

# Ingeniería de Software II

## Requerimientos

# Software

Principal activo de las empresas ➡ INFORMACIÓN

El software gestiona y produce información

¿Qué se espera del desarrollo de software?



Recursos  
Tiempo  
Dinero  
RRHH



Calidad

# Software

“...Programas de computadoras, procedimientos, posible documentación asociada y datos pertinentes a la operación de un sistema de computadora...”  
(IEEE, 1990)

1. Las instrucciones (programas que proporcionan las características y funciones).
2. Las estructuras de datos que le permiten manipular información.
3. Los documentos que lo describen (operación, uso, etc.)

# Software de calidad



# Software de calidad

- ▶ ¿El software es de calidad?
  - ¿Hace lo que el usuario quiere?
  - ¿Le soluciona el problema que intenta resolver?
  - ¿Lo hace como el cliente quiere?
  - ¿Es factible de construir?, ¿de corregir, de expandir, de mejorar?
  - ¿Se lo puede construir rápido, barato y en forma segura?
  - Fue modificado, ¿Sigue funcionando bien? ¿Le gusta al cliente?

# Software de calidad

- ▶ El software no es de calidad cuando...
  - Cada cambio es difícil de introducir
  - No se entiende el código
  - “Es feo”, lento, difícil de usar, complejo
  - No se sabe cómo probar que funciona bien.
  - Sólo una persona puede arreglarlo
  - Reinventa la rueda
  - El sistema no hace algo que el usuario pidió
  - Hace mal algo
  - Hace algo que el usuario no pidió
  - Es fácil cometer errores con él

# Ingeniería de Software



# Ingeniería de Software

Es una **disciplina de la ingeniería** que se ocupa de todos los aspectos de la producción de software desde las primeras etapas de especificación del sistema hasta su mantenimiento después de que haya entrado en uso.

*Sommerville*





# Ingeniería de Software

Es la aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo, operación y mantenimiento de software.

IEEE



# Ingeniería de Software

La ingeniería de software es una tecnología en capas.  
Es un compromiso organizado con la calidad

*(Pressman, 2010)*



# Ingeniería de Software

- ▶ Busca obtener resultados de la **calidad** requerida dentro de la **planificación** y el **presupuesto** asignado.
  - La sociedad se basa cada vez más en sistemas de software avanzados.

Tenemos que ser capaces de producir sistemas confiables de manera económica y rápida.

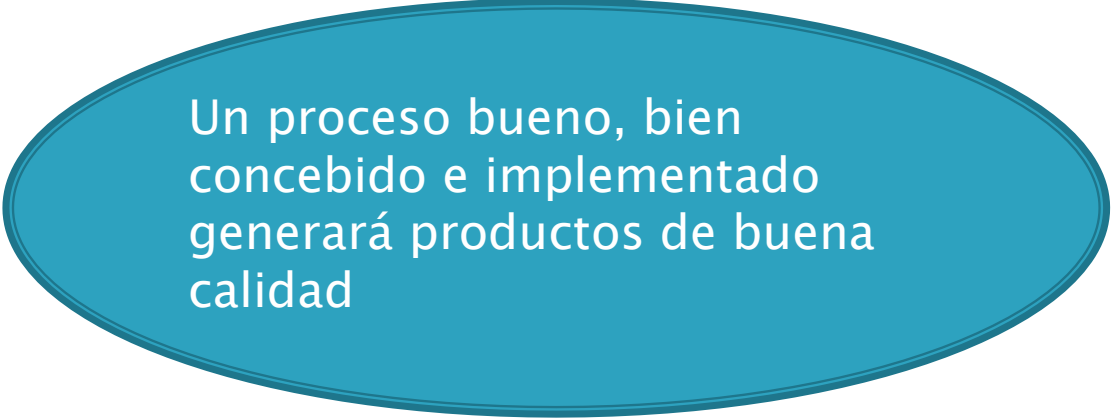
# Calidad de producto

- Una causa de sistematización y automatización que contribuye a la **baja calidad** es el **no acuerdo** entre los participantes de los grupos de desarrollo acerca de las **mejores prácticas** a aplicar en los proyectos.



