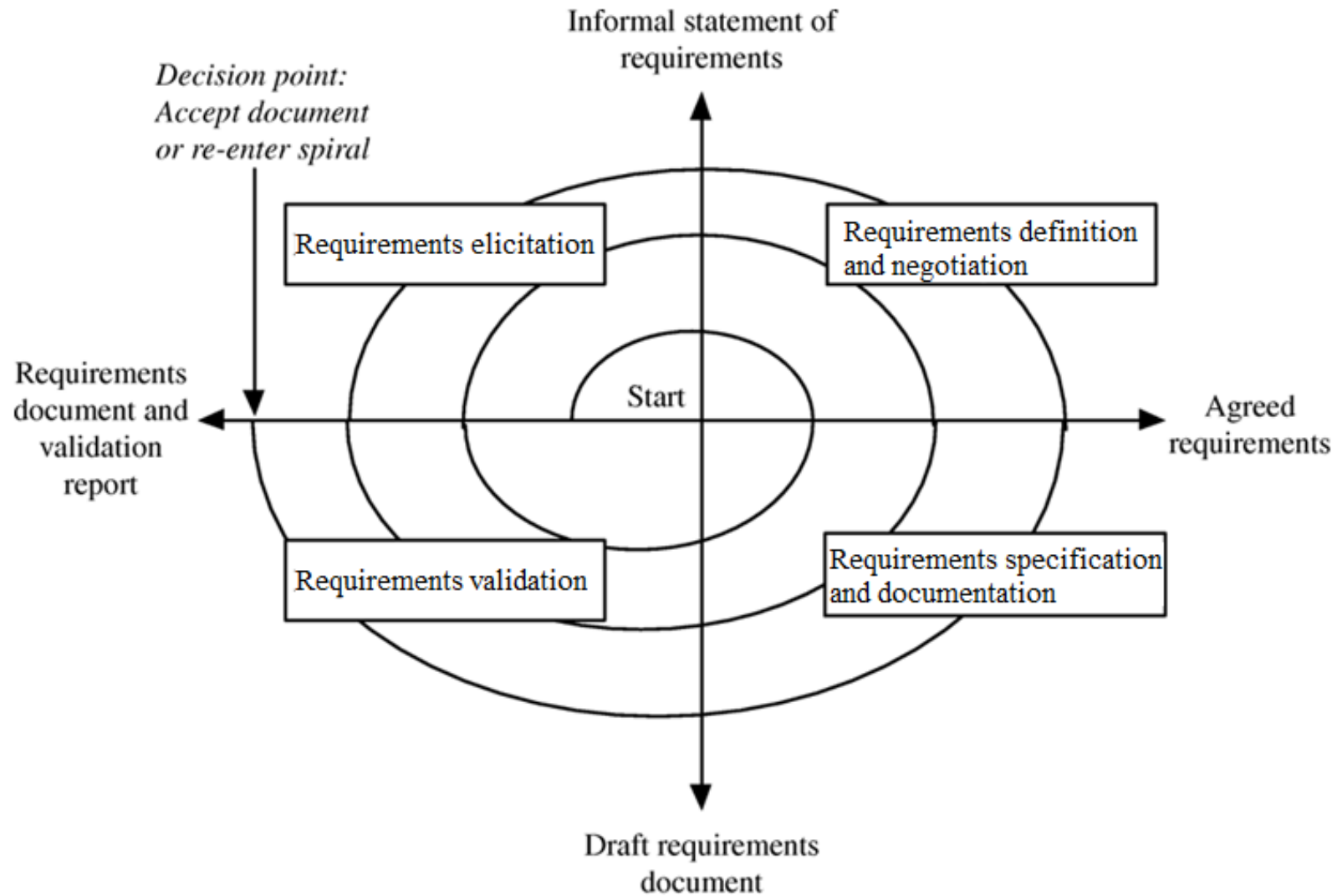
- Comprobar que son los requisitos correctos

- **Gestión de Requisitos**

- Gestión de los cambios. Ocurre en paralelo con las anteriores actividades.

Ingeniería de Requisitos

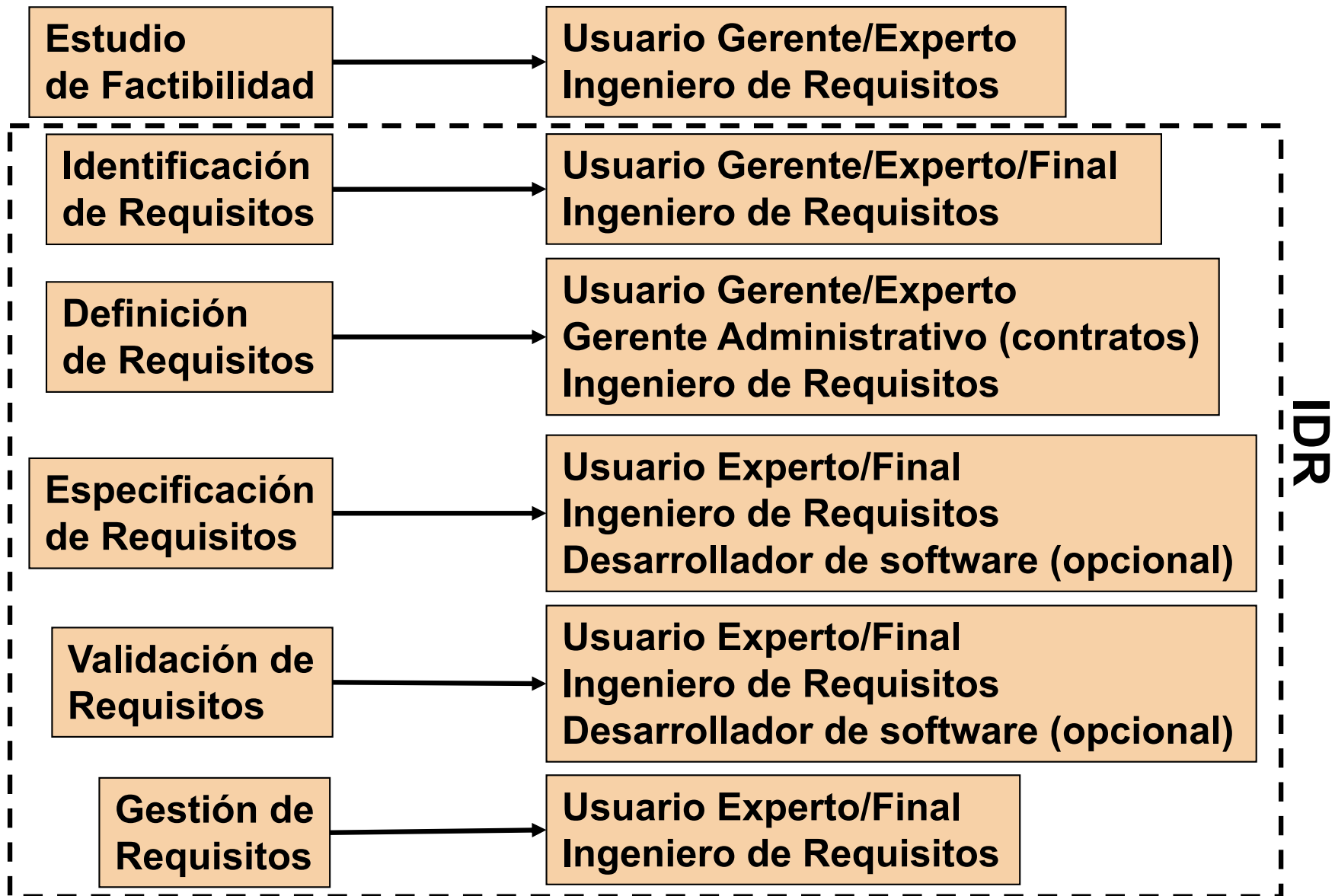
Proceso de IDR. Modelo de espiral



Source: Kotonya and Sommerville (1998, p. 35)

Ingeniería de Requisitos

Proceso de IDR. Actores



Ingeniería de Requisitos

Proceso de IDR. Identificación de Requisitos (Elicitación) (1/2)

- Identificación de Requisitos (Elicitación)
 - Actividades para **entender los objetivos y motivos** para el desarrollo del sistema software propuesto
 - **Identificación de los requisitos** que el sistema software resultante debe satisfacer para alcanzar los objetivos
- Los requisitos pueden ser muy distintos:
 - Desde modificaciones de problemas y sistemas "conocidos"
 - Hasta nuevos problemas complejos a ser automatizados o relativamente "requisitos no acotados" que están abiertos a innovación

Ingeniería de Requisitos

Proceso de IDR. Identificación de Requisitos (Elicitación) (2/2)

- Técnicas de Elicitación
 - Orientadas a la identificación de "stakeholders"
 - Analógicas: metáforas de pensamiento, personas, etc
 - Orientadas a descubrir requisitos: brainstorming, workshops,...
 - Animación de modelos, simulaciones, storyboards o sketching, casos de uso de negocio, modelos organizacionales, etc.
 - Análisis de documentos, ...

