



Ingeniería de software

Yadran Eterovic S. (yadran@uc.cl)

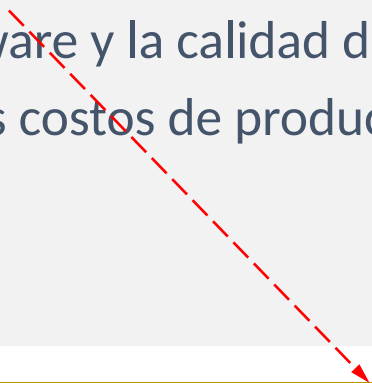
La ingeniería de software es una disciplina dedicada a la investigación, educación y aplicación de métodos y procesos de ingeniería para aumentar significativamente la productividad del software y la calidad del software y a la vez reducir el tiempo para salir al mercado y los costos de producción y de operación del software

La ingeniería de software es una disciplina dedicada a la investigación, educación y aplicación de métodos y procesos de ingeniería para aumentar significativamente la productividad del software y la calidad del software y a la vez reducir el tiempo para salir al mercado y los costos de producción y de operación del software

... productividad del software: El cuociente entre el valor funcional del software producido y el trabajo y los gastos necesarios para su producción

... de ingeniería: Aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de ...; también, aplicación práctica del conocimiento científico a ...

La ingeniería de software es una disciplina dedicada a la investigación, educación y aplicación de métodos y procesos de ingeniería para aumentar significativamente la productividad del software y la calidad del software y a la vez reducir el tiempo para salir al mercado y los costos de producción y de operación del software



... métodos y procesos: Los tres *tracks* de actividades de la IdS que ocurren simultáneamente a lo largo del *ciclo de vida del software*:

- proceso de desarrollo de software
- *software quality assurance*
- gestión del proyecto de software

Proceso de desarrollo de software

Llevar un concepto inicial a un sistema operacional, desde identificar las necesidades del negocio, conducir estudios de factibilidad, y formular los requisitos del sistema, hasta diseñar, implementar, validar y poner en marcha el sistema en el ambiente objetivo

SQA (software quality assurance)

Asegurar que las actividades de desarrollo se lleven a cabo correctamente, y que los artefactos producidos cumplan con los requisitos y estándares de calidad definidos

Gestión del proyecto de software

Estimación de esfuerzo, planificación y programación del proyecto, gestión de riesgos, administración del proyecto, etc., para asegurar que el sistema sea entregado a tiempo y dentro del presupuesto

Un proceso de software es un conjunto estructurado de actividades necesarias para desarrollar un sistema de software

Hay varios tipos distintos de procesos de software

... pero cada uno incluye estos cuatro tipos de actividades fundamentales:

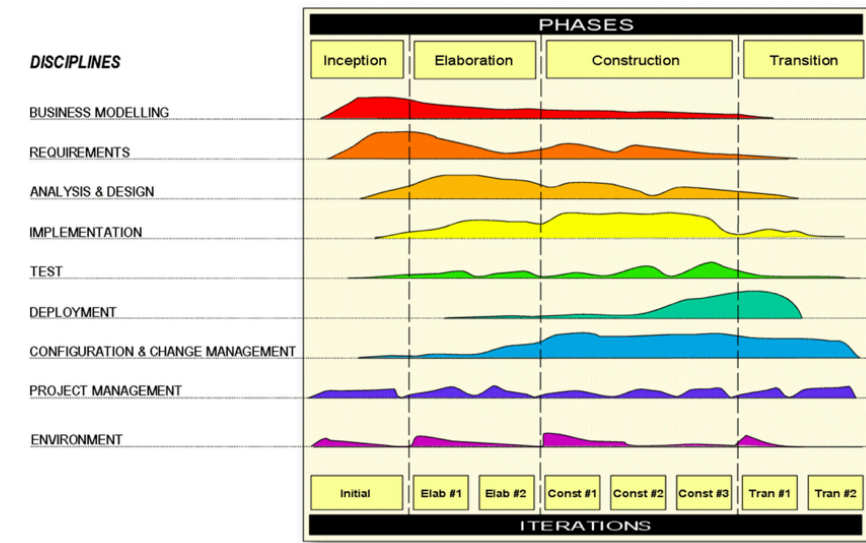
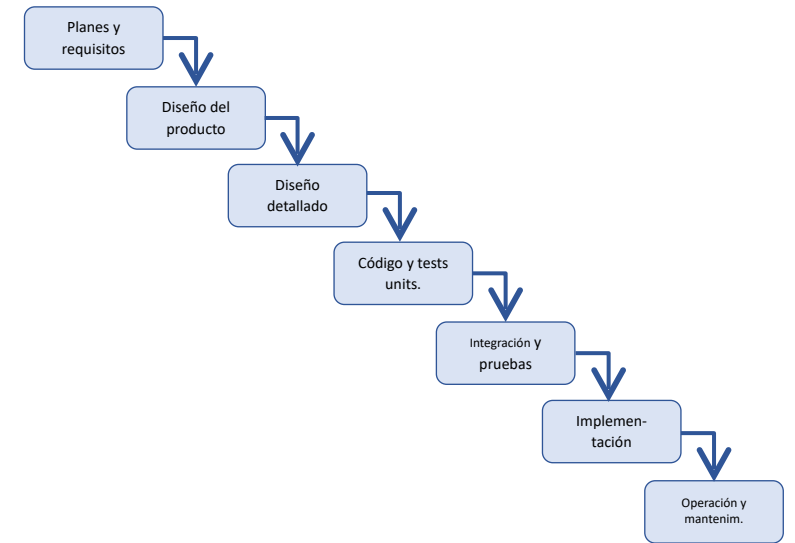
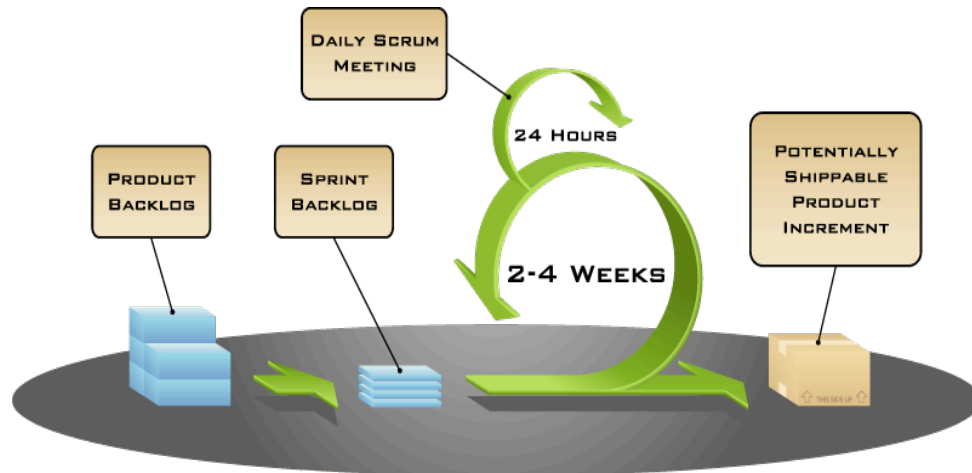
- especificación de software—definir lo que el sistema debería hacer
- diseño e implementación de software—definir la organización del sistema e implementar el sistema
- validación de software—chequear que hace lo que el cliente quiere
- evolución de software—modificar el sistema en respuesta a las necesidades cambiantes del cliente

