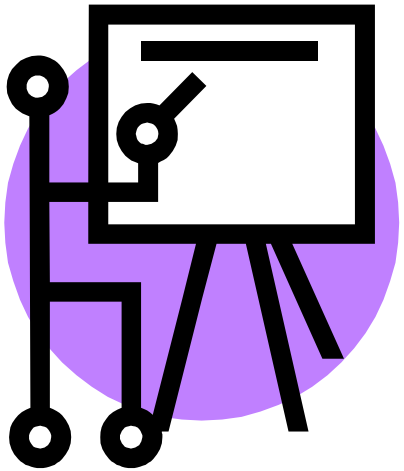
Tesina de grado
Licenciatura en Sistemas

Selección e Implementación de Prácticas de Diseño Centrado en el Usuario en un Proceso RUP

Ulises Chesini

¿De qué se trata la tesis?



- Diseño centrado en el usuario y usabilidad
- RUP
- RUP no es Centrado en el Usuario
- Trabajos previos
- Casos de Uso
- Casos de Uso con foco en el usuario
- Patrones de casos de uso
- Prototipos de IU
- Templates de IU

Diseño Centrado en el Usuario y Usabilidad

- El DCU es el conjunto de prácticas, métodos y herramientas que dirige el desarrollo de un producto de software que persigue como atributo de calidad principal la usabilidad del sistema generado.

- Usabilidad es el grado en el que un producto puede ser utilizado por los usuarios para alcanzar las metas especificadas con eficacia, eficiencia y satisfacción.

- La usabilidad puede verse como una consecuencia de la aplicación correcta de un proceso de desarrollo centrado en el usuario.

DCU - Principios básicos

- Participación activa de los usuarios
- Hacer foco temprano en los usuarios y sus tareas
- Actividades de diseño integradas
- Prototipar y evaluar iterativamente
- Filosofía de DCU