Ingeniería de Requisitos

Proceso de IDR. Definición de Requisitos

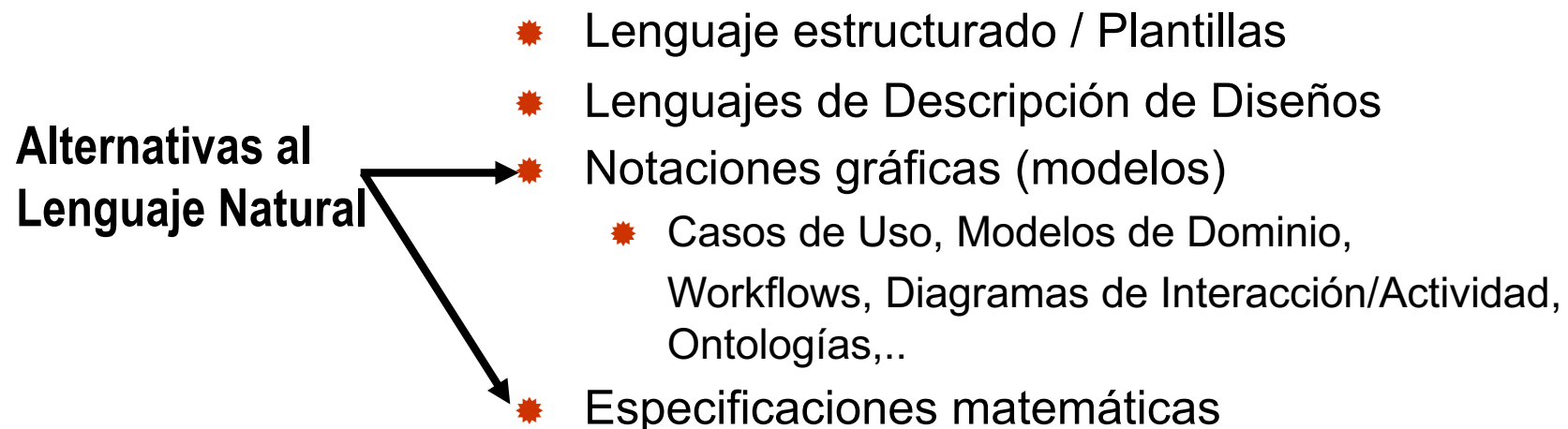
- Definición de Requisitos
 - Descripciones “**orientadas al cliente**” de la funcionalidad del sistema y de las restricciones de su operación (uso)
 - Habitualmente esta fase implica una actividad de “**negociación**” para tratar con los requisitos **contradictorios** o con **los alcances reales** de los requisitos identificados
 - Debe especificar “**comportamiento externo**” del sistema independiente de aspectos tecnológicos y del particionamiento interno del sistema.
 - **Modelos estáticos o dinámicos del dominio del problema**: modelos de dominio, modelos de tareas, casos de uso, etc.
 - Puede servir de base para el **contrato** de desarrollo del sistema.
 - Incluye **Requisitos Funcionales y No-Funcionales**
 - Requisitos Funcionales son descripciones de los servicios que el sistema debe proporcionar
 - Requisitos No-Funcionales son restricciones o atributos de los servicios y funciones ofertadas por el sistema

Ingeniería de Requisitos

Proceso de IDR. Especificación de Requisitos

- Especificación de Requisitos
 - **Añaden detalle** a la definición de requisitos. Descripciones detalladas y precisas de la funcionalidad del sistema y restricciones.
 - **Objetivo:** comunicar QUÉ se requiere de los desarrolladores.
 - (Habitualmente) representadas con **modelos** (describen el sistema a desarrollar a un nivel de abstracción alto)
 - Puede servir de base para el **contrato** de desarrollo del sistema.

Técnicas:



Ingeniería de Requisitos

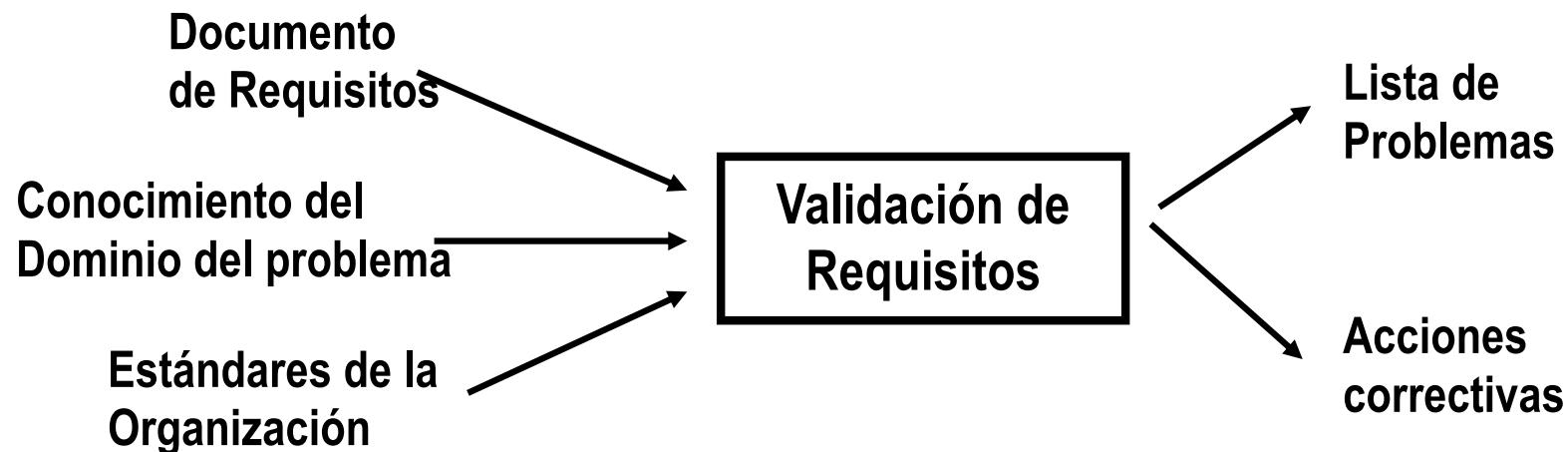
Proceso de IDR. Validación (1/2)

- Relacionadas con demostrar que los requisitos definen el sistema que el cliente realmente desea
- Los costos de errores en requisitos son muy elevados por lo que es una actividad importante
 - Reparar errores de requisitos después de entregar un sistema puede costar hasta 100 veces más que reparar un error de implementación
- **Técnicas:**
 - El *prototipado* es una técnica importante para la validación de requisitos
 - Revisiones o *Walkthroughs*
 - Auditorías
 - Matrices de trazabilidad

Ingeniería de Requisitos

Proceso de IDR. Validación (2/2)

- Es el último paso de el proceso de Ingeniería de Requisitos
- Validar los requisitos es “certificar” que éstos representan una descripción aceptable (correcta y completa) del sistema a desarrollar
- Principal problema: No hay un documento contra el cual validar los requisitos! (diseño vs. programa)



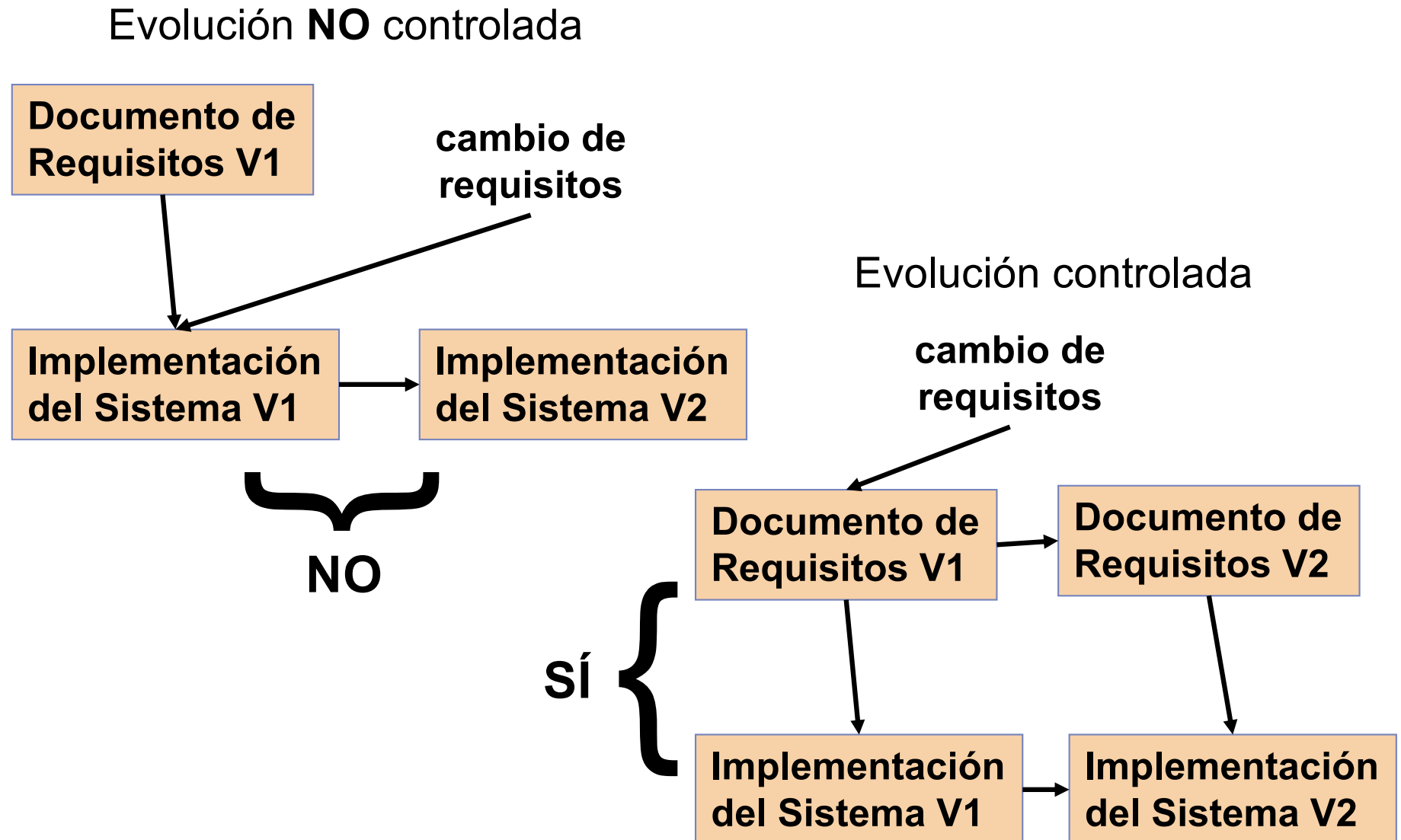
Ingeniería de Requisitos

Proceso de IDR. Gestión de Requisitos (1/3)

- Los requisitos *siempre* evolucionan
 - Mejor entendimiento del espacio del problema
 - Cambio en los objetivos de la organización
- Esta evolución/cambios pueden ocurrir:
 - Durante el desarrollo del sistema
 - En fase de producción
- El problema no es el cambio sino el *cambio incontrolado* por tanto es *esencial* planearlos (Gestión de Requisitos)

Ingeniería de Requisitos

Proceso de IDR. Gestión de Requisitos (2/3)



Ingeniería de Requisitos

Proceso de IDR. Gestión de Requisitos (3/3)