Un modelo de proceso de software es una representación abstracta de un proceso

- presenta una descripción de un proceso desde alguna perspectiva particular

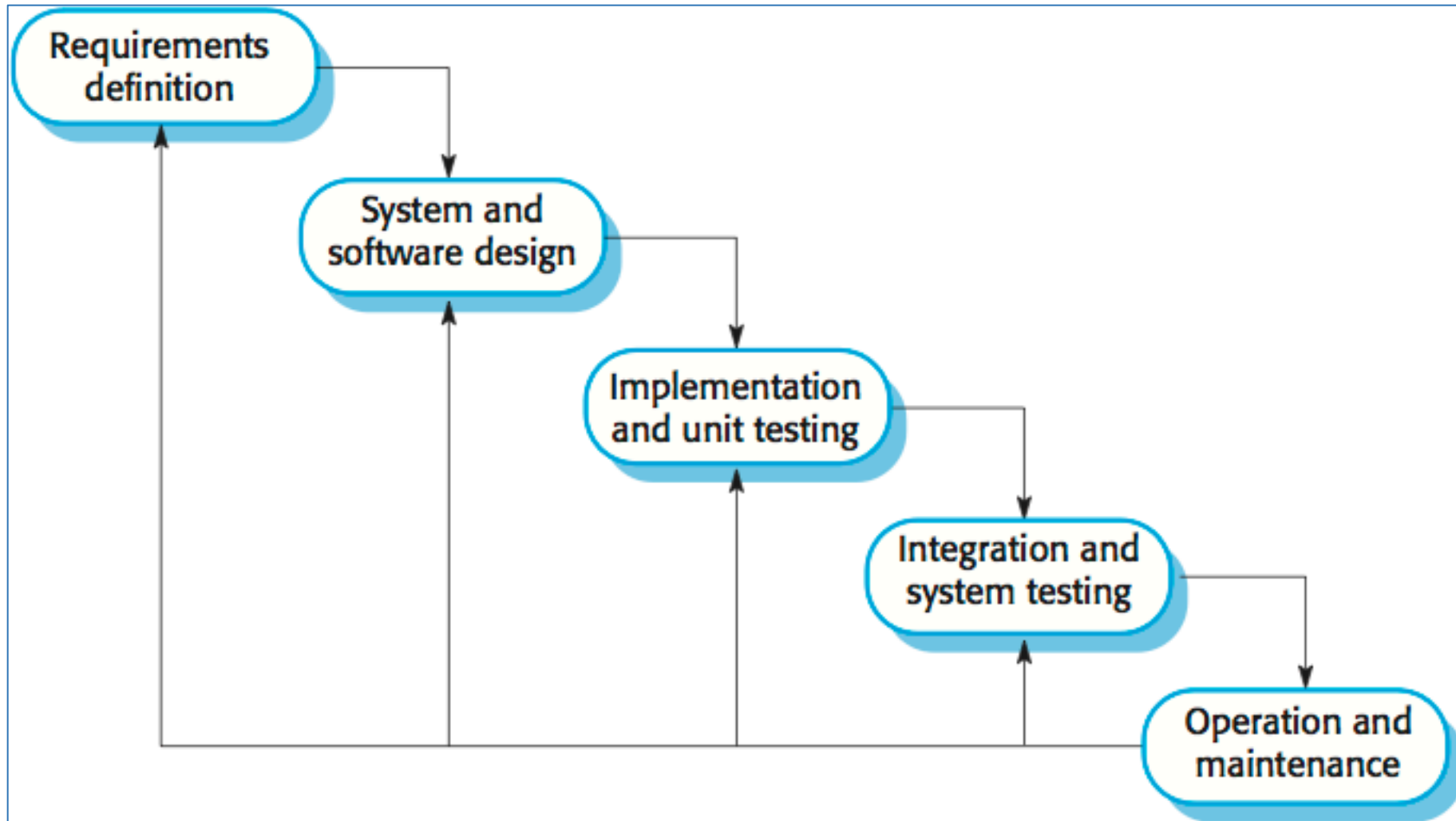
Cuando describimos y discutimos procesos de software, normalmente hablamos de:

- las actividades, p.ej., especificar un modelo datos, diseñar una interfaz de usuario, ...
- la ordenación en el tiempo de estas actividades



