

Casos de uso una: propuesta para la reunión de requerimientos

Maria de los Angeles Fernandez Benassati Pablo Fillottrani
Departamento de Ciencias e Ingenieria de la Computación
Univeridad Nacional del Sur
Av. Alem 1253 – 8000 Bahía Blanca
mdominel@uns.edu.ar prf@cs.uns.edu.ar

1 Introducción

En este trabajo se estudia la propuesta “conducido por casos de uso” [1] para la recolección y documentacion de requerimientos del usuario en forma exitosa. Esta técnica difiere substancialmente de los enfoques tradicionales ya que emplea herramientas tomadas de las tecnologías tradicionales como así también introduce algunos cambios e implementa otras completamente nuevas.

Existen varios principios y guías para alcanzar en forma satisfactoria los requerimientos de los usuarios y entre los cuales se pueden enumerar:

- a. Reducir el riesgo mediante un enfoque iterativo o incremental involucrando al usuario en reveer o escribir casos de uso.
- b. Enfocar las interacciones del negocio, para ello se debe separar el análisis del diseño y mantener los casos de uso desprovistos de consideraciones técnicas.
- c. Reducir el volumen de requerimientos.
- d. Eliminar requerimientos duplicados e inconsistentes.
- e. Crear requerimientos que el usuario pueda entender fácilmente.
- f. Crear requerimientos que sean útiles a los diseñadores y a los desarrolladores.

Esta técnica efectiviza e implemeta todos estos principios para poder así alcanzar las funcionalidades que los usuarios esperan del sistema en forma adecuada.

2 Cuatro pasos para la captura y documentacion de requerimientos

Esta propuesta indica que los requisitos del usuario sean creados en forma iterativa e incremental. De esta manera se decrementarán los riesgos ya que son tratados en las fases tempranas del ciclo de vida.

La creación iterativa es altamente dependiente de cada situación individual, y seria inútil sugerir que exactamente el mismo número de iteraciones se necesitan para completar los requerimientos en todos los casos. Sin embargo es posible decir que la especificación iterativa de requerimientos siempre atraviesa por mismos pasos lógicos en toda ocasión.

En esta técnica existen cuatro pasos lógicos: esbozo, ampliación, enfoque y retoque. A cada uno de estos pasos se le asignó a una iteración y un nombre que son los siguientes:

- Facade
- Filled
- Focus
- Finish

A través de estas iteraciones se crean y se refinan un conjunto de requerimientos para entregar al usuario o los desarrolladores.

Así mismo ofrece un cubrimiento amplio de las siguientes partes del ciclo de vida del desarrollo de software

- Sentencia del problema

- Sentencia de trabajo
- Análisis del Riesgo
- Prototipo
- Diagrama de casos de uso y casos de uso
- Catálogo de reglas de negocio

2.1 Iteración “Facade”

Objetivos:

Su propósito es crear un esqueleto para ser completado con la ayuda del usuario en las próximas iteraciones.

Un caso de uso “Facade” consiste solo de la información mínima necesaria, incluye un nombre y una breve descripción. Se identifican también los principales actores.

Escribir esta iteración es difícil debido a que aun no se tienen los conceptos claros de la aplicación.

Tan pronto como se desarrolle una definición del sistema propuesto se debe llamar a todas las fuentes de información consultadas para que den ideas y opiniones.

Pasos en la iteración “Facade” son los siguientes:

1. Crear una sentencia del problema.
2. Identificar y rever la documentación existente.
3. Tomar el punto de vista de ejecutivo exponsor.
4. Identificar los usuarios, el grupo de gerentes, los supervisores ,los usuarios operativos y los dueños del los datos.
5. Entrevistar a los supervisores.
6. Encontrar a los actores.
7. Hacer el diagrama de contexto y crear los casos de uso Facade.
8. Comenzar con las reglas de negocio.
9. Crear el análisis de riesgo.
10. Crear la sentencia de trabajo.
11. Tener la aprobación informal del ejecutivo exponsor.

Entregas:

Esta iteración esta completa cuando, el caso de uso del contexto del sistema esta documentado y el líder del proyecto tiene la confianza que el 80% de los casos de uso han sido identificados. Las salidas de esta iteración son las siguientes:

Sentencia del problema	Completo
Sentencia de trabajo	Completo
Análisis de riesgo	parcialmente completo
Prototipo	no iniciado
Diagrama de casos de uso	nivel facade
Casos de uso	nivel facade
Reglas de negocio	nivel facade

2.2 Iteración “Filled”

Objetivos:

El objetivo de esta iteración es crear un conjunto de casos de usos y reglas de negocios que describan la aplicación completa. Aunque estos requerimientos aún sean vagos, contienen más detalle e incluyen requerimientos funcionales y no funcionales.

Durante esta iteración se invierte el tiempo entendiendo los requerimientos y explorando posibles soluciones. Se deben escribir los casos de uso basándose en las interacciones de los usuarios con el sistema y analizando los procesos.