- La gestión de requisitos se asocia el proceso de gestión de cambios en los requisitos de un sistema
- Principales actividades:
 - Gestión de *cambios* de requisitos
 - Gestión de *relaciones* entre requisitos
 - Gestión de *dependencias* entre documentos de requisitos y otros documentos producidos durante el desarrollo (modelo conceptual, BD, LC...)
 - *Impacto del cambio*
- Los gestión de requisitos no puede ser gestionada adecuadamente sin TRAZABILIDAD

Ingeniería de Requisitos

Proceso de IDR. Gestión de Requisitos. Trazabilidad (1/3)

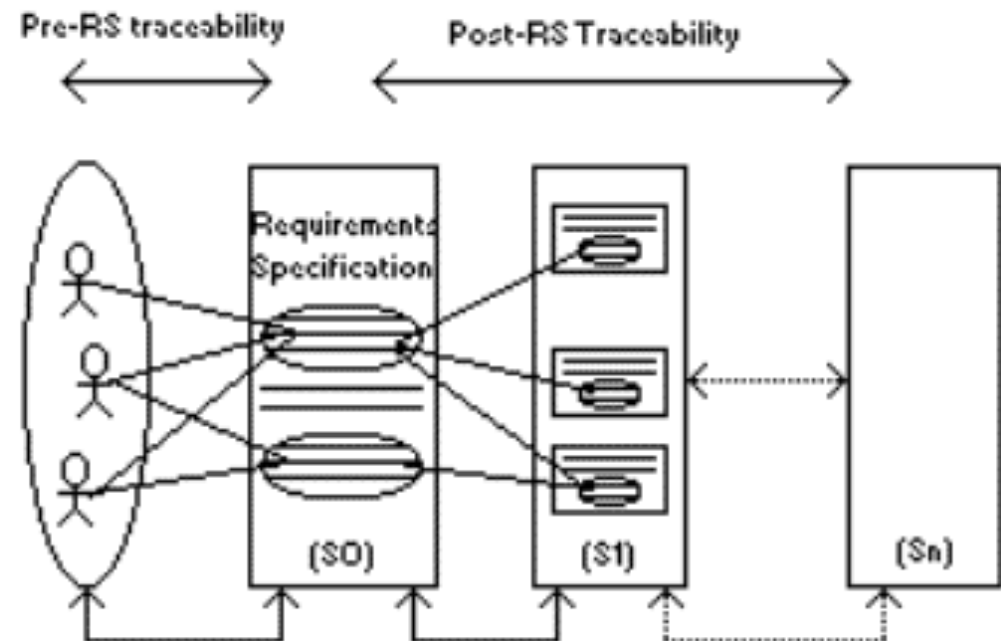
- Trazabilidad de Requisitos significa que requisitos relacionados están *conectados* de alguna forma
- La trazabilidad es una propiedad de la especificación de requisitos que refleja la facilidad de identificar requisitos relacionados
 - Los requisitos deben tener asignados *identificadores* únicos
 - Las *referencias cruzadas* relacionan requisitos usando estos identificadores
 - Producen una matriz para cada documento de requisitos mostrando requisitos relacionados. Muchas matrices pueden ser necesarias para representar diferentes tipos de relaciones
- Las herramientas CASE proveen facilidades de *soporte a la trazabilidad*

Ingeniería de Requisitos

Proceso de IDR. Gestión de Requisitos. Trazabilidad (2/3)

IEEE Std. 830:

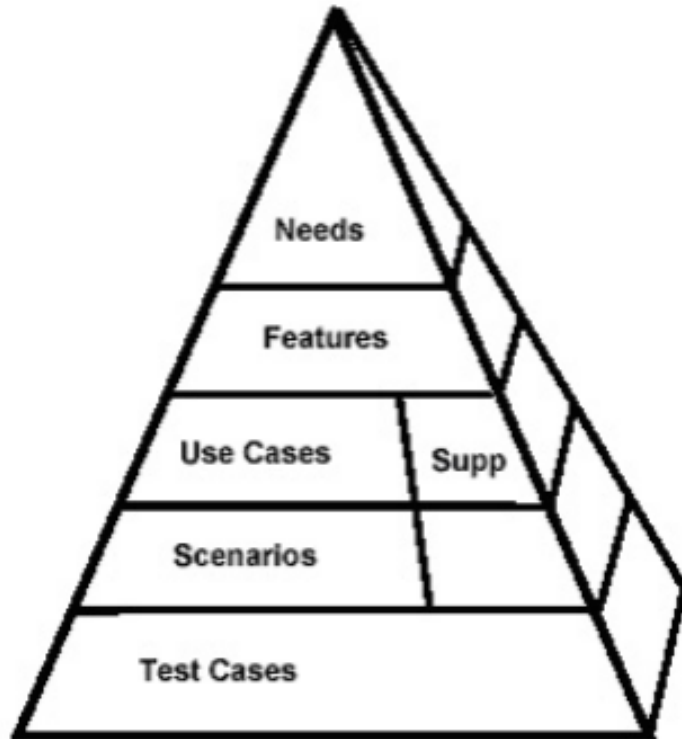
- **Backward traceability (Pre Trazabilidad):** conecta los requisitos con su origen en otros documentos
- **Forward traceability (Post Trazabilidad):** conecta los requisitos con componentes de diseño o implementación



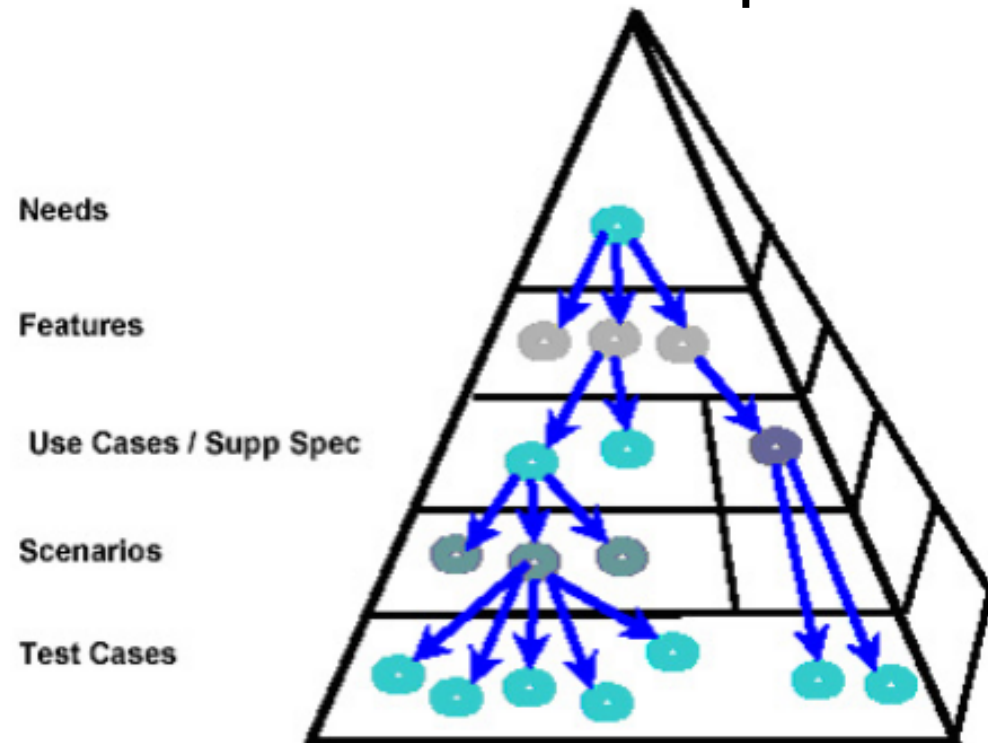
Ingeniería de Requisitos

Proceso de IDR. Gestión de Requisitos. Trazabilidad (3/3)

Pirámide de Requisitos



Trazabilidad en la Pirámide de Requisitos



Needs: Necesidades de usuarios o del negocio (objetivos)

Features (características). Funcionalidad de alto nivel. Agrupa funcionalidades más específicas

Use cases: Casos de Uso, son los Requisitos Funcionales

Supp: Requisitos No Funcionales (RNF) o Requisitos suplementarios

Scenarios: descripción de la funcionalidad de RF o casos de uso

Test cases: Pruebas que se realizan para comprobar los RF y RNF

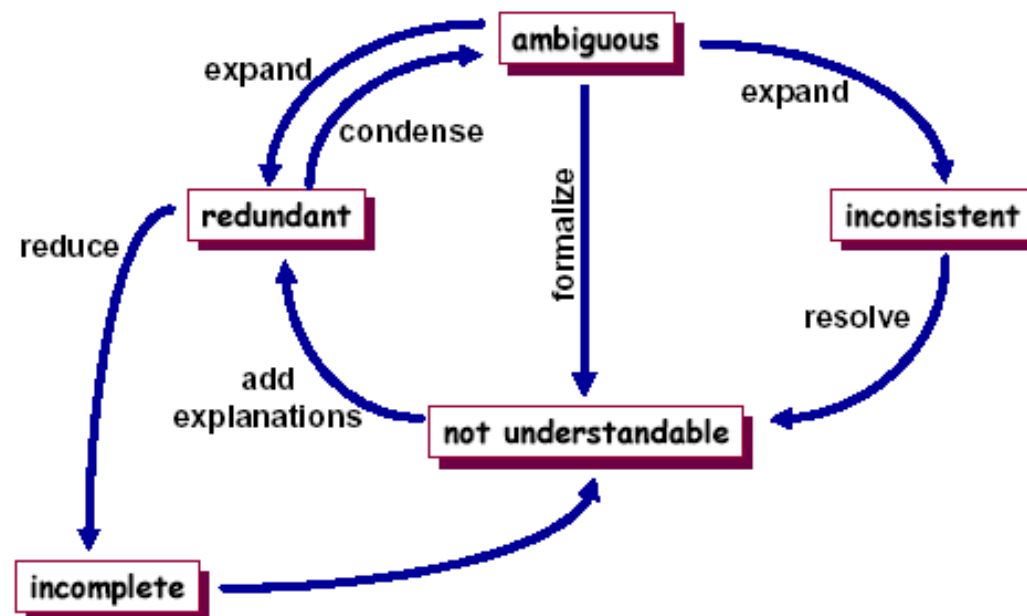
Contenido

- Procesos
 - Entradas y salidas
 - Variabilidad
- Proceso de IDR
 - Fases
 - Elicitación
 - Definición
 - Especificación
 - Validación
 - Gestión de requisitos
- ERS (Documentación – Estándares)
- Herramientas de soporte
- Mejora de procesos
 - Modelo de Madurez
 - Modelo de Madurez para la IDR

Ingeniería de Requisitos. ERS

El producto final de la IDR es un documento al que se denomina Especificación de Requisitos Software (ERS).