# Calidad de producto

- ▶ La **calidad** es un **valor en sí mismo** y no un gasto que las empresas deben realizar para que su negocio prospere.
- ▶ Un gran porcentaje de las empresas continúan acortando presupuestos orientados a garantizar la calidad y priorizando objetivos de corto plazo.



Un proceso bueno, bien concebido e implementado generará productos de buena calidad

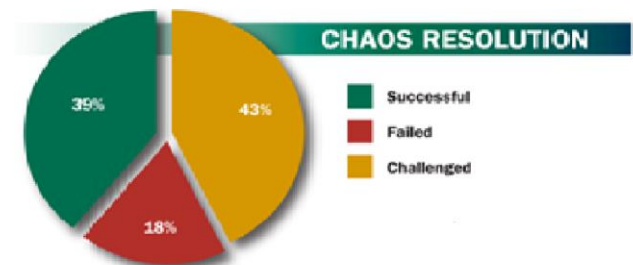
# Chaos report

- ▶ **Exitoso:** El proyecto se completa en el tiempo y con el presupuesto planificado, con todas las funciones y características especificadas originalmente.
- ▶ **Comprometido/Cuestionado (“Challenged”):** El proyecto se completa y es operacional, pero con tiempos y presupuesto mayores a los estimados y/o con menor cantidad de características y funciones de las especificadas inicialmente.
- ▶ **Fallado o Cancelado:** El proyecto se cancela antes de ser completado o nunca es implementado.

Standish Group Chaos Report

<https://www.standishgroup.com>

Project failure rates



# Chaos report

## ► Por qué fallan los proyectos?

Project Challenged Factors	% of Responses
1. Lack of User Input	12.8%
2. Incomplete Requirements & Specifications	12.3%
3. Changing Requirements & Specifications	11.8%
4. Lack of Executive Support	7.5%
5. Technology Incompetence	7.0%
6. Lack of Resources	6.4%
7. Unrealistic Expectations	5.9%
8. Unclear Objectives	5.3%
9. Unrealistic Time Frames	4.3%
10. New Technology	3.7%
Other	23.0%

# Chaos report

- Factores importantes para el éxito de un proyecto (listado ordenado según valoración resultante).

Project Success Factors	% of Responses
1. User Involvement	15.9%
2. Executive Management Support	13.9%
3. Clear Statement of Requirements	13.0%
4. Proper Planning	9.6%
5. Realistic Expectations	8.2%
6. Smaller Project Milestones	7.7%
7. Competent Staff	7.2%
8. Ownership	5.3%
9. Clear Vision & Objectives	2.9%
10. Hard-Working, Focused Staff	2.4%
Other	13.9%



# Chaos report

- ▶ Usuarios poco involucrados
  - En las primeras etapas y durante todo el proyecto, la falta de comunicación puede dificultar la creación de una visión compartida de las metas y los objetivos.
  - La comunicación debe ser una prioridad.
  - Los miembros del equipo deben tener oportunidades para plantear cualquier inquietud.

Se debe trabajar por generar un entorno abierto y de colaboración, donde todo aquel que tenga algo que decir o aportar el proyecto, pueda hacerlo libremente y sin sentirse coaccionado.