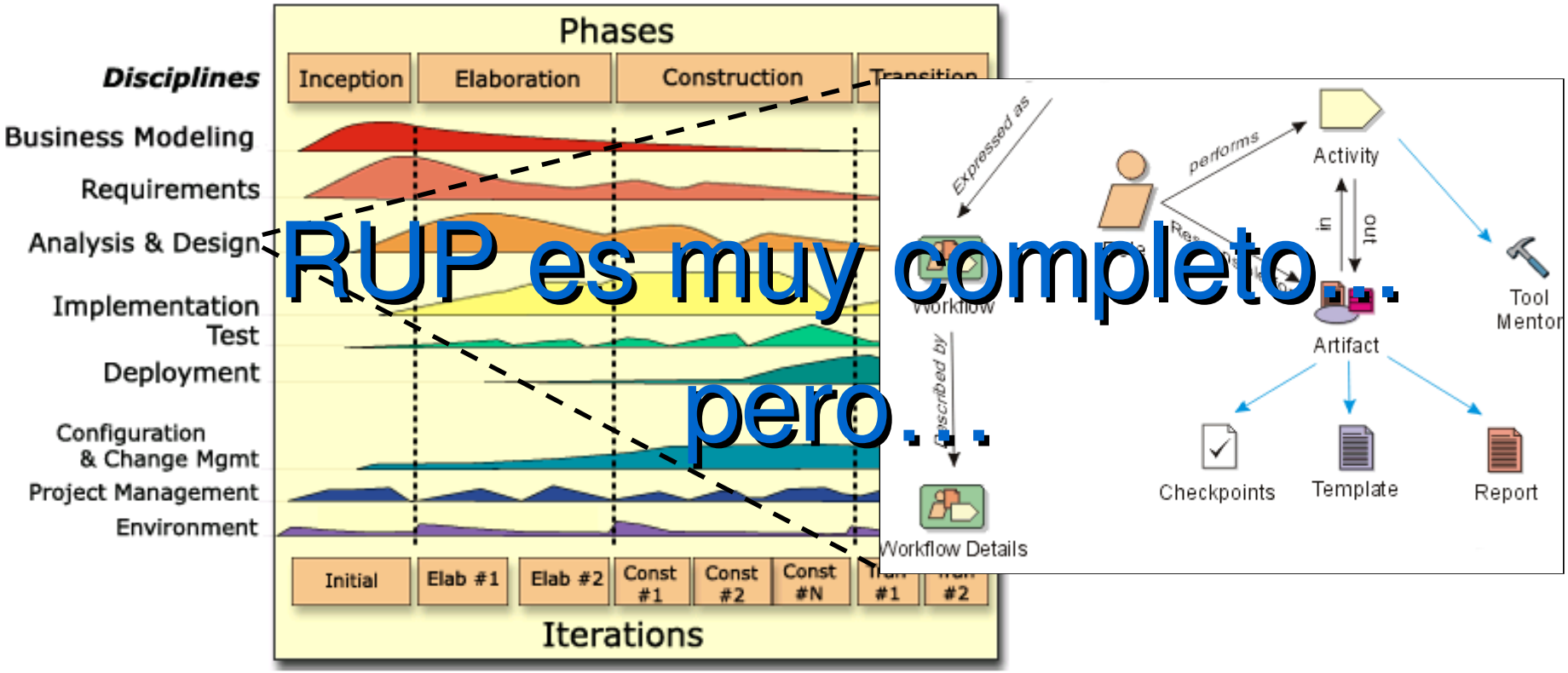
Ventajas de un sistema usable

- Aumenta la satisfacción del usuario
- Facilita el aprendizaje y el uso
- Reduce los costos de desarrollo y mantenimiento
- Aumenta las ganancias
- Mejora la eficacia y la productividad
- Aumenta la tasa de éxitos y reduce los errores de usuario
- Disminuye los costos de soporte y capacitación
- Aumenta el retorno de la inversión

RUP

- Está fundamentado en tres conceptos claves:
  - Basado en Casos de Uso
  - Es Iterativo e Incremental
  - Centrado en la Arquitectura

RUP



RUP es muy completo...

pero...

RUP no es un proceso centrado en el usuario

- 30 roles, ninguno para el usuario
- ¿Cómo se menciona a los usuarios?
  - *"Un **Actor del Negocio** representa un rol jugado por alguien **o algo** en relación al ambiente del negocio. "*
  - *"Un **trabajador de negocio** se usa para representar los roles que los humanos **o el software** tengan en la organización."*
  - *"Un **actor** define un conjunto de roles que usuarios del sistema pueden jugar cuando interactúan con él. Un actor puede **ser una persona o un sistema externo**".*

RUP no es un proceso centrado en el usuario

- No se describe como el usuario debe participar de:
  - la elaboración de la vision y alcance
  - la definición y revisión de los requerimientos funcionales y no funcionales
  - la definición y revisión de los casos de uso
  - la definición y revisión de los prototipos de interfaz de usuario
  - las pruebas de usabilidad
  - la aceptación final del producto

RUP no es un proceso centrado en el usuario

- RUP adopta UML como lenguaje de modelado
  - UML no provee suficiente soporte para el diseño de interfaces de usuario.
  - Por lo tanto, RUP no ofrece una notación, *templates* ni *checklists* para el diseño de IU
- Tampoco ofrece lineamientos sobre el proceso de diseño de UI o sobre cómo hacer análisis de la misma.
- Mapa de ruta de Usabilidad - RUP

Trabajos relacionados

- Usability Design – Extending RUP with a new discipline - Goransson, Lif y Gulliksen
- In support of user interface in the RUP – Phillips y Kemp
- Key principles for user-centred systems design - Goransson y Gullitksen
- RUPi - Sousa y Furtado

RUP está basado en Casos de Uso

- ¿Qué es un Caso de Uso?
  - "Define un conjunto de instancias de uso, donde cada instancia es una secuencia de acciones que el sistema realiza y lleva a un resultado observable de valor para un actor particular" [Jacobson, 1999].

- Los casos de uso modelan comportamiento, **interacción entre el usuario y el sistema**.

RUP está basado en Casos de Uso

- Un CU suele tener esta forma:

Caso de uso: Extraer dinero de Caja de Ahorro
Pre-condición: El usuario se registró en el sistema con su pin de acceso
Pos-condición: Se realizó la extracción. El saldo de la caja de ahorro se decrementó por la suma del monto extraído. Se realizó el asiento en el sistema contable.