- Es una *descripción completa* de los objetivos y del comportamiento externo del sistema, es decir, *qué* debe hacer el software sin describir *cómo* debe hacerlo. (Qué vs Cómo).
 - su contenido difiere de *quién* la elabore y de la *finalidad* del software. Podrá ser más detallado o más general.



Ingeniería de Requisitos. ERS

ERS: grado de formalidad

Considerar 2 proyectos diferentes:

- a) Proyecto pequeño: **1 programador, 2 meses de trabajo. El programador habla con el cliente y escribe una especificación de 5 páginas**
- b) Proyecto grande: **50 programadores, 2 años de trabajo. Un equipo de analistas modelan los requisitos y los documentan en 500 páginas**

	Proyecto A	Proyecto B
Propósito	Reflejar el entendimiento del programador. Retroalimentación al cliente	Documento de lo que se tiene que desarrollar. Información para los diseñadores y programadores
Planificación y administración	La especificación es irrelevante. Los recursos están asignados	Se usa para estimar los recursos necesarios y planear el desarrollo
Lectores	Principal: el autor de la especificación Secundario: el cliente	Principal: diseñadores, programadores, testers, gestores Secundario: clientes

Ingeniería de Requisitos. ERS

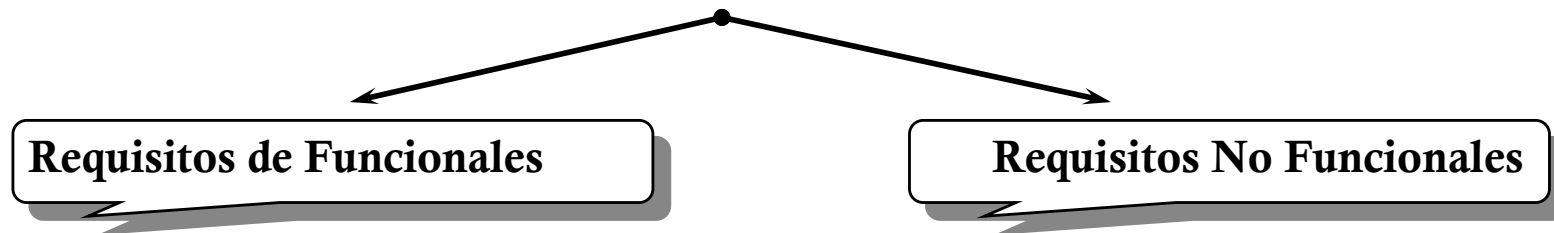
- Propósito de la ERS:
 - Comunicación: establecer un mecanismo de comunicación entre los clientes, usuarios, analistas y diseñadores.
 - Contractual: expresa un acuerdo y un compromiso (contrato legal)
 - Evaluación: soportar todas las actividades de *pruebas* sobre el sistema.
 - Control de cambios: la evolución del sistema durante su desarrollo y posteriormente en su mantenimiento.

Ingeniería de Requisitos. ERS contenido

¿ Qué debe incluirse en una ERS ?

Una descripción concisa de toda la interfaz externa del sistema con su entorno, incluyendo otro software, puertos de comunicación, hardware y usuarios.

Básicamente existen dos TIPOS de requisitos:



Ingeniería de Requisitos. ERS contenido

Requisitos funcionales y no funcionales

- Requisitos Funcionales (RF)

Describen la *funcionalidad* o los **servicios** que se espera que el sistema proveerá, sus entradas y salidas (cómo las entradas se transforman por el sistema en salida).

Ejemplos:

- 1.- “El usuario podrá registrar los datos de los nuevos contratos de alquiler”
- 2.- “El sistema permitirá eliminar la información de cuentas que ya no tienen movimientos”

Ingeniería de Requisitos. ERS contenido

Requisitos funcionales y no funcionales

- Requisitos No Funcionales (RNF)

Se refieren a las *propiedades emergentes* (atributos) del sistema como la fiabilidad, el tiempo de respuesta, la capacidad de almacenamiento, la capacidad de los dispositivos de entrada/salida, ajuste a estándares, etc.

Ejemplos:

- 1.- “Será necesario que la comunicación requerida entre el dispositivo APSE y el usuario se pueda expresar utilizando el conjunto de caracteres estándar de ADA.”
- 2.- “El proceso de desarrollo del sistema y los documentos a entregar estarán sujetos al proceso y a los productos a entregar definidos en XYZCo-SP-STAN-95.”
- 3.- “El sistema no deberá revelar a sus operadores ninguna información personal de los clientes excepto su nombre y número de referencia.”

Ingeniería de Requisitos. ERS contenido

¿ Qué NO debe incluirse en una ERS ?

No debe incluirse la siguiente información:

- ***Requisitos de proyecto:*** Tales como personal necesario, planificación del proyecto, costes, actividades, fases, ...
- ***Diseños:*** Una ERS no debe entrar en ningún detalle de diseño, excepto, en todo caso, en restricciones de diseño.
- ***Planes de garantía del producto:*** Tales como planes para evaluar la calidad del software, planes de verificación y validación, ...

Ingeniería de Requisitos. ERS atributos

Atributos de una ERS

Algunas características deseables para una buena ERS [IEEE 1984b]:

- **No ambigua**
- **Completa**
- **Fácil de verificar**
- **Consistente**
- **Fácil de modificar**
- **Trazable**: incluir referencias de los requisitos hacia su origen y su representación en artefactos de diseño e implementación
- **Identificable**: fácil de encontrar requisitos concretos en el documento
- **Categorizada**: incluir información sobre prioridad, estabilidad,...
- **Fácil de utilizar**: durante la fase de explotación y mantenimiento
- **Independiente de implementación**: no hacer referencia a aspectos de diseño o implementación

Ingeniería de Requisitos. ERS estándares (IEEE/ANSI 830-1998)

IEEE/ANSI Std. 830-1998

- Buen punto de partida para la descripción de requisitos sw.
- Estructura:



Table of Contents

1. Introduction

- 1.1 Purpose
- 1.2 Scope
- 1.3 Definitions, acronyms, and abbreviations
- 1.4 References
- 1.5 Overview

2. Overall description

- 2.1 Product perspective
- 2.2 Product functions
- 2.3 User characteristics
- 2.4 Constraints
- 2.5 Assumptions and dependencies

3. Specific requirements (See 5.3.1 through 5.3.8 for explanations of possible specific requirements. See also Annex A for several different ways of organizing this section of the SRS.)

Appendixes

Index

Ingeniería de Requisitos. ERS estándares (IEEE/ANSI 830-1998)

IEEE/ANSI Std. 830-1998



- Los *Requisitos Específicos* pueden organizarse:
 - Modo de operación: normal, entrenamiento, emergencia, ...
 - Tipo de Usuario: administrador, empleado, ...
 - (Clases de) objetos: atributos y servicios
 - Aspecto (feature): tipo de servicio que requiere unas entradas
 - Estímulo: por funcionalidad en términos de estímulos
 - Respuesta: toda la funcionalidad en función de un tipo de respuesta (ej. generar nómina)
 - Jerarquía funcional: organizada por input, output o datos comunes (pueden acompañarse con DFDs y DER)

Ingeniería de Requisitos. ERS estándares (IEEE/ANSI 830-1998)

A.3 Template of SRS Section 3 organized by user class

- 3. Specific requirements
 - 3.1 External interface requirements
 - 3.1.1 User interfaces
 - 3.1.2 Hardware interfaces
 - 3.1.3 Software interfaces
 - 3.1.4 Communications interfaces
 - 3.2 Functional requirements
 - 3.2.1 User class 1
 - 3.2.1.1 Functional requirement 1.1
 - .
 - .
 - .
 - 3.2.1.*n* Functional requirement 1.*n*
 - 3.2.2 User class 2
 - .
 - .
 - .
 - 3.2.*m* User class *m*
 - 3.2.*m*.1 Functional requirement *m*.1
 - .
 - .
 - .
 - 3.2.*m*.*n* Functional requirement *m*.*n*
 - 3.3 Performance requirements
 - 3.4 Design constraints
 - 3.5 Software system attributes
 - 3.6 Other requirements



Ingeniería de Requisitos. ERS estándares (IEEE/ANSI 830-1998)

A.4 Template of SRS Section 3 organized by object

- 3. Specific requirements
 - 3.1 External interface requirements
 - 3.1.1 User interfaces
 - 3.1.2 Hardware interfaces
 - 3.1.3 Software interfaces
 - 3.1.4 Communications interfaces
 - 3.2 Classes/Objects
 - 3.2.1 Class/Object 1
 - 3.2.1.1 Attributes (direct or inherited)
 - 3.2.1.1.1 Attribute 1
 - .
 - .
 - 3.2.1.1.*n* Attribute *n*
 - 3.2.1.2 Functions (services, methods, direct or inherited)
 - 3.2.1.2.1 Functional requirement 1.1
 - .
 - .
 - .
 - 3.2.1.2.*m* Functional requirement 1.*m*
 - 3.2.1.3 Messages (communications received or sent)
 - 3.2.2 Class/Object 2
 - .
 - .
 - .
 - 3.2.*p* Class/Object *p*
 - 3.3 Performance requirements
 - 3.4 Design constraints
 - 3.5 Software system attributes
 - 3.6 Other requirements

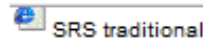


Ingeniería de Requisitos. ERS estándares

- **IEEE 12207-2008**, Systems and Software Engineering — Software life cycle processes
 - Define un framework común para procesos de ciclos de vida de sistemas, productos software y servicios
 - Se aplica a la adquisición (compra), desarrollo y mantenimiento
 - Incluye una terminología, las actividades, artefactos (y su estructura básica)
 - En particular, aquellos aspectos de la definición del sistema necesarios para proveer el contexto de los productos y servicios
- **IEEE-Std-1233-1998**, Guide for Developing System Requirements Specifications
 - Similar al estándar IEEE/ANSI 830-1998 pero orientado al sistema (en general, no sólo Sw)
 - Incluye la identificación, organización, presentación y modificación de requisitos

Ingeniería de Requisitos.

