

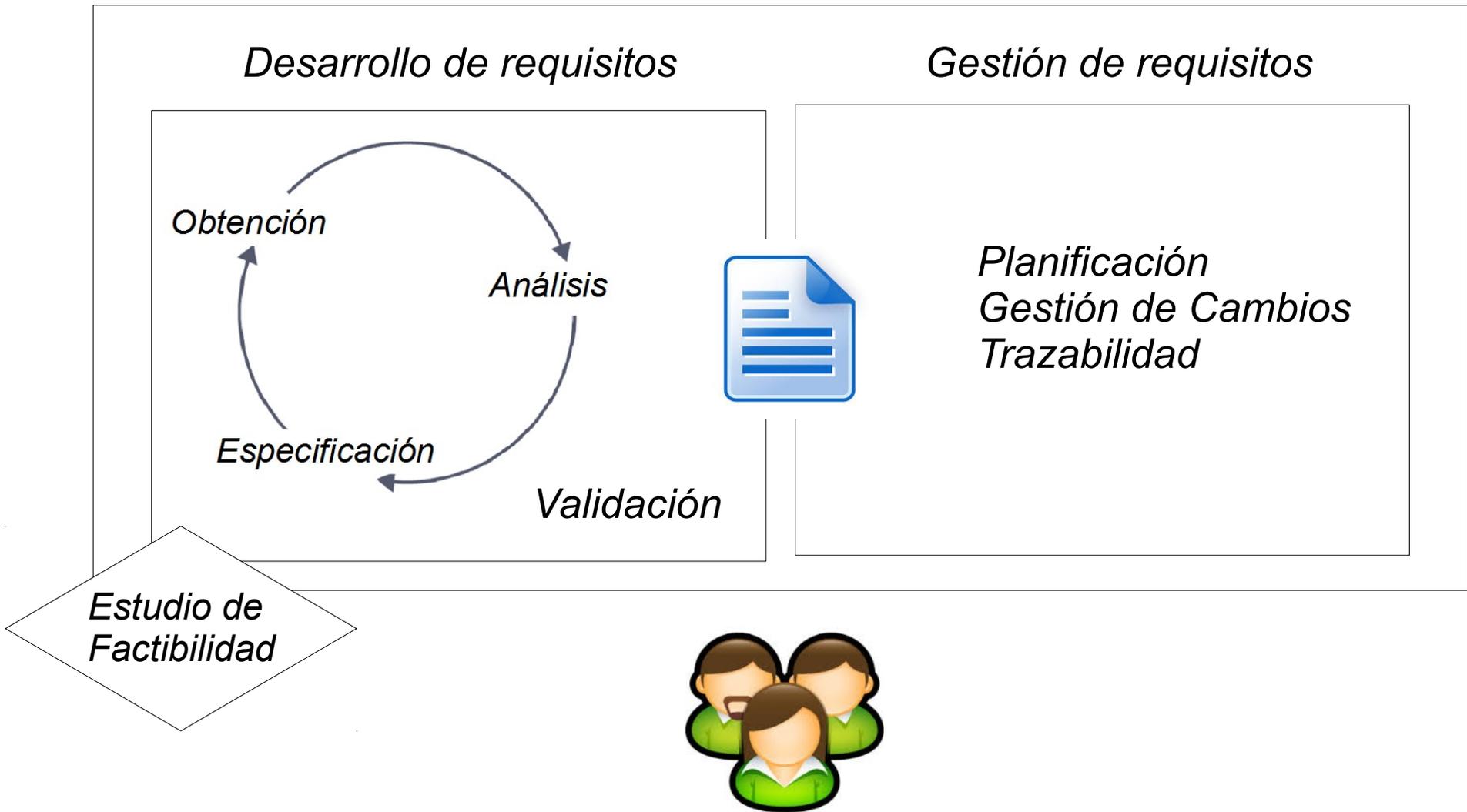
Ingeniería de Software

Ingeniería de Requisitos Clase 2

Sommerville 10 — Capítulo 4 (sección 4.5).

Wieggers — Capítulo 7 y 8

Actividades de la ingeniería de requisitos



Técnicas para obtención de requisitos

- Entrevistas
- Investigar antecedentes
- Workshops
- Focus groups
- Observaciones - Etnografías
- Cuestionarios
- Análisis de las interfaces del sistema
- Análisis de la interfaz de usuario
- Análisis de documentación
- Tormenta de ideas
- Escenarios
- Casos de uso
- Prototipado
- Modelado de procesos
- Historias de usuario

Observaciones - Etnografías

- Implican observar a los usuarios mientras realizan sus actividades. Pueden ser silenciosas o interactivas.
 - En general los usuarios no son muy precisos al describir sus tareas. Esto puede ser porque son tareas complejas y difíciles de explicar. En otros casos, están tan familiarizados que omiten detalles de forma inconsciente.
-
- | | |
|---|--|
| <ul style="list-style-type: none">• Ventajas• Confiable• Muy rico• Desarrolla empatía | <ul style="list-style-type: none">• Desventajas• Muy costosas• Efecto Hawthorne• Hay que tener cuidado en generalizar (sesgo particular/local) |
|---|--|

Questionarios / Encuestas

- Son una manera de estudiar a grandes grupos de usuarios para entender sus necesidades.
- Antes de usar el enfoque:
 - Determinar la información que se precisa.
 - Determinar el enfoque más adecuado:
 - Abierto, cerrado, combinado.
 - Múltiple opción, valor en escala, orden relativo.
 - Desarrollar cuestionario.
 - Probarlo con perfil típico.
 - Analizar resultado de las pruebas.
- Su principal uso es para validar y obtener datos estadísticos sobre preferencias.