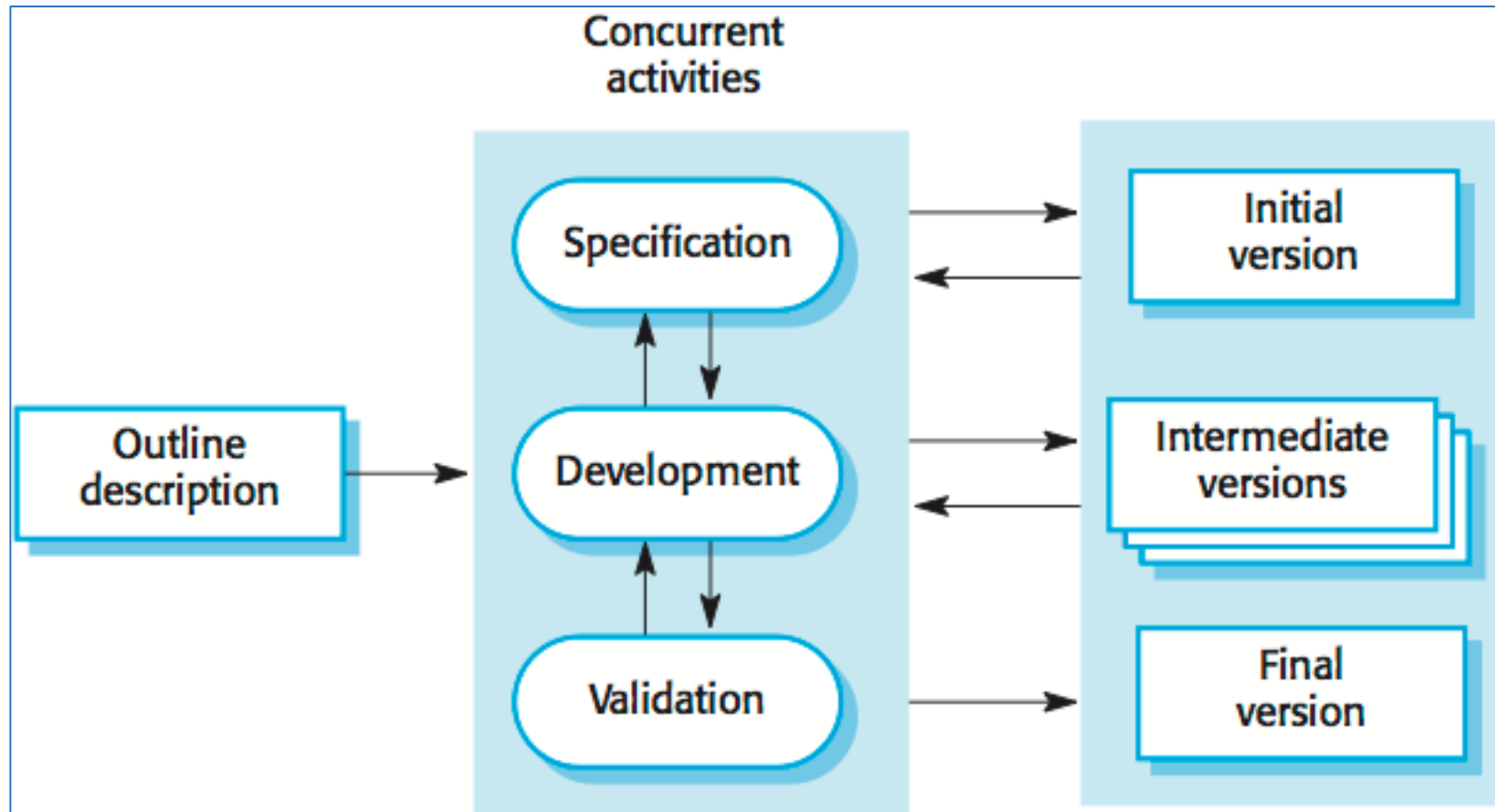
El modelo de cascada:

- modelo dirigido por un plan
- fases separadas y distintas de especificación, diseño, implementación, testing, y mantenimiento



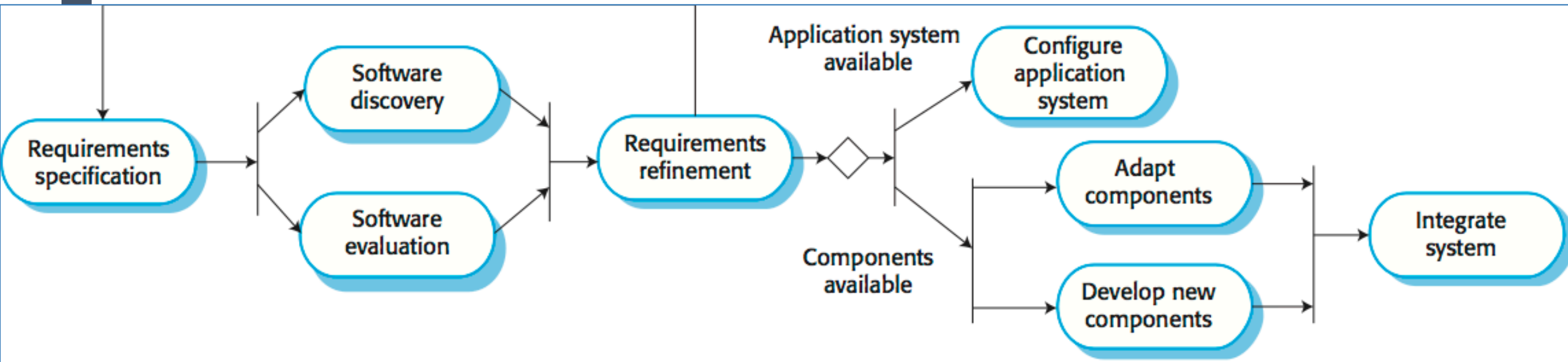
Desarrollo incremental:

- especificación, desarrollo y validación están intercaladas
- el sistema es desarrollado como una serie de versiones (incrementos) en que cada versión agrega funcionalidad a la versión anterior
- puede ser dirigido por un plan o ágil



Integración y configuración:

- basado en la existencia de un número significativo de componentes y/o sistemas reusables
- el proceso de desarrollo se enfoca en integrar estas componentes en un sistema, más que en desarrollarlas desde cero
- puede ser dirigido por un plan o también ágil



Los procesos de software *reales* son secuencias intercaladas de actividades de gestión, técnicas y de colaboración

... con el objetivo de especificar, diseñar, implementar y probar un sistema de software

Las cuatro actividades básicas del proceso—especificación, desarrollo, validación y evolución—son organizadas diferentemente en diferentes procesos de desarrollo:

- en el modelo de cascada, están organizadas en secuencia
- en desarrollo incremental, están intercaladas

Los requisitos son capacidades que el sistema debe proporcionar

Requisitos funcionales:

- afirmaciones sobre las capacidades que el sistema proporciona a sus usuarios y el valor que esto tiene para el negocio

Requisitos no funcionales:

- desempeño
- calidad
- inocuidad (*safety*)
- seguridad
- interfaces

Identificar los requisitos es la parte más difícil del proceso de desarrollo

El equipo de desarrollo no sabe lo suficiente acerca del sistema ni del dominio del sistema

Clientes y usuarios no saben lo que el software puede hacer ni cómo expresar sus necesidades

La falta de antecedentes en común crea una barrera de comunicación entre el equipo y los clientes y usuarios

Los requisitos no se pueden especificar de manera clara

La importancia y dificultad de identificar los requisitos son subvaloradas

Los requisitos no funcionales son subestimados o simplemente no son identificados

Los requisitos cambian a lo largo de todo el ciclo de vida

Los relatos de usuario son declaraciones breves de propósito

... que describen algo que el sistema necesita hacer para algún usuario:

Como un <rol de usuario>, yo puedo <actividad> de modo que <valor para el negocio>

(... en oposición a los requisitos tradicionales:

algo que el sistema debe hacer—para cumplir una necesidad de negocio u obligación contractual, a menudo implícita)

Upload File

As a wiki user I want to upload a file to the wiki so that I can share it with my colleagues.

| |
|--|
| 1. Introduction |
| 1.1 Purpose |
| 1.2 Document conventions |
| 1.3 Intended audience |
| 1.4 Additional information |
| 1.5 Contact information/SRS team members |
| 1.6 References |
| 2. Overall Description |
| 2.1 Product perspective |
| 2.2 Product functions |
| 2.3 User classes and characteristics |
| 2.4 Operating environment |
| 2.5 User environment |
| 2.6 Design/implementation constraints |
| 2.7 Assumptions and dependencies |
| 3. External Interface Requirements |
| 3.1 User interfaces |
| 3.2 Hardware interfaces |
| 3.3 Software interfaces |
| 3.4 Communication protocols and interfaces |
| 4. System Features |
| 4.1 System feature A |
| 4.1.1 Description and priority |
| 4.1.2 Action/result |
| 4.1.3 Functional requirements |
| 4.2 System feature B |

| |
|---|
| 5. Other Nonfunctional Requirements |
| 5.1 Performance requirements |
| 5.2 Safety requirements |
| 5.3 Security requirements |
| 5.4 Software quality attributes |
| 5.5 Project documentation |
| 5.6 User documentation |
| 6. Other Requirements |
| Appendix A: Terminology/Glossary/Definitions list |
| Appendix B: To be determined |