00_Extracción realizada exitosamente ***(escenario principal)***
1. El usuario solicita extraer dinero de la caja de ahorro
2. El sistema solicita el pin de acceso
3. El usuario ingresa el pin de acceso
4. El sistema verifica que haya suficiente crédito.
5. El sistema realiza a extracción.
6. FIN

A1_ El pin de acceso es incorrecto **(*escenario alternativo)***
2.a. Condición: El pin ingresado es incorrecto.
2.a.1. El sistema informa que el pin es incorrecto.
2.a.2. Va al paso 2 del escenario principal

E1_No hay crédito ***(escenario de excepción)***
4.a. Condición: No hay crédito en la caja de ahorro
4.a.1. El sistema informa al usuario que no hay crédito.
4.a.2. FIN

Casos de uso con foco en los usuario

- Un problema común con los CU
  - No existe una forma estandarizada de escribirlos
  - Escritos con foco en el sistema

Pautas para orientar casos de uso en el usuario

- Granularidad
- Agrupación por tareas (situaciones de trabajo)
  - Tareas de nivel mayor para cada actor
  - CU lógicamente conectados
  - CU conectados secuencialmente en el tiempo
  - CU que requieren acceso al mismo conjunto de información
- Referencias a la Interfaz de Usuario
  - Casos de uso escenciales en las primeras iteraciones

Pautas para orientar casos de uso en el usuario

- ¿Por qué es importante identificar situaciones de trabajo?
  - encontrar tempranamente una navegación apropiada entre pantallas.
  - encontrar patrones de interacción que se puedan reusar en otras situaciones de trabajo.
  - detectar fácilmente candidatos a patrones de casos de uso.

Pautas para orientar casos de uso en el usuario

- Resumen:
  - Generar la lista inicial de CU pensando en las tareas que realiza el usuario (modelo de tareas).
  - Agrupar esas tareas en “situaciones de trabajo”.
  - Escribir los CU con formato de CU esencial en las primeras iteraciones.
    - *Luego enriquecer los CU con navegaciones conocidas y referencias a las interfaces que son incorporadas al sistema.*
  - En iteraciones avanzadas intentar encontrar patrones de casos de uso [ahora los explico].
  - Refinar los patrones en base al *feedback*

Patrones de Casos de Uso

- Consistencia:
  - “Principio por el cual las cosas que están de alguna forma relacionadas se deben presentar y tratar en una forma similar, mientras que las que no están relacionadas deben presentarse y tratarse de manera distinta”.
  - Para tareas (casos de uso) similares el sistema debe comportarse de la misma manera independientemente del módulo en el que se encuentre.

Patrones de Casos de Uso

- Patrón de caso de uso:
  - "*template* con los elementos comunes a un conjunto cohesivo de casos de uso, que abstrae una forma particular en que el usuario interactúa con el sistema para cumplir un objetivo".

Patrones de Casos de Uso

Responsable: Analista F	Responsable: Analista C
Caso de uso: Autorizar Factura	Caso de uso: Autorizar Orden de Compra
Pre-condiciones: -usuario logueado	Pre-condiciones: -usuario logueado -Orden de Compra seleccionada en estado habilitado para ser autorizada
•El usuario realiza una búsqueda según CU1_Buscar Facturas •El usuario selecciona una factura •El usuario elige la opción autorizar •El sistema valida las condiciones para autorización RN01_Validaciones de autorización de factura •El sistema pide la clave necesaria •El usuario ingresa la clave •El sistema valida la clave •El sistema deja a la factura en estado autorizado. •FIN A2. 7. la clave ingresada no es correcta 7.1 el sistema informa al usuario que la clave no es valida 7.2 va al paso 5 del escenario principal. E1. 4. Las validaciones fallan 4.1 el sistema informa al usuario que falló la validación 4.2 va al paso 9	•El usuario elige la opción autorizar •El sistema pide la clave necesaria •El usuario ingresa la clave •El sistema valida la clave •El sistema deja a la Orden de Pago en estado autorizado. •FIN A2. 7. la clave ingresada no es correcta 7.1 el sistema informa al usuario que la clave no es valida 7.2 va al paso 4 del escenario principal.