Questionarios / Encuestas

- Preparar preguntas bien escritas es el mayor desafío.
- Recomendaciones
 - Proveer opciones para todas las posibles respuestas.
 - Hacer que las opciones sean mutu excluyentes.
 - Utilizar preguntas cerradas para análisis estadístico y abiertas para recolectar ideas o necesidades nuevas.
 - Siempre probar el cuestionario antes de usarlo.
 - No incluir demasiadas preguntas.
- **Ventajas**
 - Economía de escala.
 - Conveniente para quien contesta.
 - Respuestas anónimas.
- **Desventajas**
 - Menos rico.
 - Problemas por no-respuesta.
 - Esfuerzo de desarrollo.

Análisis de las interfaces del sistema

- Implica examinar los otros sistemas con los que se conecta el sistema.
- Revela requisitos funcionales relativas al intercambio de datos y servicios entre sistemas.
- Para cada sistema que se deba comunicar con el nuestro se identifican las funcionalidades que nos puedan generar requisitos. Esos requisitos pueden describir los datos a pasar a otros sistemas, los datos a recibir de otros sistemas y las reglas sobre los datos (por ejemplo criterios de validación).
- Es útil para revisar las validaciones de la información a comunica o recibir (quizás, por ejemplo, se puedan omitir validaciones)

Análisis de la interfaz de usuario

- Implica estudiar sistemas existentes para determinar requisitos de usuario y funcionales.
- Lo mejor es utilizar sistemas existentes, pero sino se pueden utilizar screenshots (por ejemplo de manuales de usuario).
- Puede ayudar a identificar una lista completa de pantallas y a descubrir características potenciales del nuevo sistema. Se utiliza para identificar pasos comunes de los usuarios al realizar tareas así como para crear borradores de casos de uso.
- No hay que asumir que una funcionalidad es necesaria porque se encuentra en un sistema existente. O mantener la interfaz de usuario parecida o que se comporte de forma similar a la estudiada.

Análisis de documentación

- Contempla examinar toda la documentación existente en busca de requisitos potenciales del software.
- La documentación más útil incluye especificación de requisitos, procesos de negocio, lecciones aprendidas y manuales de usuario de aplicaciones existentes o similares.
- Es a una forma rápida de introducirse rápidamente en un nuevo dominio o en un sistema existente.

Tormenta de ideas

- Ayuda a la participación de todos los involucrados.
- Reglas: no se permite criticar ni debatir, generar tantas ideas como sea posible, mutar y combinar ideas.
- **Fase de generación**
 - Los principales stakeholders se reúnen y se establecen objetivos.
 - Se pide que cada uno escriba sus ideas.
- **Fase de reducción**
 - Se leen las ideas y se establece si es válida.
 - Se agrupan ideas y se hacen las definiciones necesarias.
 - Se priorizan (opcional).

Historias y escenarios

- Las historias y los escenarios describen cómo se puede utilizar el sistema para una tarea particular.
- Describen qué hacen las personas, qué información usan y producen y qué sistemas utilizan en ese proceso.
- Las historias se escriben como un texto narrativo que describen a alto nivel el uso del sistema, los escenarios en general tienen información estructurada como por ejemplo:
 - Una descripción de la situación de partida.
 - Una descripción del flujo normal de los eventos
 - Una descripción de lo que puede salir mal.
 - Información sobre otras actividades concurrentes.
 - Una descripción del estado cuando finaliza el escenario.

Escenarios - ejemplo

INITIAL ASSUMPTION:

The patient has seen a medical receptionist who has created a record in the system and collected the patient's personal information (name, address, age, etc.). A nurse is logged on to the system and is collecting medical history.

NORMAL:

The nurse searches for the patient by family name. If there is more than one patient with the same surname, the given name (first name in English) and date of birth are used to identify the patient.

The nurse chooses the menu option to add medical history.

The nurse then follows a series of prompts from the system to enter information about consultations elsewhere on mental health problems (free text input), existing medical conditions (nurse selects conditions from menu), medication currently taken (selected from menu), allergies (free text), and home life (form).

WHAT CAN GO WRONG:

The patient's record does not exist or cannot be found. The nurse should create a new record and record personal information.

Patient conditions or medication are not entered in the menu. The nurse should choose the 'other' option and enter free text describing the condition/medication.

Patient cannot/will not provide information on medical history. The nurse should enter free text recording the patient's inability/unwillingness to provide information. The system should print the standard exclusion form stating that the lack of information may mean that treatment will be limited or delayed. This should be signed and handed to the patient.

OTHER ACTIVITIES:

Record may be consulted but not edited by other staff while information is being entered.