# Chaos report

- ▶ Para garantizar la calidad...
  - Lograr el **involucramiento** en el proyecto por parte de clientes y usuarios.
  - Lograr el **compromiso** de la organización (desarrollo) con el proyecto.
  - Lograr un claro **entendimiento** de los requerimientos.
  - Evaluar objetivos
  - Capacitar miembros
  - Motivar a los involucrados
  - Visibilizar objetivos
  - Generar valor agregado en cada entrega



La solicitud del usuario



Lo que entendió el líder del proyecto



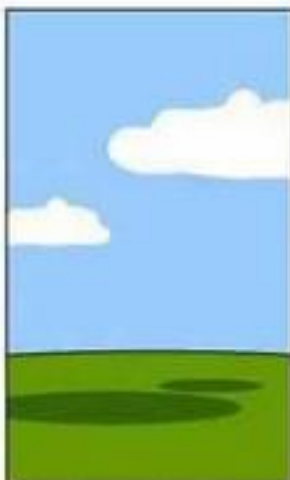
El diseño del analista de sistemas



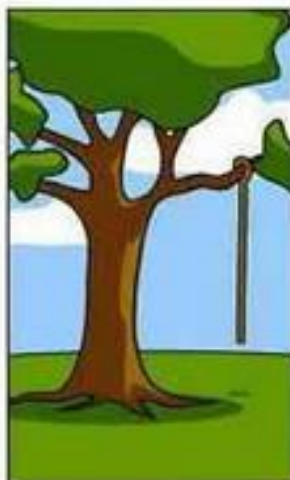
El enfoque del programador



La recomendación del consultor externo



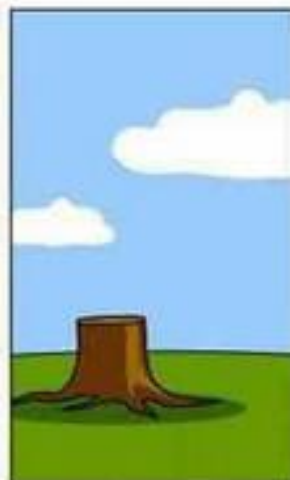
La documentación del proyecto



La implantación en producción



El presupuesto del proyecto



El soporte operativo



Lo que el usuario realmente necesitaba

# Costo relativo de los errores

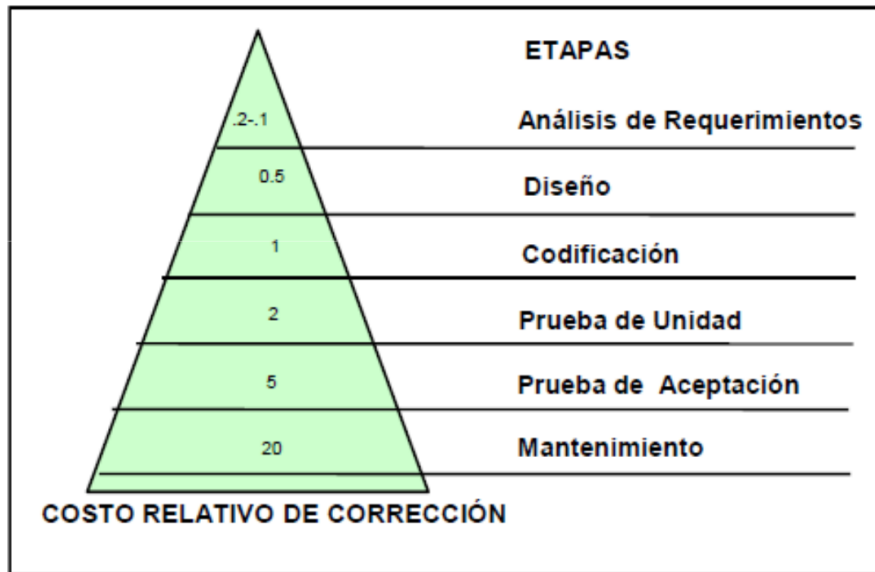


Gráfico 1

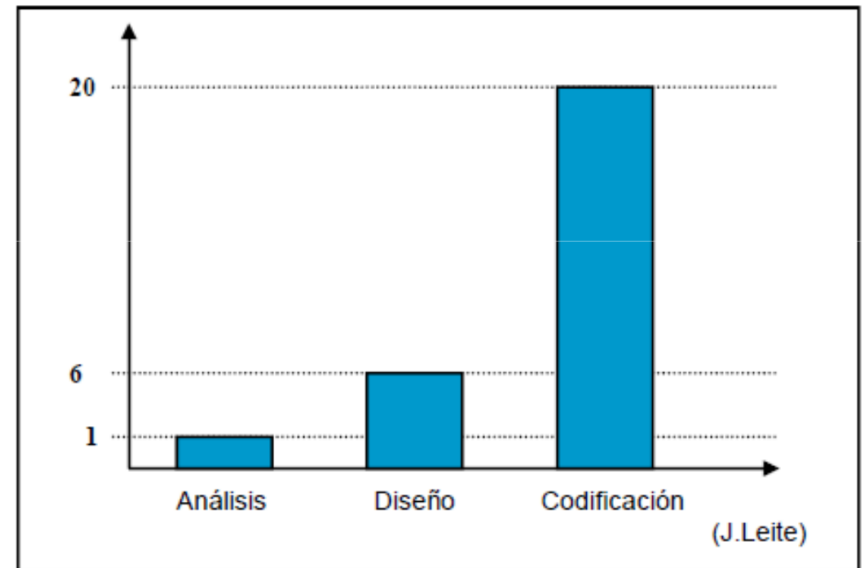
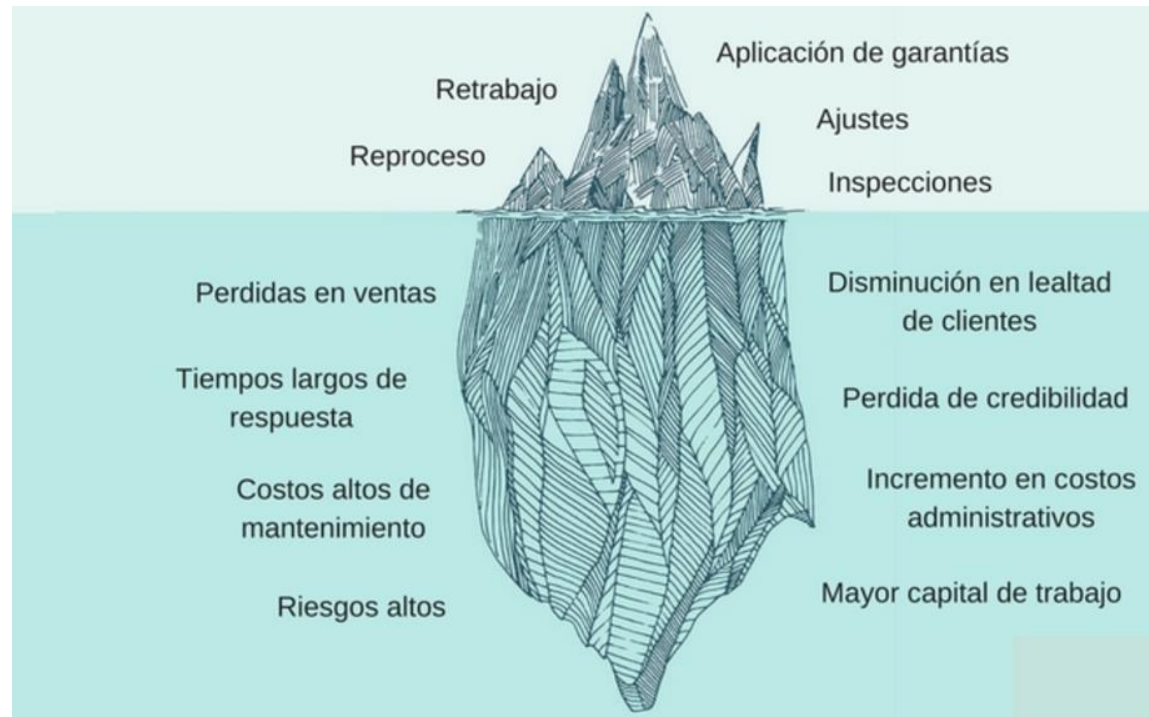


