



junio del 2001

informática

GESTIÓN DE REQUERIMIENTOS PARA LA PROGRAMACIÓN A DISTANCIA

Resumen / Abstract

En el trabajo se propone un conjunto de plantillas en forma de cuestionarios que permiten definir los requerimientos por los clientes de manera estandarizada. Se incluye un método de trabajo que se basa en la definición jerárquica de los requerimientos funcionales. Para la realización de la gestión de los requerimientos se propone una herramienta de software que permita el control de los cambios y facilite generar documentos para lograr el objetivo fundamental de la gestión de requerimientos: el acuerdo con los clientes. Se incluye en la herramienta la utilización de los casos de uso y de los diagramas de actividad, ambos artefactos de UML.

This work intends a group of insoles in form of questionnaires that allow to define the requirements for the clients in a standardized way. A work method is included that is based in the hierarchical definition of the functional requirements. For the realization of the administration of the requirements it intends a software tool that allows the control of the changes and it facilitates to generate documents to achieve the fundamental objective of the administration of requirements: the agreement with the clients. It is included in the tool the use of the cases of use and of the activity diagrams, both artefacts of UML.

Palabras clave / Key words

Requerimientos de software, modelo CMM

Software requirements, model CMM (Capability Maturity Model)

INTRODUCCIÓN

Los países en desarrollo tienen la posibilidad de desarrollarse en el área del software, ya que los productos de software son puramente valor adicionado.^{1,2} Algunos países ya lo han logrado, Chile ha sido el principal exportador de software de América Latina desde 1992 y la India es el país de mayores logros entre los países en vías de desarrollo. Para los países en vías de desarrollo la realización de software a distancia para los países del primer mundo es una posibilidad real de obtención de ingresos.

Una de las áreas de procesos que más incide en la realización de software a distancia es la correspondiente a la gestión de requerimientos. La no realización adecuada de la gestión de requerimientos es una de las causas de los fracasos de los proyectos. Lo anterior ha provocado que a este aspecto se le brinde una atención especial en los últimos años. Corrobora la afirmación anterior el hecho de que en el modelo CMM (Capability Maturity Model).³ se incluya entre las áreas de proceso del nivel 2 a la gestión de requerimientos.

En el caso de los proyectos realizados a distancia la situación aún es más crítica ya que la interacción con los usuarios potenciales prácticamente es nula y las exigencia de calidad de los productos es superior. En este trabajo se presenta un procedimiento de trabajo a utilizar por la

Sofía Álvarez Cárdenas, Ingeniera Hidráulica, Doctora en Ciencias Técnicas, Profesora Titular, Centro de Estudios de Ingeniería de Sistemas (CEIS), Instituto Superior Politécnico José Antonio Echeverría (ISPJAE)
e-mail:sofia@ceis.ispjae.edu.cu

Arturo Aguiar Vila, Ingeniero Informático, ISPJAE

Ernesto Aguiar Alonso, Ingeniero Informático, ISPJAE

Yaquelin González Padilla, Ingeniera Informática, ISPJAE, Ciudad de La Habana

entidad productora de software subcontratada para la realización de software a distancia que pretende organizar y perfeccionar la gestión de requerimientos en esas circunstancias. Se incluye la definición de una plantilla que ayuda a la adquisición de los requerimientos por la entidad subcontratada.

GESTIÓN DE REQUERIMIENTOS

El término requerimiento (o requisito) puede definirse como una condición que debe cumplirse o una capacidad que debe tener el sistema.⁴

CMM (Capability Maturity Model),³ plantea que el propósito de la gestión de requerimientos es establecer un entendimiento común entre el cliente y el proyecto de software de los requerimientos del usuario que serán abordados por el proyecto de software.

El entendimiento común entre clientes y usuarios se establece a través de un contrato o compromiso y contiene los requerimientos técnicos y los no técnicos por lo que este constituye la base para la estimación, el planeamiento, ejecución y seguimiento de las actividades del proyecto de software a través del ciclo de vida del software. Si los requerimientos del sistema asignados al software son cambiados, entonces las actividades, productos de trabajo y planes del software afectados son ajustados también.⁵

El procedimiento de análisis de requerimientos consta de los siguientes pasos:⁴

- Análisis del problema.
- Definir el sistema.
- Control del alcance del proyecto.
- Refinamiento de la definición del sistema.
- Gestión de los requerimientos que cambian.