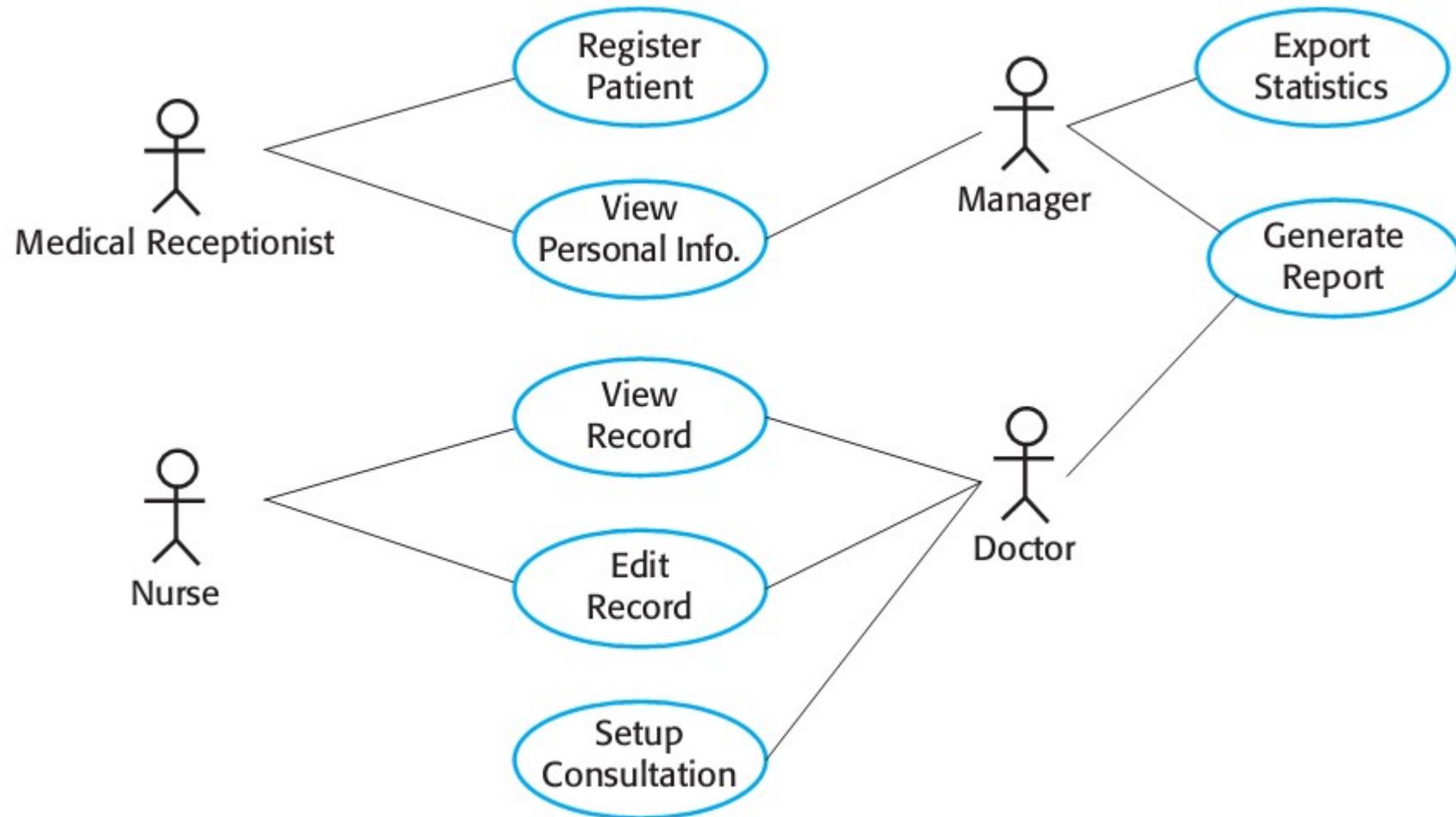
SYSTEM STATE ON COMPLETION:

User is logged on. The patient record including medical history is entered in the database, a record is added to the system log showing the start and end time of the session and the nurse involved.

Casos de uso

- Los casos de uso son una técnica incluida en UML que se basa en el concepto de los escenarios.
- Un conjunto de casos de uso debe describir todas las posibles interacciones con el sistema. Se identifica cada tipo de interacción entre los distintos actores y el sistema y se les da un nombre.
- Constan de un modelo gráfico de alto nivel complementado con una descripción más detallada de cada caso de uso.

Casos de uso



Casos de uso

- Formato simple y estructurado donde los usuarios y desarrolladores pueden trabajar juntos.
- No son de gran ayuda para identificar aspectos no funcionales.
- Mientras se definen los casos de uso, puede ser un buen momento para definir pantallas u otros objetos con los que el usuario interactúa.
- Pueden ser usados en el diseño y en el testing del sistema.
- **Ventajas**
 - Permiten ver fácilmente la interacción del sistema con los demás actores (usuarios u otros sistemas)
 - Muy útiles cuando la implementación se va a hacer OO y con UML.
- **Desventajas**
 - No son demasiado útiles para describir sistemas sin usuarios y/o con pocas interfaces.
 - No modelan bien requisitos no funcionales y restricciones de diseño.

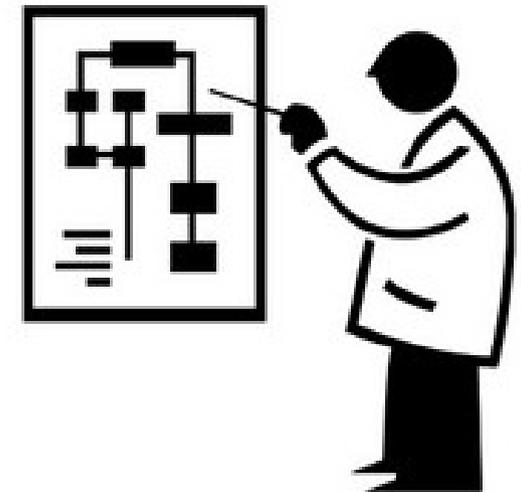
Historias de usuario

- Refieren a descripciones cortas y de alto nivel de las funcionalidades expresadas en los términos del cliente.
- Una historia de usuario típica tiene la forma:
 - *“As a <role>, I want <goal/desire> so that <benefit>.”*
- Pretenden contener justo la información necesaria para que los desarrolladores puedan producir una estimación razonable del esfuerzo para su implementación.
- La idea es evitar perder demasiado tiempo relevando detalles de requisitos que luego cambian demasiado o son desestimados.