Gráfico 2

# Costos vs. calidad

- Costos visibles e invisibles relativos a la calidad del software



# Requerimiento



# Requerimiento

## ► Necesidad de stakeholders

- Descripciones de lo que el sistema debe hacer, los servicios que presta y las restricciones en su operación.
- Reflejar las necesidades de los clientes para un sistema que sirve a un fin determinado.
- Acuerdo entre los **stakeholders** (en relación al comportamiento esperado y al criterio de aceptación)
- Reducción del esfuerzo: menos retrabajo y omisiones.
- Una base para estimaciones de recursos (esfuerzo, tiempos, dinero).
- Mejoras en el mantenimiento.



# Universo de información (UdI)





# Universo de información (UdI)

## ► UdI

- Es el **contexto general** en el cual el software deberá ser desarrollado y operado.
  - fuentes de información
  - personas relacionadas – actores.
- Realidad influenciada por el conjunto de objetivos definidos.
- También llamado “Universo de Discurso (UdeD)”, “Dominio de Aplicación”, “Macrosistema” o “Negocio”.

# Stakeholders



# Stakeholders

- ▶ Todos aquellos que tienen algún interés en el cambio que se está considerando.
- ▶ Quiénes pierden/ganan con el software.



# Stakeholders

Ejemplo 1:



*Las partes interesadas típicas en una empresa*



# Stakeholders

## Ejemplo 2:

Posibles stakeholders de un sistema automatizado de señalización ferroviaria:

- Los Operadores responsables de manejar el sistema de señalización
- Tripulación del tren
- Gerentes ferroviarios
- Pasajeros
- Ingenieros de instalación y mantenimiento de equipos
- Autoridades de certificación de seguridad
- El equipo de desarrollo del sistema/software
- El dueño de la empresa que desarrolla el software
- Y muchos más...

# Stakeholders

Ejemplo 3:

Interesados en una plataforma de elearning

# Stakeholder

<i>Stakeholder</i>	<i>Descripción</i>
Docente universitario	Responsable del proceso de enseñanza aprendizaje de una asignatura correspondiente a una carrera de las universidades involucradas.
Secretario Académico	Responsable de la Secretaría Académica de la Facultad. Regulador
Director de Carrera	Director de carrera en una Facultad. Beneficiario Regulador
Director de Proyecto de investigación	Director de Proyecto de Investigación. Desarrollador
Integrante de Proyecto de Investigación	Integrante de Proyecto de Investigación Desarrollador
Docente Investigador	Responsable del proceso de enseñanza aprendizaje de una asignatura que, a su vez, participa de un proyecto de investigación. Beneficiario del producto. Usuario
Coordinador general sistema de carrera a distancia	Responsable del área de Educación a Distancia, de la coordinación de las carreras que se dictan con esta modalidad, los profesores involucrados, tutores, diseñadores de contenido, etc. Beneficiario del producto, experto, regulador
Alumno Grado	Usuario, beneficiario del producto.
Alumno Posgrado	Usuario beneficiario del producto.

# Stakeholders

<i>Stakeholder</i>	<i>Descripción</i>
Alumno de grado que se encuentra realizando un PFC	Desarrollador del producto. Alumno de grado que se encuentra realizando un Proyecto Final de Carrera relacionado con el producto de software.
Asesor pedagógico	Asesor pedagógico perteneciente o no a las instituciones de referencia. Experto, Consultor
Consultor de TIC	Consultor de TIC aplicadas a la educación.
Responsable proyecto	Docente Investigador responsable del proyecto Responsable, Decisor
Asesor Técnico	Asesor Técnico sobre conectividad y cuestiones relacionadas con los entornos en donde se instalará el producto de software. Consultor
Diseñador de contenido	Diseñador de contenido de material para educación a distancia. Usuario. Beneficiario funcional
Técnicos	Operador de plataforma de elearning. Usuario, consultor
Profesor tutor de carrera	Tutor de carrera de educación a distancia.
Corrector de estilo	Correctores del material para una adecuada comunicación con el alumno. Consultor
tesistas	Tesistas de maestría, doctorales.
Director de tesis	Directores y codirectores de tesis de posgrado