Análisis del problema

Se realiza para comprender el problema, las necesidades de los involucrados y proponer soluciones de alto nivel por lo que se analiza para ello el proyecto desde el punto de vista de los negocios. Se emplean las técnicas de recopilación de información tales como: tormentas de ideas, entrevistas, prototipos, cuestionarios, y análisis competitivos. El resultado debe ser una lista de solicitudes o necesidades descritas textual y gráficamente que se les ha asignado prioridades relativas entre ellas.

Definir el sistema

Definir el sistema significa convertir las necesidades de los involucrados en una descripción significativa de lo que será el sistema. En las primeras etapas de la definición del sistema se definen los requerimientos, el detalle de los requerimientos así como la prioridad de estos. También se calcula el esfuerzo estimado, riesgos, y alcance inicial. Como parte de esta etapa pudieran incluirse prototipos iniciales, y modelos de diseño directamente vinculado a las solicitudes más importantes de los involucrados. El resultado de la definición del sistema es una descripción que contiene textos en lenguaje natural y gráficos.

Control del alcance del proyecto

Para dirigir un proyecto eficientemente es necesario darle prioridad cuidadosamente a los requerimientos, basándose en la información suministrada por todos los involucrados.

El desarrollo de muchos proyectos se ve afectado porque los desarrolladores se ponen a realizar aquellas facilidades que encuentran atractivas o interesantes en vez de darle prioridad en las etapas tempranas del desarrollo a aquellas tareas orientadas a disminuir los riesgos del proyecto. El sistema debe desarrollarse incrementalmente, escogiendo cuidadosamente en cada iteración los requerimientos que disminuyan riesgos conocidos del proyecto. Para lograrlo debe negociarse el alcance de cada iteración o incremento con los involucrados del proyecto.

Refinamiento de la definición del sistema

Se obtiene una definición detallada del sistema de forma que los involucrados puedan entender y llegar a un acuerdo firmado. Esta definición contiene la funcionalidad y el cumplimiento de cualquier requerimiento legal o regulatorio, de usabilidad, rendimiento, confiabilidad, soporte, mantenimiento, etc. Todos los esfuerzos deben encaminarse a lograr la comprensión de la audiencia, cosa que muchas veces implica generar documentos diferentes, para diferentes tipos de audiencias.

Los casos de uso, frecuentemente combinados con los prototipos son una herramienta muy poderosa para transmitir el propósito y definir los detalles del sistema. Estos ayudan a enmarcar los requerimientos en un contexto y narran cómo el sistema va a ser usado.

Gestión de los requerimientos que cambian

A pesar de que el proceso de definición de los requerimientos se efectúe con mucha precisión, siempre van a existir cambios en estos. Lo que hace los requerimientos difíciles de gestionar no es solo que el cambio en un requerimiento puede implicar tiempo perdido en desarrollar un nuevo módulo del sistema que probablemente ya estaba terminado, sino que también muchas veces el cambio de un requerimiento tiene un impacto considerable en otros requerimientos. Para ello, se deben indicar las dependencias que puedan existir entre requerimientos. Se debe incluir el control de versiones y dependencias al manejo de los requerimientos. Es importante incorporar un proceso de control de cambios que requiera que todos los cambios se revisen y sean aprobados por un equipo designado de miembros. Frecuentemente a este equipo se le denomina Comité de Control de Cambios (Change Control Board) y se define como el grupo representativo que dispone y coordina los cambios ingenieriles de un producto.

Requerimientos

La definición de los requerimientos de software se realiza de forma iterativa siendo un proceso que comienza con la definición inicial de los requerimientos y su completamiento posterior. Se recomienda la definición jerárquica de los requerimientos, de

manera de ir de un nivel de abstracción general a uno específico.

Los requerimientos se clasifican en funcionales y no funcionales, a continuación se detallan estos tipos.

Los requerimientos funcionales especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, utilizable, rápido o confiable. A continuación se propone una clasificación de ellos basada en la propuesta de RUP (Rational Unified Process),⁵ y de Suzanne y James Robertson.⁴

- Apariencia o interfaz externa.
- Usabilidad.
- Rendimiento.
- Soporte.
- Portabilidad.
- Seguridad y privacidad.
- Políticos y Culturales.
- Legales.
- Confiabilidad.
- Interfaz interna.
- Ayudas y documentación en línea.
- Hardware.
- Software.
- Restricciones en el diseño y la implementación.