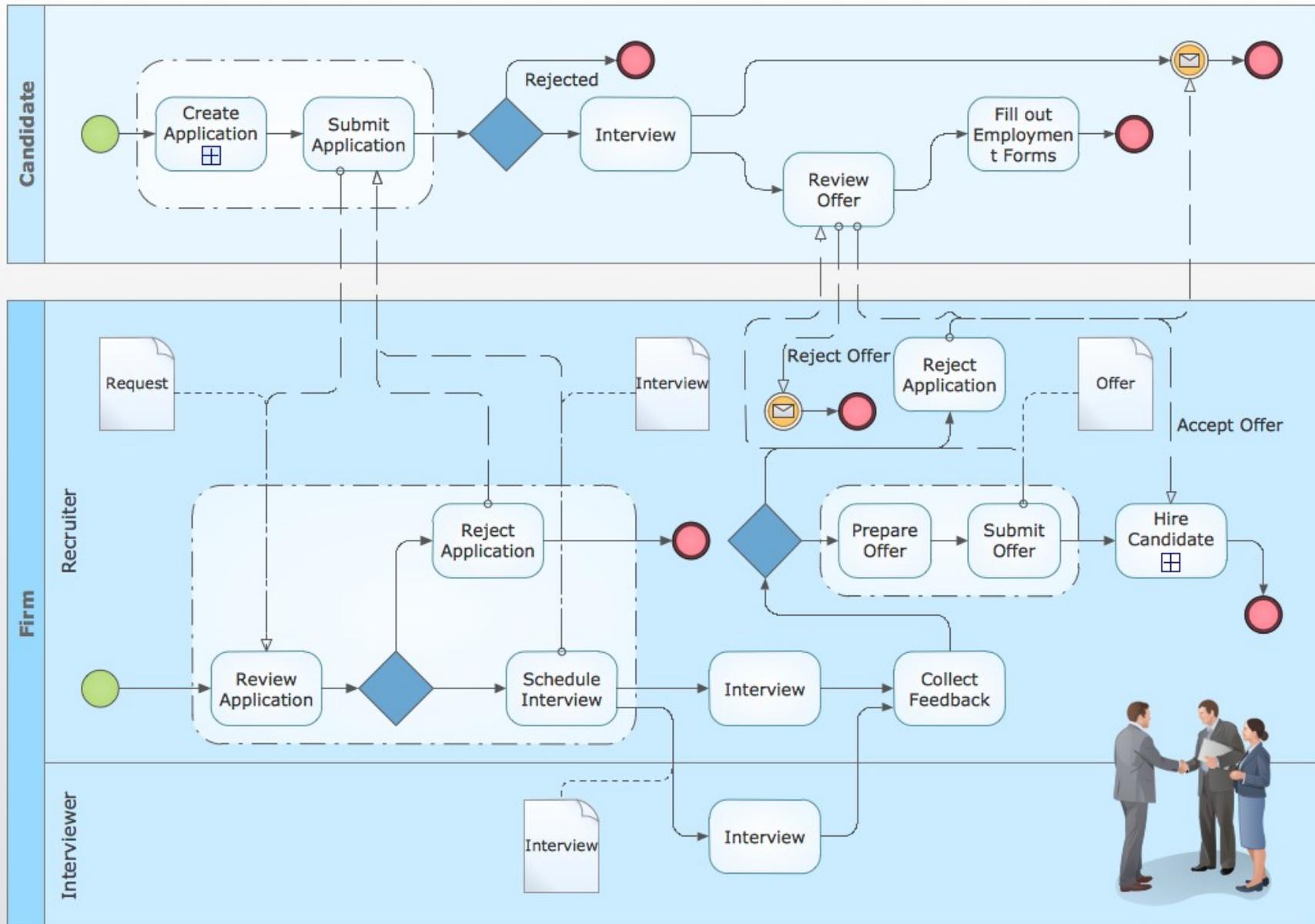
Modelado de Procesos

- Permiten entender el trabajo con múltiples pasos, roles o departamentos.
 - Iniciados por un evento.
 - Incluyen actividades manuales, automáticas o combinaciones de ambos tipos.
- Pueden volverse complejos y pesados si no se estructuran con cuidado.
- Los procesos complejos se pueden descomponer para ayudar su entendimiento.