ERS estándares (RUP – ERS tradicional)



<Project Name> Software Requirements Specification For <Subsystem or Feature>

Sección: Funcionalidades

- La sección se organiza por subsistemas y características
- Los requisitos funcionales se capturan como Casos de Uso (numerados) expresados en lenguaje natural
- Se puede usar una plantilla asociada a cada caso de uso

1.	Introduction
1.1	Purpose
1.2	Scope
1.3	Definitions, Acronyms and Abbreviations
1.4	References
1.5	Overview
2.	Overall Description
3.	Specific Requirements
3.1	Functionality
3.1.1	<Functional Requirement One>
3.2	Usability
3.2.1	<Usability Requirement One>
3.3	Reliability
3.3.1	<Reliability Requirement One>
3.4	Performance
3.4.1	<Performance Requirement One>
3.5	Supportability
3.5.1	<Supportability Requirement One>
3.6	Design Constraints
3.6.1	<Design Constraint One>
3.7	Online User Documentation and Help System Requirements
3.8	Purchased Components
3.9	Interfaces
3.9.1	User Interfaces
3.9.2	Hardware Interfaces
3.9.3	Software Interfaces
3.9.4	Communications Interfaces
3.10	Licensing Requirements
3.11	Legal, Copyright and Other Notices
3.12	Applicable Standards
4.	Supporting Information

Ingeniería de Requisitos.

ERS estándares (ESA PSS-05-03)

- European Space Agency (ESA)
- ESA PSS-05-0x describe los estándares para todo sobre desarrollado o adquirido por la Agencia Espacial Europea (ESA)
- ESA PSS-05-03 proporciona una guía de cómo producir los requisitos del software y documentarlos
- Web de descarga:
http://www.esa.int/TEC/Software_engineering_and_standardisation/TECBUCUXBQE_2.html

1 INTRODUCTION

- 1.1 Purpose
- 1.2 Scope
- 1.3 Definitions, acronyms and abbreviations
- 1.4 References
- 1.5 Overview

2 GENERAL DESCRIPTION

- 2.1 Relation to current projects
- 2.2 Relation to predecessor and successor projects
- 2.3 Function and purpose
- 2.4 Environmental considerations
- 2.5 Relation to other systems
- 2.6 General constraints
- 2.7 Model description

3 SPECIFIC REQUIREMENTS

(The subsections may be regrouped around high-level functions)

- 3.1 Functional requirements
- 3.2 Performance requirements
- 3.3 Interface requirements
- 3.4 Operational requirements
- 3.5 Resource requirements
- 3.6 Verification requirements
- 3.7 Acceptance testing requirements
- 3.8 Documentation requirements
- 3.9 Security requirements
- 3.10 Portability requirements
- 3.11 Quality requirements
- 3.12 Reliability requirements
- 3.13 Maintainability requirements
- 3.14 Safety requirements

4 REQUIREMENTS TRACEABILITY MATRIX

Ingeniería de Requisitos. ERS estándares

Documento ERS utilizado en la Administración Pública de Canadá

1.0	Introduction	1
1.1	Purpose	1
1.2	Scope	1
2.0	System Overview	2
3.0	Domain Model	3
3.1	Class Diagrams	3
3.2	Class Specifications	3
4.0	Throw-Away Prototyping	3
5.0	Requirements	4
5.1	Use Case Requirements	4
5.1.1	Actor List	4
5.1.2	Use Case Diagrams	5
5.1.3	Use Case Specifications	5
5.1.4	Use Case Standard Template	7
5.2	Business Rules	8
5.3	Non-Functional Requirements	8
5.3.1	System Requirements	8
5.3.2	Usability Requirements	8
5.3.3	Performance Requirements	9
5.3.4	Security Requirements	9
5.3.5	Delivery Requirements	9
5.3.6	Legal Requirements	9
5.3.7	Interoperability Requirements	9
5.3.8	Scalability Requirements	9
5.4	Interface Requirements	9
5.4.1	Machine Interfaces	9
5.4.2	External System Interfaces	10
5.4.3	Human-Computer Interface Considerations	10
5.4.4	Input and Output Requirements	10

Contenido

- Procesos
 - Entradas y salidas
 - Variabilidad
- Proceso de IDR
 - Fases
 - Elicitación
 - Definición
 - Especificación
 - Validación
 - Gestión de requisitos
- ERS (Documentación – Estándares)
- Herramientas de soporte
- Mejora de procesos
 - Modelo de Madurez
 - Modelo de Madurez para la IDR

Herramientas de soporte a la IDR

- Herramientas de soporte al proceso **IS** desde finales de los 80. Más orientados al soporte de la programación, análisis y diseño
- Tipos de herramientas para el soporte al proceso de **IDR**

- Herramientas de modelado y validación

Para métodos estructurados (tipo SADT) o lenguajes de especificación de requisitos. Incluyen verificación de consistencia.

Usando métodos formales (Z, VDM,...) se verifican además inconsistencias matemáticas

- Herramientas de gestión de repositorios de requisitos

Además de facilidades para almacenamiento y recuperación proveen mecanismos para la gestión de cambios.

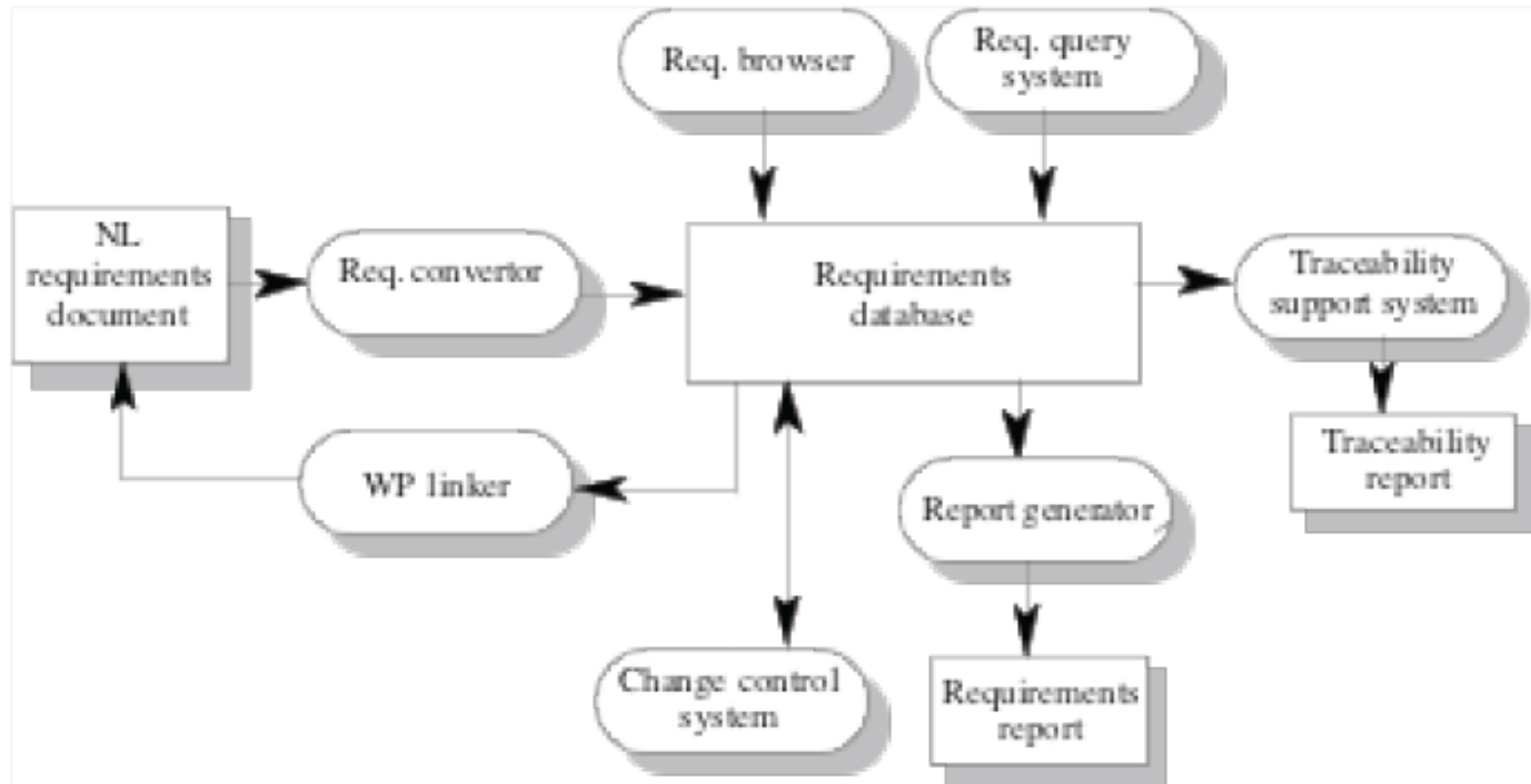
DOORS, AnalystPro, Requisite Pro. Normalmente incluyen interfaces a procesadores de textos, planillas, ...

Herramientas de soporte a la IDR

- Las herramientas de soporte a la IDR deben proveer facilidades tales como:
 - Sistema de *repositorio* para:
 - almacenar y recuperar requisitos
 - consultar requisitos
 - *Análisis y validación* de requisitos y generación de documentos (informes, vistas, consistencia)
 - Facilidades para la gestión de cambios (*análisis de impacto*)
 - Facilidades de *trazabilidad* y ayuda para encontrar dependencias
 - Integración con documentos del origen de requisitos (*pre-trazabilidad*) y con documentos de diseño e implementación (*post-trazabilidad*)
 - Control de cambios
 - ...

Herramientas de soporte a la IDR.