Patrones de Casos de Uso

Caso de uso: Autorizar Factura	Caso de uso: Autorizar Orden de Pago
Pre-condiciones: -Usuario logueado -Factura en estado autorizable	Pre-condiciones: -Usuario logueado -Orden de pago estado autorizable
•El usuario realiza una búsqueda según CU1_Buscar Facturas •El usuario selecciona una Factura •El usuario elige la opción autorizar •El sistema valida las condiciones para autorización RN01_Validaciones de autorización de Factura •El sistema pide la clave necesaria •El usuario ingresa la clave •El sistema valida la clave •El sistema deja a la Factura en estado autorizado. •FIN A2. 7. La clave ingresada no es correcta 7.1 el sistema informa al usuario que la clave no es valida 7.2 va al paso 5 E1. 4. Las validaciones fallan 4.1 el sistema informa al usuario que falló la validación 4.2 Continúa en el paso 9	•El usuario realiza una búsqueda según CU1_Buscar Ordenes de pago •El usuario selecciona una Orden de pago •El usuario elige la opción autorizar •El sistema valida las condiciones para autorización RN09_Validaciones de autorización de Ordenes de pago •El sistema pide la clave necesaria •El usuario ingresa la clave •El sistema valida la clave •El sistema deja a la Orden de Pago en estado autorizado. •FIN A2. 7. La clave ingresada no es correcta 7.1 El sistema informa al usuario que la clave no es valida 7.2 va al paso 5 E1. 4. Las validaciones fallan 4.1 El sistema informa al usuario que falló la validación 4.2 Continúa en el paso 9

Patrones de Casos de Uso

Caso de uso: Autorizar **<Comprobante>**
Pre-condiciones: -usuario logueado
•El usuario realiza una búsqueda según **<CU_Buscar comprobante>** •El usuario selecciona un **<Nombre de comprobante>** •El usuario elige la opción autorizar •El sistema valida las condiciones para autorización **<RNXX_Validaciones de autorización de tipo comprobante>** •El sistema pide la clave necesaria •El usuario ingresa la clave •El sistema valida la clave •El sistema deja al comprobante en estado autorizado. •FIN A2. 7. La clave ingresada no es correcta 7.1 el sistema informa al usuario que la clave no es valida 7.2 va al paso 5 E1. 4. Las validaciones fallan 4.1 el sistema informa al usuario que falló la validación 4.2 Continúa en el paso 9

Patrones de Casos de Uso

- Ventajas:
  - Representan la captura y formalización del conocimiento sobre la interacción con el sistema y la forma de modelarlo
  - Optimizar la forma de interacción usuario-sistema
  - Facilitar la comunicación con otros miembros del equipo de desarrollo
  - Unificar criterios
  - Facilita las tareas de Control de Calidad, Diseño y Testing