Modelado de Procesos - Ejemplo

Hiring Process Example



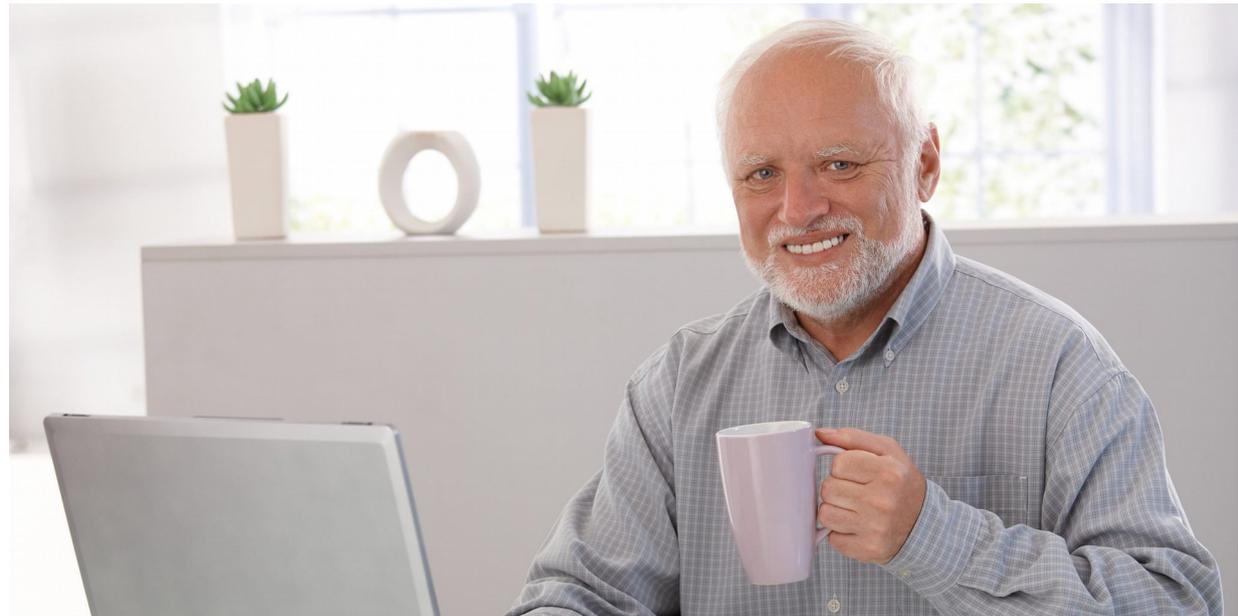
Selección de técnicas a utilizar

- En general es necesario utilizar más de una técnica y la elección está muy relacionada al tipo de proyecto.
- Wiegers hace algunas sugerencias de ejemplo (incluye sólo las técnicas presentadas en el libro):

	Interviews	Workshops	Focus groups	Observations	Questionnaires	System interface analysis	User interface analysis	Document analysis
Mass-market software	x		x		x			
Internal corporate software	x	x	x	x		x		x
Replacing existing system	x	x		x		x	x	x
Enhancing existing system	x	x				x	x	x
New application	x	x				x		
Packaged software implementation	x	x		x		x		x
Embedded systems	x	x				x		x
Geographically distributed stakeholders	x	x			x			

A considerar

- Organizar los requisitos según
 - Viewpoints – puntos de vista (stakeholders)
 - Módulos o subsistemas (arquitectura)
- No esperar que los clientes presenten una lista completa, corta y completa de sus necesidades.



- **Un caso de uso describe una secuencia de interacciones entre un sistema y un actor externo que resultan en un resultado de valor para el actor.**
- Cada escenario comprende una instancia de uso del sistema. Un caso de uso contempla un conjunto de escenarios relacionados.
- Son independientes del método de diseño y del lenguaje que se utilice para la implementación.

Actores

- Un actor es una persona (o quizás otro sistema) que interactúa con el sistema para realizar un caso de uso.

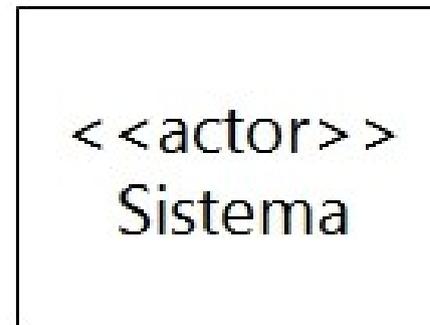
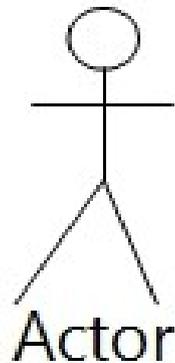
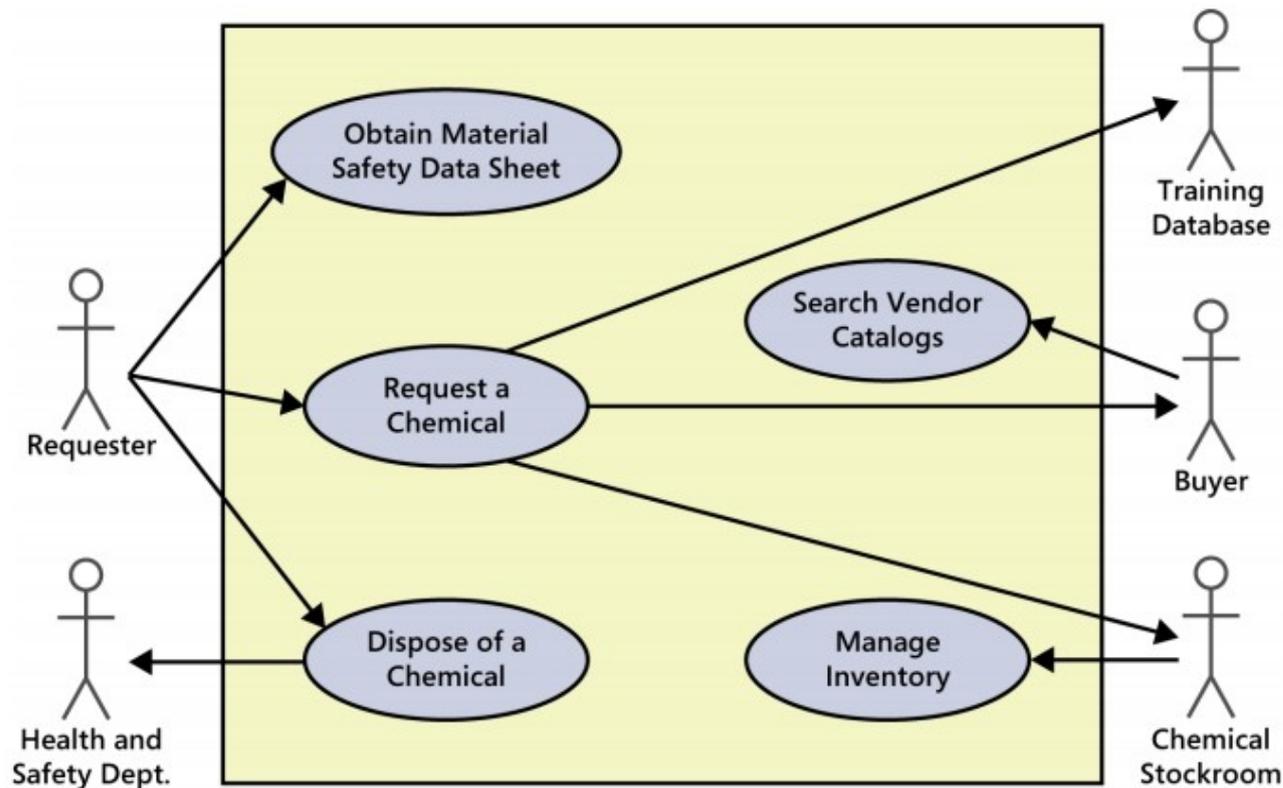