# Requerimiento

- ▶ **Necesidad de un stakeholder**
  - Cualquier cosa que un cliente necesite (punto de vista del cliente).
  - Cualquier cosa que necesite ser diseñada (punto de vista del diseñador).
- ▶ Se necesita que todos los stakeholders del proyecto lleguen a un entendimiento compartido de los términos usados para describir los requerimientos.

-

# Requerimiento

1. Una condición o capacidad necesaria para un usuario para resolver un problema o alcanzar un objetivo.
2. Una condición o capacidad que debe ser alcanzada por un sistema o componente de un sistema, para satisfacer un contrato, estándar, especificación u otro documento formalmente impuesto.
3. Una representación documentada de una condición o capacidad dada en 1 o 2.

# Clasificación de Requerimientos

- ▶ En cuanto a su contenido:
  - Funcionales
  - No Funcionales
  - Inversos

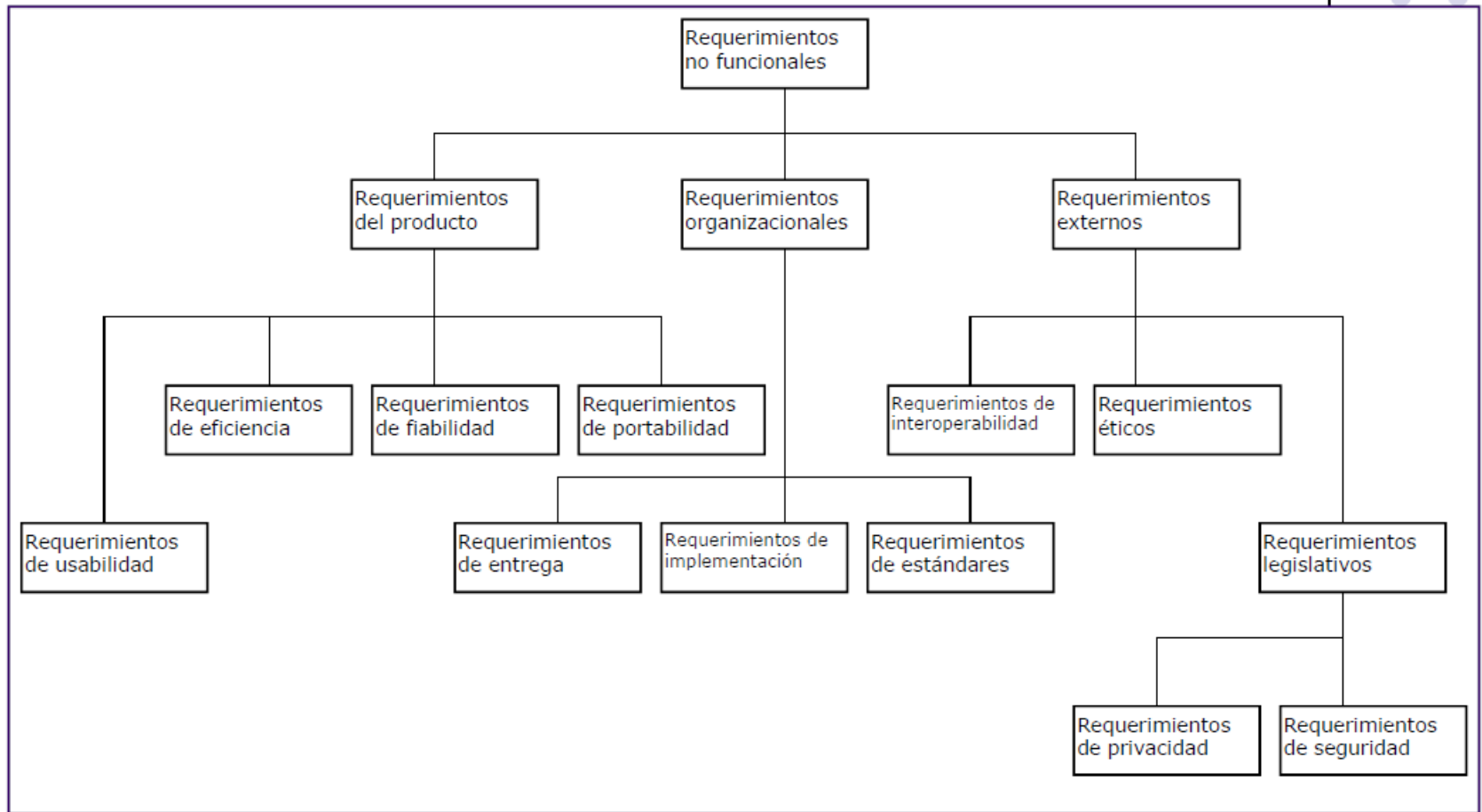
# Clasificación de Requerimientos.