La actividad más estrechamente asociada a la ingeniería es el diseño

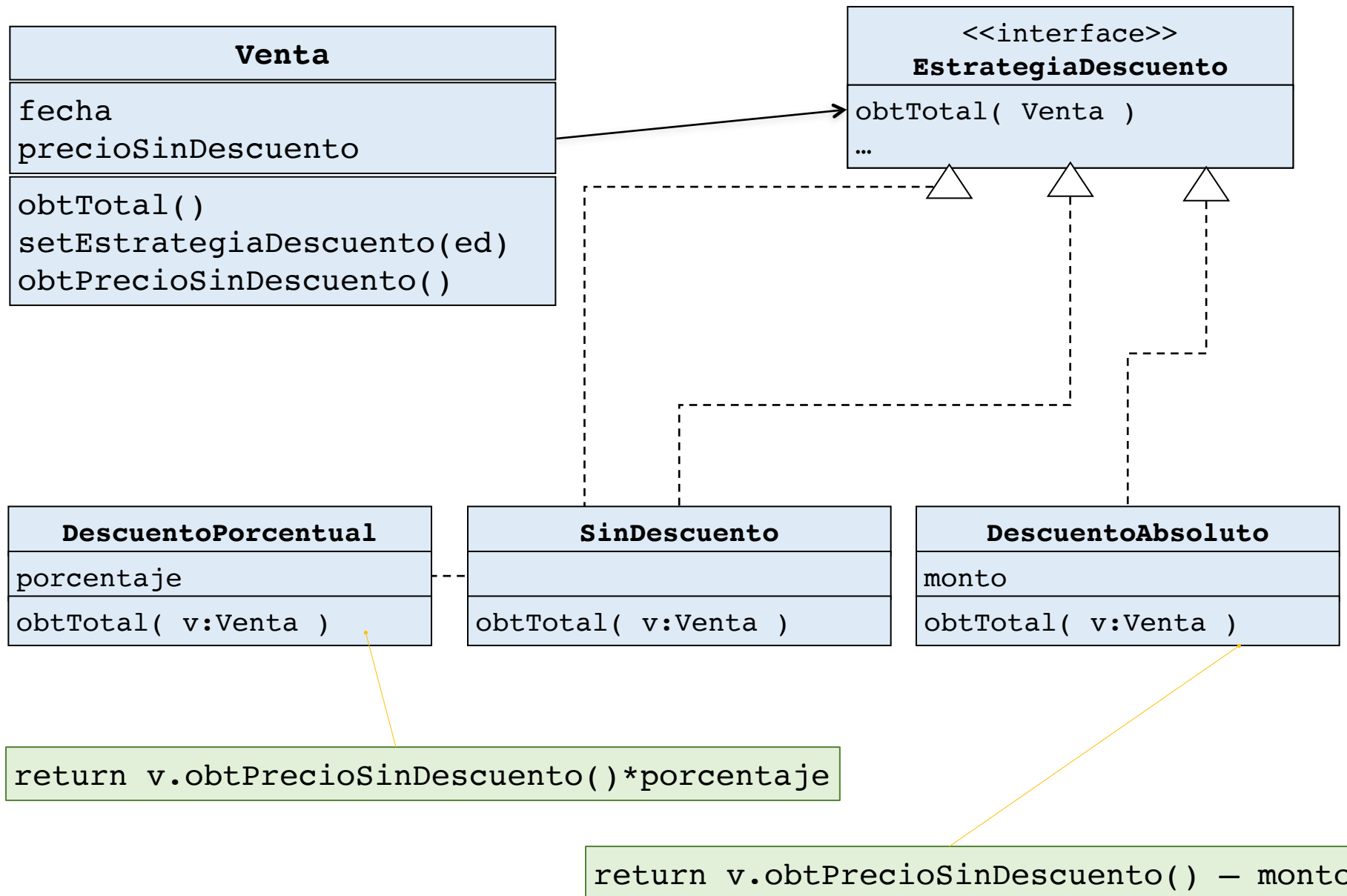
Diseño e implementación de software es el proceso de convertir la especificación del sistema en un sistema ejecutable:

- diseño de software
- implementación—traducir el diseño a un programa ejecutable