Diagrama de Casos de Uso

- Provee una representación de alto nivel de los requisitos de usuario.



Elementos esenciales de un Caso de Uso

- Identificador único.
- Breve descripción.
- Precondiciones.
- Poscondiciones.
- *Una lista enumerada de pasos que muestran la secuencia de interacciones entre el actor y el sistema.*

Descripción

- Es una descripción breve, a alto nivel, del caso de uso y del flujo de acciones que comprende.
- Ejemplo: *Realizar Transferencia* - Descripción
 - Este caso de uso comienza cuando un usuario se loguea al sistema y le indica que desea realizar una transferencia.
El usuario indica las cuentas de origen y destino de la transferencia y el monto de la misma. El sistema se comunica con el banco, el cual realiza la transferencia especificada.

Precondiciones

- Describen condiciones que debe cumplir el sistema para que se pueda iniciar el caso de uso.
- Ejemplo: *Impresión de Factura* - Precondiciones
 - El proceso de liquidación de la factura ha sido iniciado. La factura se encuentra pendiente de impresión.

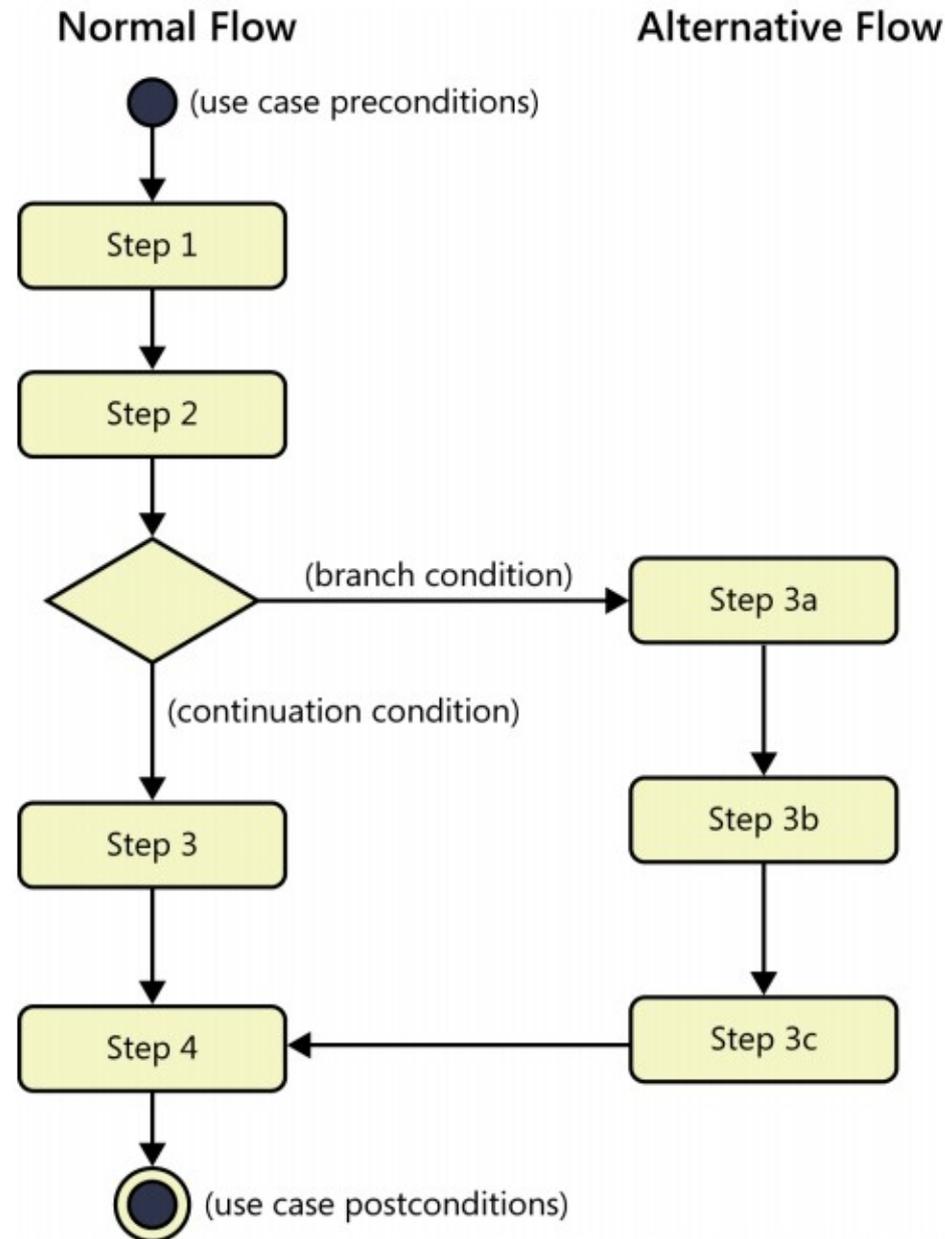
Poscondiciones

- Las poscondiciones describen el estado del sistema luego de la ejecución exitosa del caso de uso.
- Ejemplo: *Balance Contable Anual*- Poscondiciones
 - El ejercicio contable queda cerrado, esto implica que a partir de ese momento no es posible registrar nuevos asientos contables para ese ejercicio.

Flujo normal y flujos alternativos

- El flujo principal muestra la secuencia de pasos más común que se lleva a cabo para la ejecución del caso de uso.
- Además del flujo principal, un caso de uso puede tener varios flujos alternativos. Un flujo alternativo describe un escenario alternativo al principal (casos particulares y manejo de excepciones/errores).

Flujo normal y flujos alternativos



Ejemplo de flujo normal

1. El usuario ingresa al sitio.
2. El sistema solicita que se registre.
3. El usuario ingresa sus datos.
4. El sistema envía sus datos al banco.
5. El banco valida los datos.
6. El sistema le permite al usuario ingresar y le ofrece un menú de servicios.
7. Usuario: indica al sistema que desea realizar una transferencia