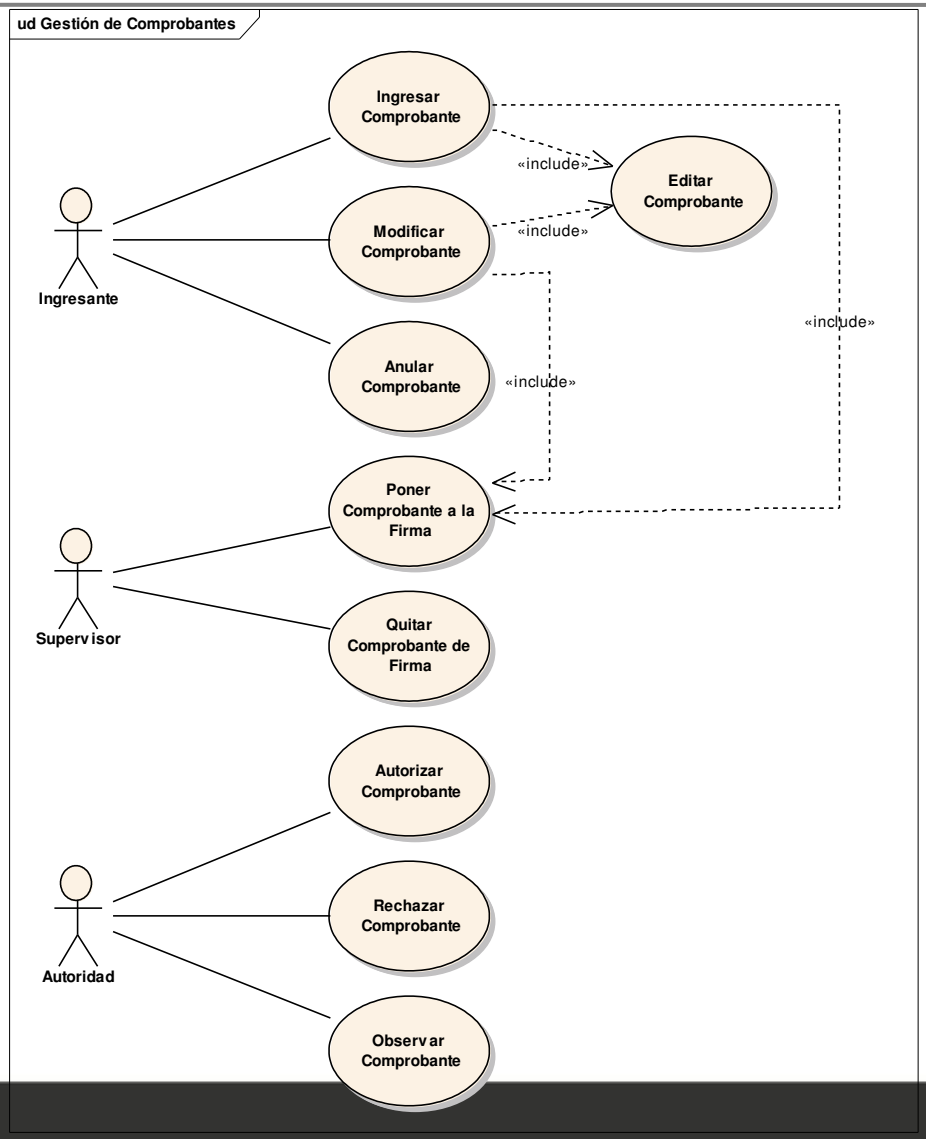
- Los patrones representan interacciones óptimas, o casi óptimas, entre el usuario y la clase de tarea o actividad que el patrón realiza.

Patrones de casos de uso

- Otras Ventajas:
  - Reduce los tiempos de especificación de requerimientos y aumenta la productividad de la tarea de relevamiento y análisis de requerimientos.
  - Genera requerimientos más completos y con una menor proclividad a tener errores.
  - Genera estimaciones más precisas.
  - Facilita las tareas de controles de calidad.
  - Favorece la abstracción y reuso de soluciones de diseño e implementación.

Extendiendo el patrón a un conjunto de casos de uso

- ¿Recuerdan las Situaciones de trabajo?



Prototipos de Interfaz de Usuario

- "Ejemplo de la interfaz de usuario final que se construye para explorar y/o validar el diseño de la interfaz de usuario"
  - Ayudan a explorar el dominio del problema junto a los *stakeholders* en la etapa de análisis.
  - Ayuda a explorar el dominio de la solución en la etapa de diseño.
  - Comunican diferentes diseños de IU posibles.
  - Base para explorar la usabilidad del sistema.