- Un sistema genérico de soporte para la gestión integral de requisitos



Herramientas de soporte a la IDR. Analyst Pro

- Una de las herramientas más conocidas del mercado (DOORS, RequisitePro, ...)
- *Documentación y comunicación* en cualquier proyecto software, incluyendo: gestión de requisitos, análisis, desarrollo de especificaciones, desarrollo y gestión de casos de prueba y procesos de negocio
- La herramienta va dirigida a:
 - analistas de sistemas, jefes de proyecto, programadores y probadores (testers) de software
- Se basa en un modelo de representación de requisitos donde cada requisito queda caracterizado por una serie de *atributos*. Además, cada requisito puede referenciar a otros (*trazabilidad*)

Herramientas de soporte a la IDR. Analyst Pro

Tres tipos de requisitos:

- **Requisitos de la aplicación:** incluye los requisitos globales de la aplicación, como, p.e., estándares en la presentación, seguridad, interfaces, requisitos del sistema, etc.
- **Requisitos del negocio:** incluye los requisitos propios del negocio para el cual se está desarrollando la aplicación. Es independiente del diseño e implementación del sistema.
- **Requisitos de diseño:** incluye los requisitos asociados a objetos concretos de diseño, como ventanas, formularios, páginas web, etc.

Herramientas de soporte a la IDR. Analyst Pro

1. Se crea un proyecto, grupos de trabajo según el perfil (programador, analista, testeador, ...), y se definen los participantes asignándolos a un grupo:

Project Details

File Edit Projects View Action Tools Help

◀ ◁ ▷ ▶ ➕ Add ✓ Save ✖ Delete ✕ Cancel

Project Info **Users** Exit

Users: Create/View User Groups

	Last Name	MI	First Name	Login User Name	Given Password	User Group
▶	John	K	Miller	KMiller	demo	
■	Han	W	Train	HanW	han	Programmer's Group

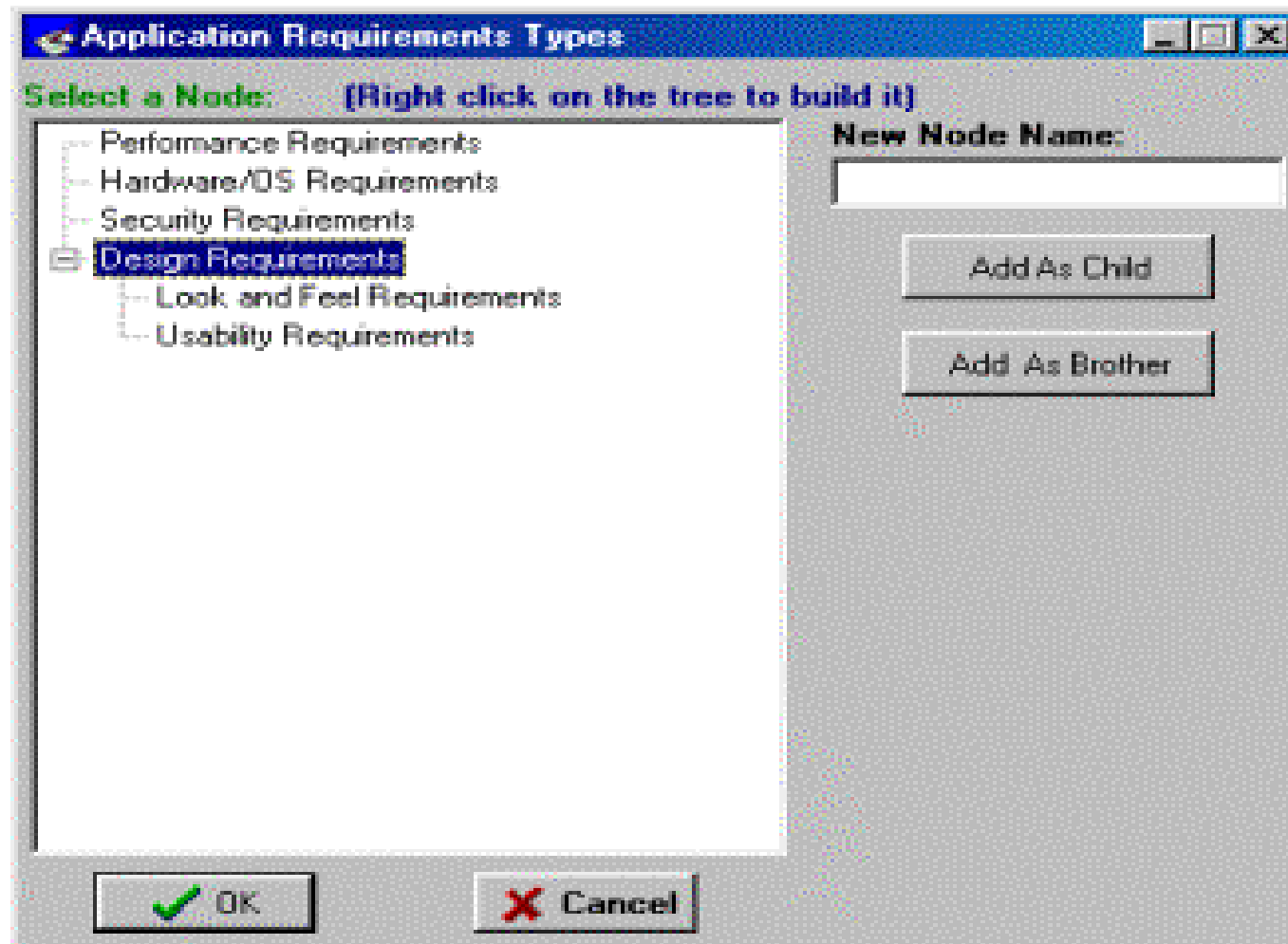
User Last Name: John Phone: 7034347888 Email: john@analysttool.com

Reset Password

Read/Write Project: Human Resources Automation

Herramientas de soporte a la IDR. Analyst Pro

2. Se crea una jerarquía de requisitos para cada tipo de requisitos:



Herramientas de soporte a la IDR. Analyst Pro

3. Se introducen los requisitos (aplicación, negocio y diseño):

The screenshot displays the 'Requirements' window in Analyst Pro, specifically the 'Requirements Analysis' tab. The window features a menu bar (File, Edit, View, Action, Tools, Reports, Help) and a toolbar with buttons for navigation and actions like Add, Save, Delete, Cancel, Delegate, Assign, and Exit. Below the toolbar, there are four tabs: 'Requirements Analysis', 'Application Requirements', 'Process Requirements', and 'Design Requirements'. The 'Requirements Analysis' tab is active, showing a form for entering requirement details. The form includes fields for 'Req Type' (Performance), 'Req Status' (Assigned), 'Req Priority' (High), and 'Version' (1.5.1). To the right, there are fields for 'Req Tag' (A 4 1), 'Date' (05/29/2001), 'Req Author' (Train Han), 'Req Source' (Customer review), 'Parent Tag' (empty), and 'Ref & Links' (System Requirements Document). A 'Notes' section is also present at the bottom right. The bottom status bar shows 'Read/Write' and 'Project: Human Resources Automation'.

Requirements

File Edit View Action Tools Reports Help

◀ ◁ ▷ ▶ Add Save Delete Cancel Delegate Assign 1 of 4 Exit

Requirements Analysis Application Requirements Process Requirements Design Requirements

Req Type: Performance

Req Status: Assigned

Req Priority: High

Version: 1.5.1

Req Tag: A 4 1 Date: 05/29/2001

Req Author: Train Han

Req Source: Customer review

Parent Tag: ...

Ref & Links: System Requirements Document ...

Notes:

Requirement(s): (Double Click to Launch Word)

All forms in the application shall load within 3 seconds when using a Pentium 266 MHz (or higher) with 32 MB RAM.

Read/Write Project: Human Resources Automation

Herramientas de soporte a la IDR. Analyst Pro

4. A continuación, se definen los objetos de diseño (clases, módulos, subsistemas, ficheros, ...).

Los requisitos se asocian a estos objetos.

5. A partir de aquí, comienza el desarrollo de la aplicación.

Un típico ejemplo del proceso que puede seguir un requisito:

Atributo <i>Status</i>	Proceso de trabajo
	El jefe de proyecto aprueba un requisito, y asigna el valor "Aprobado" al atributo <i>status</i>
Aprobado ↓	El jefe o el analista asigna el trabajo a un programador, y cambia el valor a "Asignado"
Asignado ↓	Tras acabar el trabajo, el programador cambia el valor a "Revisión del jefe"
Revisión del jefe ↓	El jefe revisa el trabajo y lo asigna a un testeador, cambiando el valor a "Pruebas"
Pruebas ↓	El testeador acaba el trabajo, y después cambia el valor a "Revisión del jefe"
Revisión del jefe ↓	El jefe revisa los resultados de las pruebas, y cambia el valor a "Finalizado"
Finalizado	

Herramientas de soporte a la IDR. Analyst Pro (Pruebas)

Specs & Testing

File Edit Action Reports Help

Add Save Delete Cancel

Allocated Object Requirements

Design Object: PayReport

Exit

Specifications **Object Testing**

1.0 TESTING PROCEDURES:

Unit Testing Procedure:
(Double Click to Launch Word)

1. Verify the calculated total for an employee pay, if employee works:
a) Less than 40 hrs/week
b) Equal to 40 hrs/ week
c) Greater than 40 hrs/week

2. Verify that the report is printed in landscape format.

Test Results:
(Double Click to Launch Word)

Integration or System Testing:
(Double Click to Launch Word)

Verify that only Human Resources personnel, a Vice President, or the Company President can access and generate reports.