8. Sistema: lista las cuentas del usuario
9. Sistema: pide al usuario que seleccione la cuenta de origen de la transferencia de la lista o que la ingrese manualmente.
10. Usuario: selecciona cuenta de origen de la transferencia de la lista.
11. Sistema: pide al usuario que seleccione la cuenta de destino de la transferencia de la lista o que la ingrese manualmente.
12. Usuario: selecciona cuenta de destino de la transferencia de la lista.
13. Sistema: pide al usuario que ingrese el monto de la transferencia.
14. Usuario: ingresa el monto de la transferencia.
15. Sistema: se comunica con el servidor del banco y le indica que desea realizar una transferencia entre las cuentas especificadas y con el monto indicado.
16. Banco: realiza la transferencia especificada e indica que la transferencia se realizó con éxito.
17. Sistema: actualiza el monto de las cuentas involucradas.
18. Sistema: notifica al usuario que la transferencia se efectuó correctamente.

Ejemplos de flujos alternativos

10.A. El usuario desea ingresar manualmente el número de cuenta origen.

10.A.1. Usuario: selecciona ingresar manualmente el número de cuenta.

10.A.2. Sistema: pide al usuario que ingrese número de cuenta y código del banco.

10.A.3. Usuario: ingresa número de cuenta y código del banco.

10.A.4. Vuelve al punto 5.

16.A. Existe un error en la comunicación con el banco.

16.A.1. Sistema: no se puede comunicar con el banco

16.A.2. Sistema: informa al usuario que no se puede comunicar con el banco en este momento.

16.A.3. Fin CU

Flujos alternativos de flujos alternativos

12.B.4.A. Condición que lleva a este flujo alternativo

12.B.4.A.1. Paso 1

12.B.4.A.2. Paso 2

...

o bien

12.B.4.A. Condición que lleva a este flujo alternativo

1. Paso 1

2. Paso 2

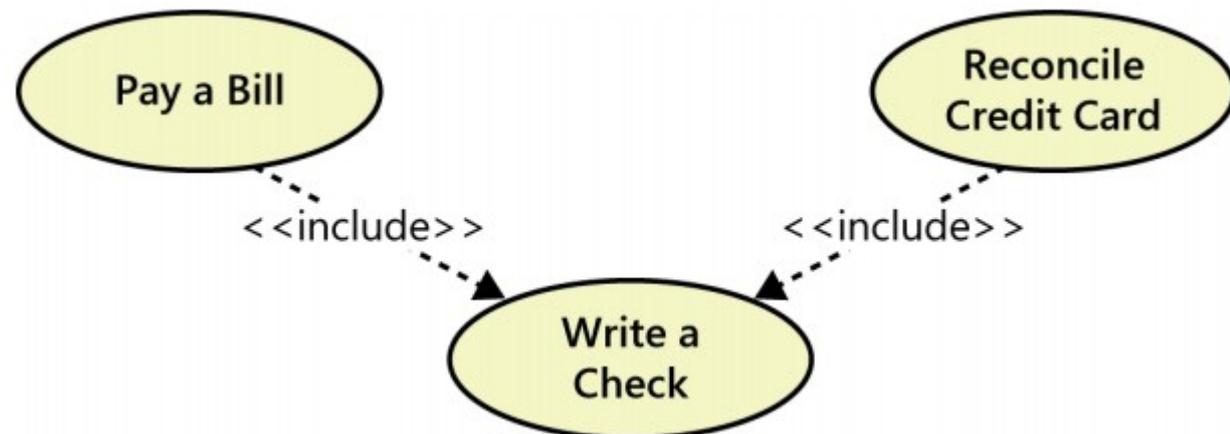
...

Casos de uso expandidos

- Usualmente se crea una primera versión del caso de uso que no contempla los flujos normales y alternativos, sino que incluye únicamente descripción, precondiciones y poscondiciones. A este estilo de escritura se le llama **alto nivel**.
- En estos casos a la versión que incluye los flujos normales y alternativos se le llama **caso de uso expandido**.