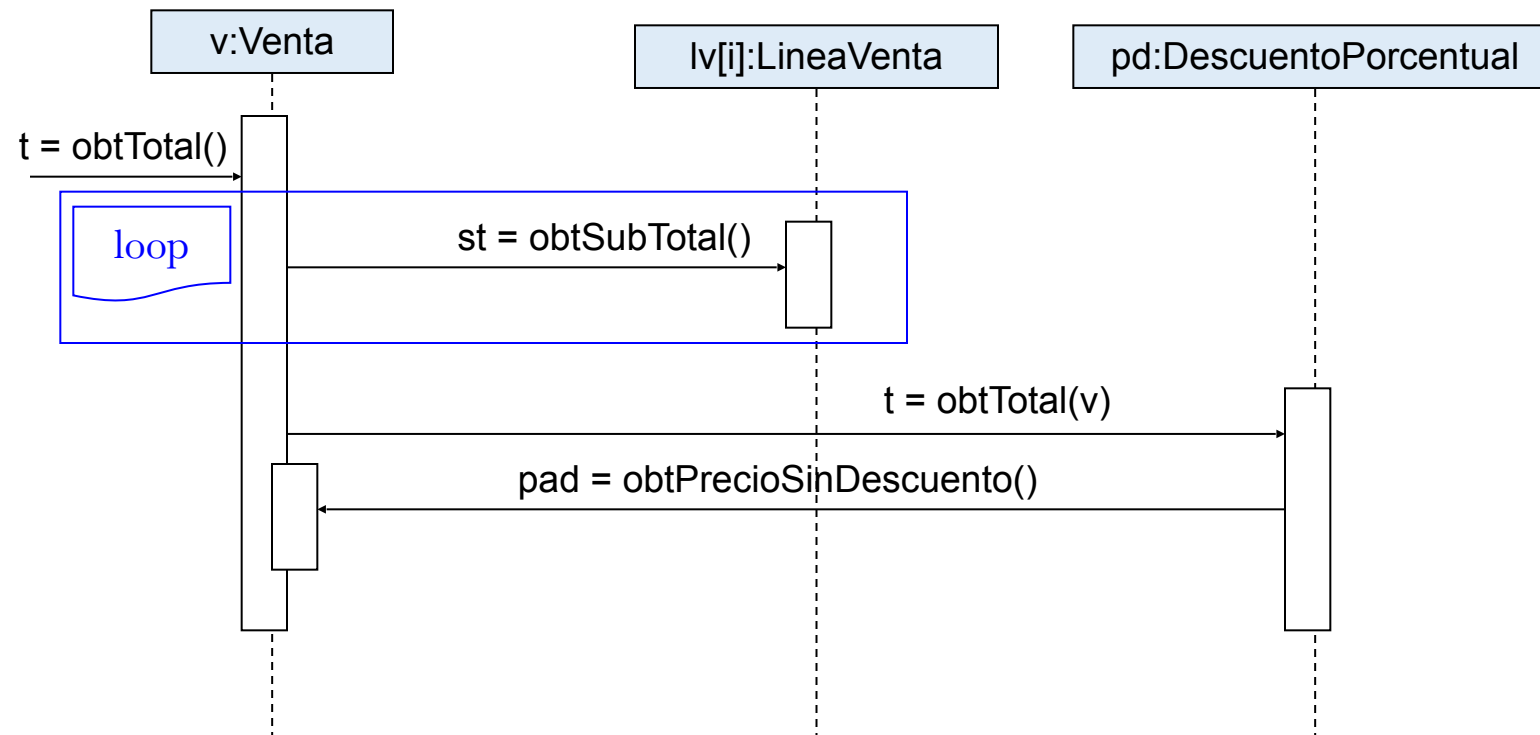
Las actividades de diseño e implementación están estrechamente relacionadas y pueden ser intercaladas

Las actividades de diseño incluyen:

- diseño arquitectónico
- diseño de interfaces con los usuarios y entre componentes del sistema
- diseño de cómo cada una de las componentes va a operar
- diseño de la base de datos



Uso de la estrategia: en la práctica, la venta se comunica directamente con una estrategia de descuento particular



La gestión del proyecto se enfoca en estimaciones, planificación, riesgos, etc.

Los proyectos de software requieren de un número grande de ingenieros de software trabajando en equipos:

- los ingenieros deben comunicarse y colaborar para llevar a cabo conjuntamente las actividades de desarrollo
- ¿cómo se forman los equipos? ¿qué roles y responsabilidades son necesarios?