2.0 SECURITY ISSUES:

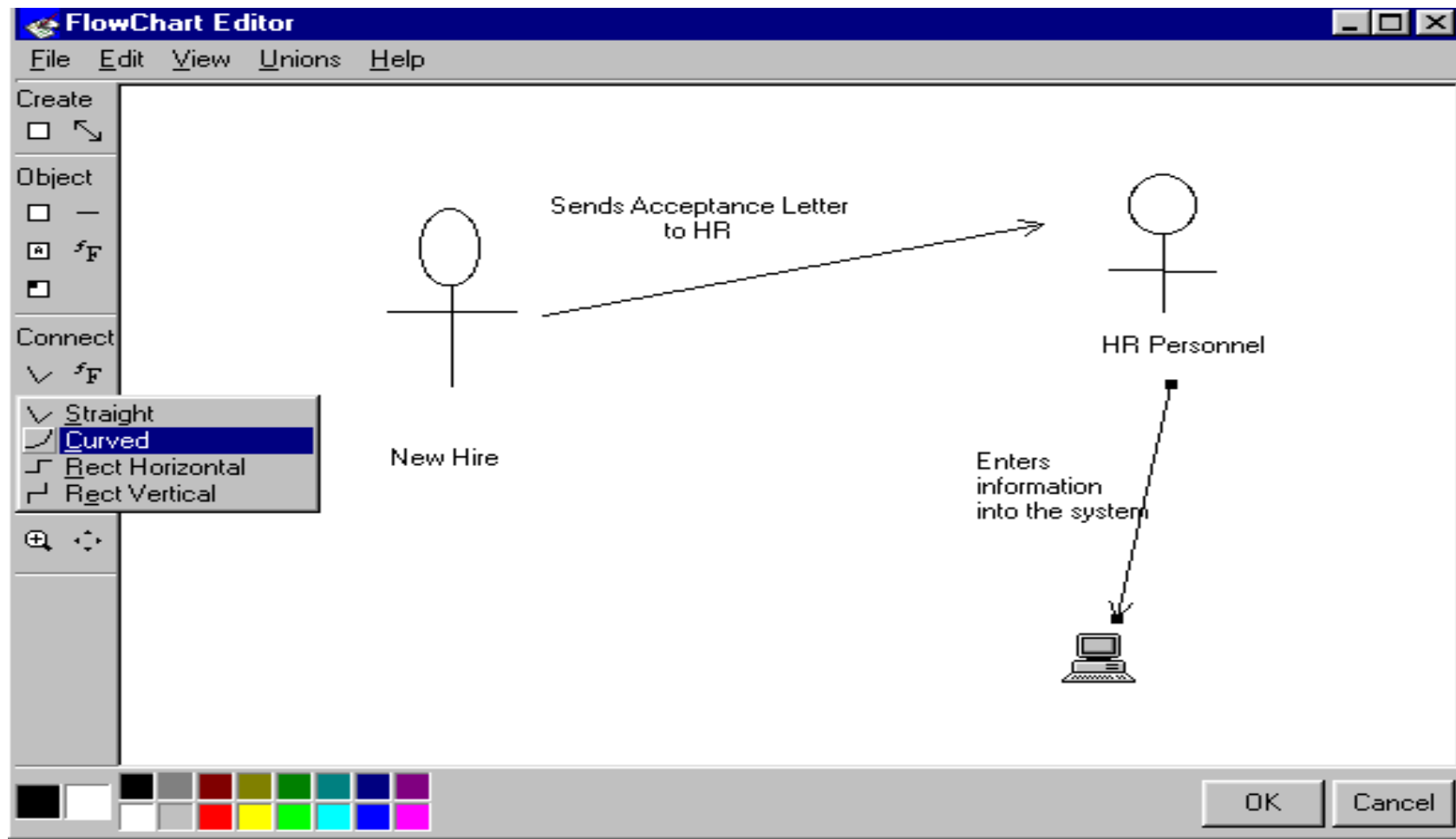
Generate Test Report

Read/Write Project: Human Resources Automation

Herramientas de soporte a la IDR. Analyst Pro

Especificación de procesos de negocio

- Analyst permite crear FlowCharts, PERO no existe ningún modo de importar diagramas UML (validación, simulación, etc)



Contenido

- Procesos
 - Entradas y salidas
 - Variabilidad
- Proceso de IDR
 - Fases
 - Elicitación
 - Definición
 - Especificación
 - Validación
 - Gestión de requisitos
- ERS (Documentación – Estándares)
- Herramientas de soporte
- Mejora de procesos
 - Modelo de Madurez
 - Modelo de Madurez para la IDR

Mejora de Procesos

- La mejora de procesos se refiere a la ***modificación*** de procesos para alcanzar un **objetivo de mejora**.
- Algunos **posibles objetivos de mejora**:
 - Mejora de la calidad
 - Reducción de tareas (trabajos duran menos tiempo)
 - Reducción de recursos (menos personal, equipos de trabajo más pequeños...)
- Algunos **posibles problemas** en procesos de IDR
 - Falta de participación de *stakeholders*
 - Necesidades del negocio no son tenidos en cuenta
 - Falta de gestión de requisitos
 - Problemas de comunicación con los *stakeholders*
 - Planificaciones muy largas y poca calidad de documentos de requisitos
 - ...

Mejora de Procesos

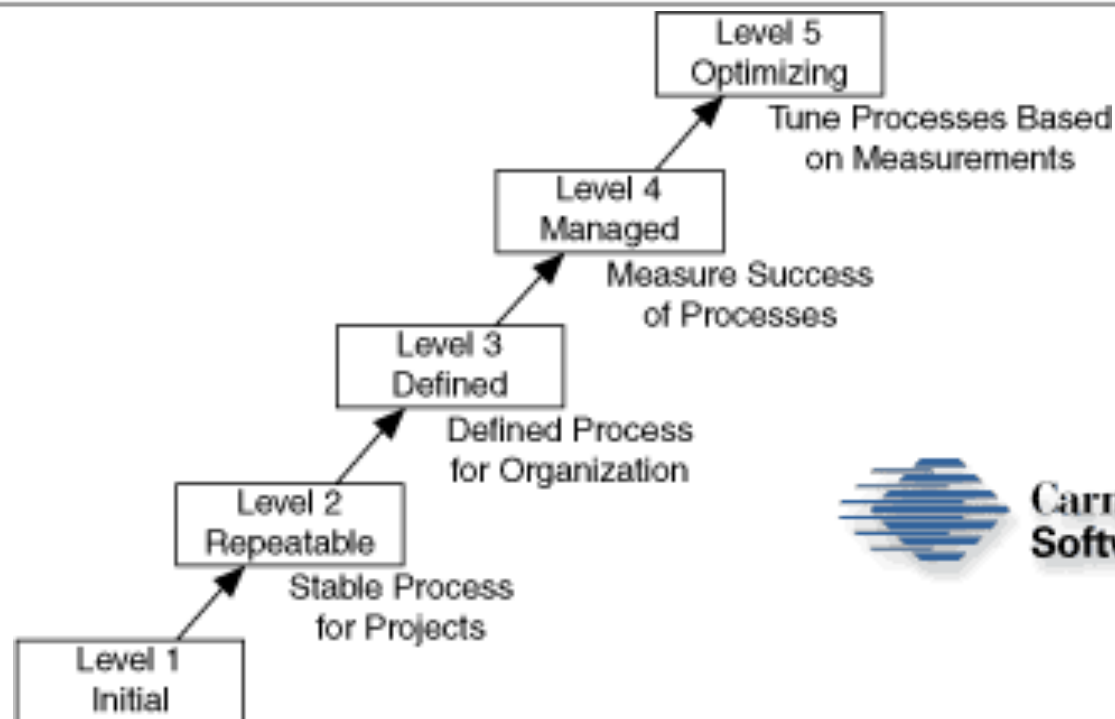
- No existe un estándar de mejora de procesos aplicable a todos los casos y a todos los contextos.
- Cada mejora se debe aplicar dependiendo:
 - Del tipo de organización (PYME, corporación...)
 - De la cultura de la organización (estricta, *freelance*, ...)
 - Del tipo de producto software a producir (empotrado, a medida,...)
 - ...

Mejora de Procesos. Modelo de Madurez

- Madurez de proceso se refiere al nivel de una organización con respecto a:
 - Definir sus procesos
 - Controlar activamente sus procesos
 - Proveer soporte sistemático y automatizado a esos procesos
- Por tanto, una organización que define un conjunto de estándares de proceso y provee herramientas de soporte es **más madura** que una que realiza actividades informales.
- El Modelo de Madurez del Software Engineering Institute (**SEI**) (**Capability Maturity Model - CMM**) es un framework para evaluar procesos de desarrollo de software en empresas.

Mejora de Procesos. Modelo de Madurez CMM. Características

1. Inicial	El proceso software está caracterizado como 'ad hoc', y en ocasiones puede ser incomprensible. Algunos procesos están definidos y el éxito depende de los esfuerzos a nivel de individuo.
2. Repetible	Los procesos de gestión de proyectos están definidos de una manera básica para realizar el seguimiento de los costes, fechas y funcionalidad. El rigor en la definición de los procesos es el justo para poder repetir éxitos previos en proyectos de similares características.
3. Definido	El proceso software para las actividades de gestión e ingeniería está documentado, estandarizado e integrado en el proceso estándar dentro de la organización. Todos los proyectos utilizan una versión estándar del proceso software aprobado por la organización y adaptado a las necesidades del proyecto para desarrollo y mantenimiento de software.
4. Gestionado	Se recogen de forma detallada medidas de los procesos software y la calidad de los productos. Los procesos y productos software son entendidos cuantitativamente y controlados.
5. Optimizado	La mejora continua de procesos se basa en los resultados cuantitativos de la aplicación de innovaciones y tecnologías en los procesos ya establecidos.



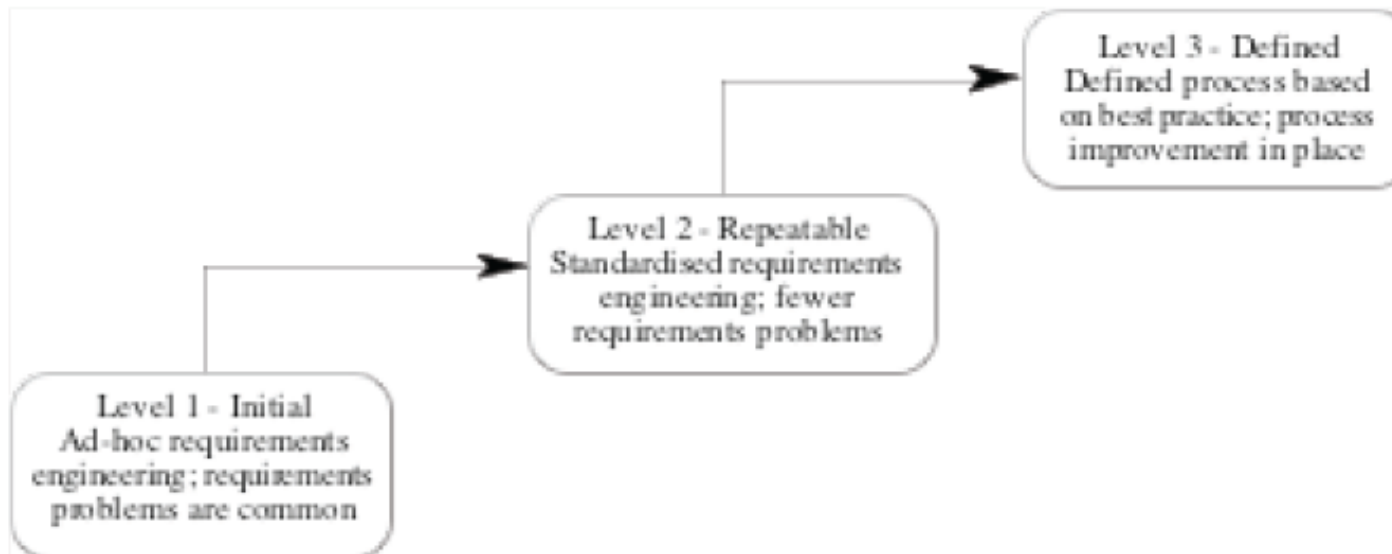
Carnegie Mellon
Software Engineering Institute