Relaciones entre CUs - Inclusión

- Permiten definir casos de uso que "ejecutan" otros casos de uso.
- Al caso de uso que incluye a otro se le llama caso base y al otro se le denomina caso incluido.
- Se puede incluir otros casos de uso tanto en el flujo principal como en algún flujo alternativo.



Relaciones entre CUs - Inclusión

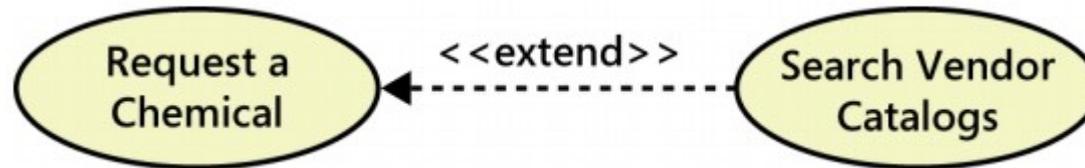
- Es como ejecutar la totalidad del caso de uso incluido y luego continuar en el punto siguiente del caso base.
- Sólo se continúa la ejecución del caso base si el caso incluido se ejecutó con éxito.

1. El usuario ingresa al sitio.
2. <<incluye>> "Autenticar"
3. El sistema le ofrece un menú de servicios.
4. Usuario: indica al sistema que desea realizar una transferencia
5. Sistema: lista las cuentas del usuario
6. ...

Relaciones entre CUs - Extensión

- La relación de extensión refiere a un fragmento de un caso de uso que extiende, agrega comportamiento, a otro caso de uso.
- Se utilizan para describir escenarios alternativos complejos.
- Una extensión sólo se ejecuta cuando se cumple una condición particular en un punto específico del caso de uso a extender. A ese punto se le llama punto de extensión.

Relaciones entre CUs - Extensión



- En el caso de ejemplo, *"Request a Chemical"* tiene como objetivo solicitar un químico de la proveeduría y tiene como alternativo solicitar el químico de un vendedor.
- El solicitante puede ejecutar el caso de uso *"Search Vendor Catalogs"* cuando solicita un químico, como parte del procesamiento del flujo alternativo.

Relaciones entre CUs – Extensión – Ejemplo

- **Caso de uso:** Retirar dinero de cajero
- **Flujo principal:**
 1. Incluye el caso de uso: Identificar Cliente
 2. El CA despliega las distintas alternativas disponibles: retiro, depósito, consulta
 3. El Cliente elige Retiro
 4. El CA pide cuenta y monto
 5. El Cliente los ingresa
 6. CA envía código de Tarjeta, PIN, cuenta y monto al SC
 7. El SC contesta: OK
 8. El Cliente pide dispensar el dinero
 9. El CA dispensa el dinero
 10. El CA devuelve la tarjeta
 11. El CA imprime el recibo
- **Puntos de Extensión:**
 - Retiro de Monedas: En el punto 8 del flujo principal

Relaciones entre CUs – Extensión – Ejemplo

- **Caso de Uso:** Retirar monedas
- **Descripción Breve:** El cliente opcionalmente puede querer retirar monedas.
- **Flujo Principal:**
Extensión de Retirar en el punto Retirar Monedas, el cliente también puede elegir “monedas”, en ese caso:
 1. El Cliente elige retirar monedas, especificando tipos de monedas y la cantidad de rollos para cada uno.
 2. El CA calcula el importe a retirar para cada moneda y el total y lo muestra.
 3. El Cliente confirma.

Estrategias para identificar casos de uso

- Identificar primero a los actores, luego a los procesos que serán soportados por el sistema y por último definir casos de uso para las actividades en las cuales interactúan los actores y el sistema.
- Crear escenarios específicos para ilustrar cada proceso de negocio, luego generalizar en casos de uso e identificar actores.
- Identificar los eventos externos a los cuales el sistema debe responder, luego relacionar esos eventos a los actores participantes y casos de uso específicos.
- Identificar que entidades de datos necesitan casos de uso de creación, lectura, actualización, borrado u otras operaciones de manipulación.

Consideraciones

- No tratar de forzar a que todos los requisitos sean contemplados por los casos de uso. En algunos casos no aporta demasiado valor hacerlo.
- Ser concretos y mantenerse dentro de los límites de los casos de uso (precondiciones y poscondiciones).
- Evitar escribir demasiados casos de uso. Evaluar si son todos necesarios y si no corresponden a distintos escenarios de un mismo caso de uso.
- No crear casos de uso muy complejos. En particular, no incluir demasiados chequeos de errores y validaciones comunes.

Consideraciones

- Los casos de uso no tienen porque desarrollarse por completo de una vez. En particular, no utilizar demasiado tiempo en detallar casos de uso que serán utilizados como insumo para otras actividades que comienzan dentro de meses o años.
- No se deberían incluir en los flujos detalles de la interfaz gráfica. Pero pueden agregarse secciones, por ejemplo, para incluir borradores de pantallas.
- No se debería incluir información de diseño o desarrollo. Así tampoco información sobre la definición de los datos.
- No escribir casos de uso que no sean entendidos por los usuarios.

Ingeniería de Software

Próxima clase: Especificación de requisitos y
Requisitos en metodologías ágiles

Qué tengo que hacer?

Leer Sommerville capítulo 4 – sección 4.4

Leer Wiegers capítulo 11.

Leer User Stories applied for agile software development – capítulos 1,
2, 3 (User Roles) y 7