Debido a la naturaleza exploratoria de este proceso produce requerimientos bien documentados que beneficia al resto del equipo de desarrollo.

Pasos en la iteración filled son los siguientes:

1. Detallar los casos de uso.
2. Crear los caso de uso filled.
3. Identificar los triggers.
 - a. Identificar las precondiciones.
 - b. Especificar el curso básico de eventos.
4. Documentar los caminos alternativos y las excepciones.
5. Buscar y documentar requerimientos no funcionales.
6. Agregar reglas de negocio.
7. Testear los casos de uso filled con escenarios.

Entregas:

Lista de casos de uso candidatos	Parcialmente completo
Diagrama de casos de uso	nivel filled
Casos de uso	nivel filled
Catalogo de Reglas de negocio	Parcialmente completo
Escenarios	varios para cada caso de uso testeado

2.3 Iteración “Focused”

Objetivos:

Esta iteración tiene como objetivo principal separar la parte esencial de la que no lo es. Es ahora cuando se decide que es importante, que se construirá y porque. Por eso se examina el negocio desde el contexto con la solución propuesta, para asegurarse que no se esta resolviendo un problema innecesario.

Cuando esta iteración esta completa, los casos de usos resultantes modelarán las interacciones de los usuarios con el sistema, los detalles que se entendieron del mismo, su alcance, su complejidad y los riesgos involucrados.

Pasos en la iteración son los siguientes:

1. Crear la matriz de contexto.
2. Identificar y remover procesos duplicados e inconsistentes.
3. Usar las relaciones de extend o include para mejorar la funcionalidad.

Hay que tener en cuenta que el resto del equipo de desarrollo usará estos casos de uso para construir el sistema, es por ello que cada caso de uso debe ser completo, exacto y conciso.

En esta iteración es fundamental la revisión, por parte de los usuarios expertos en la materia. Para ello se deben revisar cada caso de uso individual y como parte de sistema completo. La persona que revisa debe asegurar que los casos de uso describen el sistema adecuadamente. Luego el cliente indica que la salida de esta iteración es aceptable, que las suposiciones son válidas y que el camino elegido es el deseado.

Entregas:

Sentencia del problema	Completo
Sentencia de trabajo	Completo
Casos de uso	nivel focused
Diagrama de casos de uso	nivel focused
Reglas de negocio	Al menos completas
Análisis de riesgo	en camino
Prototipo	no iniciado

2.4 Iteración “Finished”

Objetivos:

Finalmente el objetivo de esta iteración es integrar los requerimientos no funcionales con los casos de usos, se agregan los requerimientos de la interfase del usuario y finalmente se empaqueta la documentación para el proceso de diseño.

Al finalizar esta fase se debería poder visualizar el sistema completo y tener la suficiente información para planificar un desarrollo iterativo.

Pasos en la iteración “finished” son:

1. Agregar la interfase del usuario a cada caso de uso.
2. Abstractar y combinar los requerimientos no funcionales y asociarlos a cada caso de uso.
3. Tomar las decisiones finales. Cada caso de uso debería ser evaluado siguiendo este criterio.
 - a. Especificar requerimientos del sistema para el cliente.
 - b. Proveer entradas para los diseñadores.
 - c. Proveer información para testeos, entrenamiento y documentación.
 - d. Describir una vista del sistema deseable para el expositor del proyecto.
 - e. Documentar toda la información que pasará a la próxima actividad.

Entregas:

Extracto del problema	Completo
Extracto de trabajo	Completo
Casos de uso	Completo
Diagrama de casos de uso	Completo
Catalogo de reglas de negocio	Completo
Análisis de riesgo	Completo
Interfaces con el usuario	Completo
Prototipo	completo.

3 Pasos a seguir

Para continuar con la metodología de desarrollo implementada por los mismos creadores de UML, denominada el *Proceso Unificado de Desarrollo* [3], aún quedan por ver las siguientes etapas a las que se le intentarán encontrar pasos lógicos sobre los cuales iterar:

Del Análisis de Requerimientos: Resta desarrollar el modelo del mundo: que se representa en un diagrama de “estructura estática de clases.”

Diseño del sistema:En esta etapa se define una subdivisión en aplicaciones del sistema (si es lo suficientemente grande) y la forma de comunicación con los sistemas ya existentes con los cuales debe interactuar .

Diseño detallado:En esta etapa se adecúa el análisis a las características específicas del ambiente de implementación y se completan las distintas aplicaciones del sistema con los modelos de control, interfaz o comunicaciones, según sea el caso.

Implementación y pruebas:Se desarrolla el código de una manera certificada.

4 Referencias

Daryl Kulak y Eamonn Guiney, “Use Case Requirements in context”

G. Booch, J. Rumbaugh y I. Jacobson, "El Lenguaje Unificado de Modelado"

I. Jacobson, G. Booch, J. Rumbaugh , "El Proceso Unificado de Desarrollo"

Doug Rosenberg, Kendall Scott, “Use Case Driven Object Modeling with UML: A Practical Approach “