La comunicación eficaz es crítica y normalmente significa un overhead importante:

- el contenido es intelectual, complejo, abstracto y fácil de malinterpretar

Todos los que van a hacer la estimación—principalmente, los desarrolladores—se sientan alrededor de una mesa

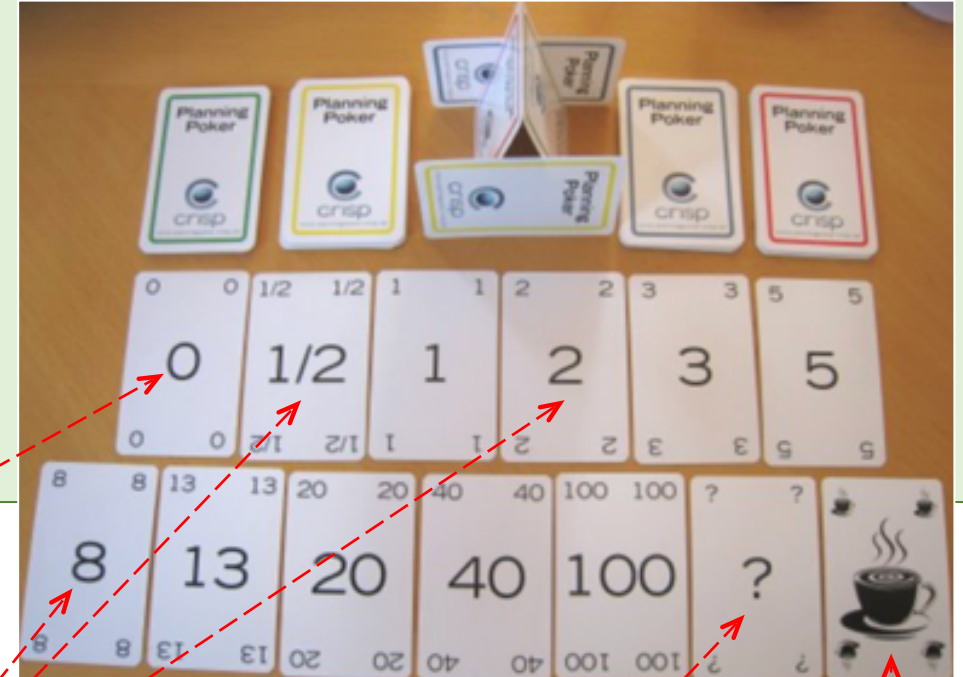
Al centro de la mesa se coloca un relato de usuario

Cada persona recibe una baraja de 13 naipes:

... hace individualmente su estimación para el relato de usuario

... y coloca el naipe correspondiente cara abajo sobre la mesa

Finalmente, todos dan vuelta sus naipes al mismo tiempo



Ya está hecho

Cada naipe representa una
estimación en días-desarrollador

*“No tengo suficiente información
para hacer la estimación”*

*“Necesito un
descanso”*

¿Cuánto esfuerzo y cuánto tiempo demanda cada fase del desarrollo de una aplicación?

| Fase | Esfuerzo % | Tiempo % |
|--------------------------|------------|----------------------|
| Planes y requisitos | 7 (2–15) | 16–24 (2–30) ← extra |
| Diseño del producto | 17 | 24–28 |
| Programación | 64–52 | 56–40 |
| Diseño detallado | 27–23 | |
| Código y tests unitarios | 37–29 | |
| Integración y pruebas | 19–31 | 20–32 |

El desarrollo de software es un problema “malvado”

Cada problema es único

El problema no tiene una formulación claramente definida

La solución sólo se puede evaluar subjetivamente, en el largo plazo, en términos de buena o mala, y nunca es suficientemente buena (siempre se puede mejorar)

El proceso de resolución es político y cada paso tiene un gran número de posibilidades

El que resuelve el problema no tiene derecho a equivocarse, porque las consecuencias son desastrosas

El software es un producto intangible

No lo podemos ver, tocar, ni medir directamente ni exactamente

→ es difícil estimar el esfuerzo requerido para desarrollar un sistema

... y es difícil medir el progreso de un proyecto de software

Hay quienes afirman que gestionar proyectos de desarrollo de software es diferente de gestionar proyectos en otras disciplinas de ingeniería

El título de un texto sobre estimaciones en proyectos de software lo dice todo:

- “Software Estimation: Demystifying the Black Art”

El desarrollo de software es una actividad riesgosa

No podemos anticipar las dificultades que vamos a encontrar, menos aún cómo las enfrentaremos

Faltan las “leyes fundamentales del software”

Sólo queda agregar holguras a los plazos

Hay que llevar a cabo gestión de riesgos, como parte de la gestión del proyecto