- ▶ **Requerimientos Funcionales (RF):** describen la funcionalidad o los servicios que se espera que el sistema proveerá.
  - Cuando se expresan como *requerimientos del usuario*, se describen de forma general; mientras que los *requerimientos funcionales del sistema* describen con detalle la función de éste, sus entradas y salidas, excepciones, etc.
- ▶ **Requerimientos No Funcionales (RNF):**
  - se refieren a las propiedades emergentes del sistema.
  - definen las restricciones del sistema: capacidad de los dispositivos de entrada/salida, representación de datos que se utiliza en las interfaces, etc.
    - Calidad: confiabilidad, disponibilidad, robustez,...
    - Factores humanos: facilidad de uso, simplicidad de las interfaces,...
    - Características de rendimiento: tiempos de respuesta, performance,...
    - Restricciones de hardware y software: compatibilidad con equipamiento y/o sistemas disponibles,...
    - Cambios y/o adaptaciones a nuevos requerimientos: adaptabilidad, reuso de componentes,...
    - Restricciones de seguridad.
- ▶ **Requerimientos Inversos (RI):** definen cómo el software nunca se debe comportar.

# RF vs. RNF

- ▶ Los requerimientos no funcionales se refieren al sistema como un todo más que a rasgos particulares del mismo; a menudo son más críticos que los requerimientos funcionales particulares.
- ▶ Mientras que el incumplimiento de un requerimiento funcional degradará al sistema, una falla en un requerimiento no funcional del sistema lo inutiliza.

# RNF: Clasificación



# RNF del producto

- ▶ Especifican el comportamiento del producto.
- ▶ Ejemplos:
  - Requerimientos de desempeño en la rapidez de ejecución del sistema y cuánta memoria se requiere;
  - De fiabilidad que fijan la tasa de fallas para que el sistema sea aceptable;
  - De portabilidad y de usabilidad.

# RNF Organizacionales

- ▶ Se derivan de las políticas y procedimientos existentes en la organización del cliente y en la del desarrollador.
- ▶ Ejemplos:
  - Estándares en los procesos que deben utilizarse;
  - Requerimientos de implementación como los lenguajes de programación o el método de diseño a utilizar;
  - Requerimientos de entrega que especifican cuándo se entregará el producto y su documentación.



# RNF Externos

- ▶ Se derivan de los factores externos al sistema y de su proceso de desarrollo.
- ▶ Estos incluyen:
  - **interoperabilidad**
    - modo en que el sistema interactúa con otros sistemas de la organización;
  - **legales**
    - asegurar que el sistema opere dentro de la ley;
  - **Éticos**
    - asegurar que será aceptado por el usuario y por el público en general.
  - **Privacidad**
    - resguardar información crítica.