Mejora de Procesos. Modelo de Madurez CMM (Capability Maturity Model). Áreas

LEVEL		FOCUS	PROCESS AREAS
CMMI-DEV v1.2 Staged Representation	5 Optimizing	Continuous Process Improvement	Organizational Innovation and Deployment Causal Analysis and Resolution
	4 Quantitatively Managed	Quantitatively Managed	Organizational Process Performance Quantitatively Project Managed
	3 Defined	Process Standardization	Requirements Development Technical Solution Product Integration Verification Validation
			Organizational Process Focus Organizational Process Definition Organizational Training Integrated Project Management Risk Management Decision Analysis and Resolution
	2 Managed	Basic Project Management	Requirements Management Project Planning Project Monitoring and Control Supplier Agreement Management Measurement and Analysis Process and Product Quality Assurance Configuration Management
	1 Initial		

Modelo de Madurez para Procesos de IDR (Sommerville-Sawyer)

- Modelo de 3 niveles propuesto por Sommerville y Sawyer en 1997
- Nivel 1 (**Inicial**) y Nivel 2 (**Repetible**) son comparables con los niveles 1 y 2 de CMM
- Nivel 3 (**Definido**) implica un programa de mejora de proceso activo que realiza evaluaciones de nuevos métodos y técnicas aplicables a sus proyectos



Modelo de Madurez para Procesos de IDR (IBM Software)

- Modelo de 5 Niveles para la IDR propuesto por IBM Software
 1. Requirements are Written
 2. Requirements are Organized
 3. Requirements are Structured
 4. Requirements are Traced
 5. Requirements are Integrated with other lifecycle tools and artifacts

Modelo de Madurez para Procesos de IDR (IBM Software)

Level 0 - No Requirements

- Are there ever really no requirements?
- Problems occur when requirements are not:
 - ▶ Agreed upon
 - ▶ Communicated
 - ▶ Used consistently
- The process becomes "code and fix"

Modelo de Madurez para Procesos de IDR (IBM Software)

Level 1 - Requirements are Written

■ Possible documents:

- ▶ Software Requirements Specification
- ▶ Product Requirements Specification
- ▶ Customer Requirements Specification
- ▶ Use cases
- ▶ User stories
- ▶ Many others

■ Possible "tools":

- ▶ Word
- ▶ Excel
- ▶ Homegrown database
- ▶ HTML
- ▶ Requirements Management tool
- ▶ These don't count:
 - Whiteboard
 - Post-it notes
 - Paper napkins
 - Paper documents

Modelo de Madurez para Procesos de IDR (IBM Software)

Level 1 - Benefits

- Better understanding of requirements
- Broader communication can occur
- A contract with the customer is possible
- Testers start to have an idea of what to test
- New staff have a clue about what the project is supposed to do
- Back-up and restore of the requirements is possible (reduce risk)

Modelo de Madurez para Procesos de IDR (IBM Software)

Level 2 - Requirements are Organized

- Requirements formatted consistently
- Central location for requirements documents
- Requirements have history
- Security
- Requirements are "good" (well written)
 - ▶ Unambiguous, testable, don't conflict, etc.

Modelo de Madurez para Procesos de IDR (IBM Software)

Level 2 - Benefits

- Requirements are easier to read
- Save time searching for documents
- No rework due to working on the wrong spec
- Better understanding of when and why a change was made
- More trust that requirements are accurate
- Requirements more usable by the development team

Modelo de Madurez para Procesos de IDR (IBM Software)

Level 3 - Requirements are Structured

- Requirements have types
 - ▶ Functional
 - ▶ Non-functional
 - ▶ Business
 - ▶ Feature
 - ▶ User
 - ▶ Software
 - ▶ Process
 - ▶ Many others...
- Requirements have attributes
 - ▶ Priority, status, stability, etc.
- Tool support in place
 - ▶ Searching
 - ▶ Sorting
 - ▶ Filtering



Modelo de Madurez para Procesos de IDR (IBM Software)

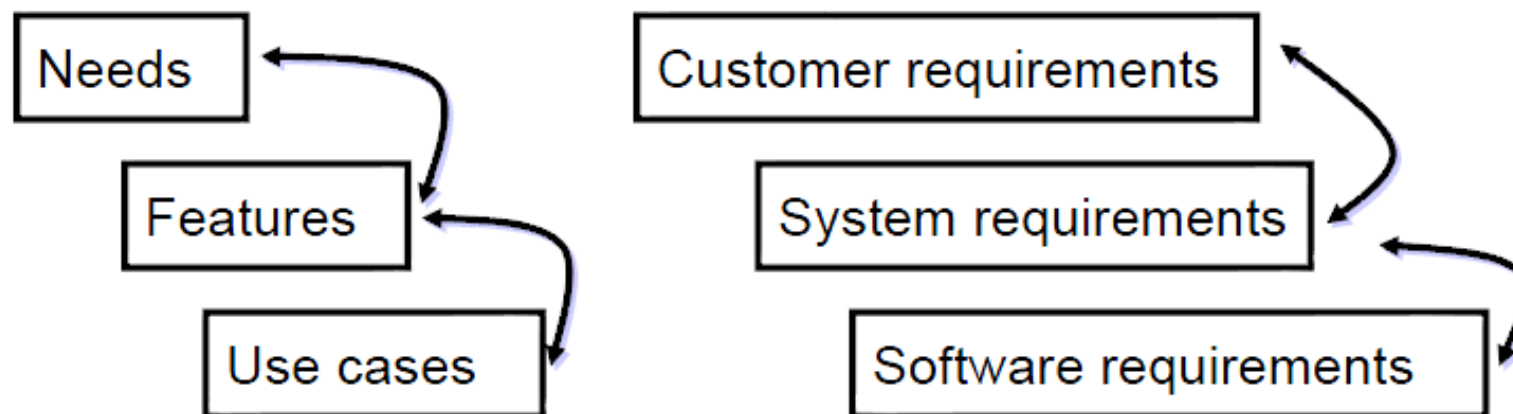
Level 3 - Benefits

- Decisions are easier to make
 - ▶ You have more information in a useable form
- Decisions are more objective
 - ▶ "Show me all of the High Priority features that are architecturally significant"
- Requirements types make for better understanding
 - ▶ Different types of requirements no longer mixed up
- Easier to find requirements with common attributes
 - ▶ "Show me all of the features in Release 1.0"

Modelo de Madurez para Procesos de IDR (IBM Software)

Level 4 - Requirements are Traced

- Keep track of:
 - Rationale for requirements
 - Source of requirements
 - How requirements are derived from each other



Modelo de Madurez para Procesos de IDR (IBM Software)

Level 4 - Benefits

- Reduce work/rework
 - ▶ Understand origin of, and rationale for, requirements - don't have the same argument twice
 - ▶ Reduce "gold plating"
- Make project management and planning better
 - ▶ Understand the impact of a change (impact analysis)
 - ▶ Ensure developers build what they are supposed to (coverage analysis)
 - ▶ Manage scope
 - ▶ Increase efficiency - know who to notify if a change happens
- Increase quality
 - ▶ Ensure that decisions made later in the process/project are consistent with those made earlier
 - ▶ Make sure that changes propagate
 - ▶ Ensure systems meets stakeholder's needs

Modelo de Madurez para Procesos de IDR (IBM Software)

Level 5 - Requirements are Integrated With...

- Design
 - ▶ Visual modeling
 - ▶ IBM Rational Rose and IBM Rational XDE
- Change management
 - ▶ Defect/enhancement request tracking
 - ▶ IBM Rational ClearQuest
- Testing
 - ▶ IBM Rational TestManager
- Project management
 - ▶ IBM Rational Project Console

Modelo de Madurez para Procesos de IDR (IBM Software)

Level 5 - Benefits

- Everything is traceable
 - ▶ Nothing gets missed
- Better communication
- Fewer mistakes
 - ▶ Easier to see holes and gaps
- Project status easier to assess
- Risks easier to identify

Modelo de Madurez para Procesos de IDR (IBM Software)

Issues and Questions

- How to make improvements
 - ▶ Incrementally - trying to increase by two levels at once can cause failure, so be careful
 - ▶ Identify the greatest pain and work towards relieving it first
- Can you skip levels?
 - ▶ Not really
 - ▶ It is hard to be at the traced level without being organized first (for example)

IBM Rational RequisitePro

- A requirements management tool to:
 - ▶ Organize a requirements project
 - ▶ Provide standard templates and outlines
 - ▶ Capture and document requirements securely
 - ▶ Define requirement types and attributes
 - ▶ Manage and report on traceability
 - ▶ Do reports
 - ▶ Interface to other development lifecycle tools

Modelo de Madurez para Procesos de IDR (IBM Software)

IBM Rational RequisitePro - Key Features by Maturity Level

IBM Rational RequisitePro Feature	Maturity Level
Dynamic integration between Microsoft Word ® and requirements database	1 - Written 2 - Organized
Secure central requirements repository	2- Organized
User security	2 - Organized
Requirements revision history	2 - Organized
Requirements project templates and document outlines	3 - Structured
User-defined requirement types, requirements attributes, and document types	3 - Structured
Requirements attributes and query capability	3 - Structured
Requirements traceability and coverage analysis	4 - Traced
Impact of requirements change	4 - Traced
Integrations with other software development tools - IBM Rational Suites	5 - Integrated