Prototipos de Interfaz de Usuario

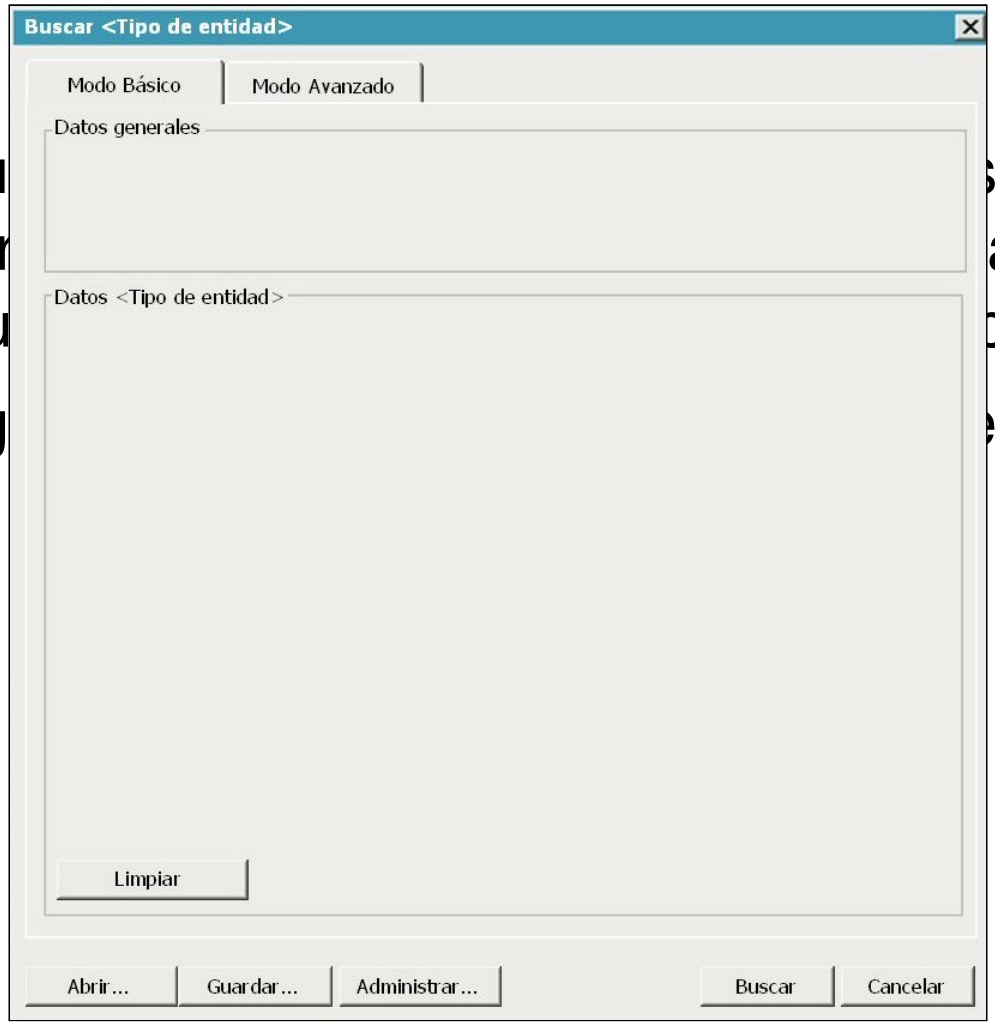
- Según RUP
  - Rol de diseñador de interfaces de usuario
  - Debe venir de un *background* creativo y de artes visuales, en vez de uno informático o de ingeniería.
  - En proyectos grandes, un grupo separado de personas deben dedicarse a cubrir este rol exclusivamente.
  - Pero...mantener un grupo de personas dedicadas solamente a crear prototipos de interfaces no es justificable económicamente.

Prototipos de Interfaz de Usuario

- Prototipar en forma temprana: DCU
- User Experience Storyboards: RUP
  - Representación abstracta de los elementos de IU, pantallas y contenidos
  - Ventajas
    - Agrega valor al construir una firme base una arquitectura de interfaz fuerte.
    - Son más rápidos de construir que un prototipo.
    - Facilita la comunicación entre los equipos.
    - Ayuda al equipo de IU a determinar qué debe ir en las interfaces antes de decidir cómo deben verse o de que se comiencen a construir.

Templates de Interfaz de Usuario

- Un nú s cubren el mayor as: búsqu pantallas de config es



Templates de Interfaz de Usuario

- Ventajas:
  - No se crea cada pantalla desde cero una y otra vez
  - Predefine cuestiones de navegación y comportamiento de la interfaz
  - Estandariza las interfaces
- Un *template* debe incluir documentación
  - Cuándo y cómo usarlo
  - Buenas prácticas para el uso
- Es recomendable el uso de alguna herramienta de prototipado que permita transformar el *template* en una pantalla concreta fácil y rápidamente

Templates de Interfaz de Usuario

- Relación con los patrones de casos de uso
  - un patrón -> un *template* de IU
- Relación con la Arquitectura
  - RUP es centrado en la Arquitectura
  - Una arquitectura describe la estructura de la aplicación en términos de componentes
  - Los *templates* ayudan a encontrar componentes de arquitectura
  - Esto permite reusar estándares de usabilidad

Repasemos

RUP

DCU - Usabilidad

RUP no es DCU

Casos de uso centrados
en usuarios

Patrones de Casos de uso

Templates de IU

Trabajo Futuro

- Métricas
- Más prácticas en otras disciplinas
- Refinar la relación con la Arquitectura

¿Dudas, preguntas,comentarios?

Fin

Muchas gracias!