PROCEDIMIENTO PARA LA GESTIÓN DE REQUERIMIENTOS CUANDO SE PROGRAMA A DISTANCIA

La gestión de requerimientos cuando se programa a distancia cobra características especiales, los requerimientos se envían en formas diversas por los clientes y los problemas generados por una comunicación inadecuada aún son más críticos. Por otra parte debe garantizarse que los productos requieran una menor actividad de soporte pues esto pudiera encarecer sensiblemente el producto y perderse las ganancias por estas razones o perderse al cliente en un futuro, por tanto, el asunto de la calidad del producto a obtener cobra una mayor importancia.⁶⁻⁸

Se propone el siguiente procedimiento basado en las mejores prácticas:⁹

1. Se emite una plantilla para que los clientes (empresa productora de software que subcontrata) definan los requerimientos. Se recomienda la plantilla presentada en el anexo 1.
2. La entidad subcontratada a partir de los requerimientos define los casos de uso para profundizar en los requerimientos y resuelve vía correo cualquier insuficiencia o cualquier ambigüedad presente en los requerimientos enviados según la plantilla del anexo 1.

Para cada requerimiento se debe definir cómo será probado en la aplicación. Se elabora un documento con las correcciones

incluidas que será fuente para la estimación de tiempo personas y esfuerzo así como para la realización del contrato con el cliente. Esta definición de requerimientos será motivo de control de configuración mediante el uso de un software de los disponibles en el mercado. Se recomienda reflejar la funcionalidad definida en un prototipo como medio de alcanzar el contrato con el cliente.

3. Se programa la realización de revisiones del software y en cada una de ellas se revisan las componentes de software contra los requerimientos de prueba definidos involucrados en la componente; todos los productos que constituyen la componentes y los utilizados en la prueba deben ser mantenidos bajo control de configuración.

4. Todos los involucrados (desarrolladores, clientes, usuarios) deben tener disponible los requerimientos, para ello se recomienda la publicación en internet de los requerimientos con los perfiles de usuarios bien definidos.

SOFTWARE DE APOYO AL PROCEDIMIENTO

El CASE (Computer Aided Software Engineering) desarrollado permite la realización de la gestión de requerimientos. Permite la definición jerárquica de requerimientos así como el control de cambios a los requerimientos. Se incluyen plantillas para los principales reportes necesarios para una gestión de requerimientos acorde con el procedimiento planteado antes. La herramienta permite la profundización de los requerimientos mediante la utilización de casos de uso y diagramas de actividad ambos artefactos del lenguaje de especificación UML (Unified Modeling Language).¹⁰

CONCLUSIONES

La realización de software a distancia es una posibilidad de ingreso importante para los países en vías de desarrollo. Sin embargo, en esas condiciones se requiere la obtención de productos de alta calidad para lo cual la gestión de requerimientos se convierte en un área de procesos vital. El procedimiento para la gestión de requerimientos presentado pretende ayudar a resolver esta problemática para la entidad productora de software subcontratada para la realización de software a distancia. Se recomienda el uso de un software de apoyo para la gestión de requerimientos. ☐

REFERENCIAS

1. BAEZA_YATES, R. A.: *Computing in Chile: The Jaguar or the Pacific Rim? Communications of the ACM*, Vol. 38, No 9 1995.
2. CORREA, C. M.: "Strategies for Software Exports from Developing Countries, *World Development*, Vol. 24, No. 1 pp. 171-182, 1996.
3. PAULK, M. C. et al.: *Capability Maturity Model for Software Version 1.1*, Carnegie Mellon University, SEI-93-TR-024, 1995
4. ROBERTSON, SUZANNE Y JAMES ROBERTSON. *Mastering the Requirements Process*, first edition, Addison Wesley, 1999.

5. **JACOBSON, I.; G BOOCH AND J. RUMBAUGH:** *The Unified Software Development Process*, Addison-Wesley, 1999.
6. **ASADA, M. AND P. M. YAN:** *Strengthening Software Quality Assurance* *Hewlett-Packard Journal*, Vol.49, No.2, pp.89(9), May, 1998
7. **ROYCE, W.:** *Software Project Management*, Addison-Wesley Publishing, 1998.
8. **THAYER, R.** (editor): "Software Engineering Project Management," *IEEE Computer Society*, 1997.
9. **CARNEGIE MELLON:** *Software Engineering Institute, The Capability Maturity Model: Guidelines for Improving The Software Process*, Addison-Wesley Publishing Company, 1995.
10. **BOOCH, G; J. RUMBAUGH AND I. JACOBSON:** *The Unified Language User Guide*, Addison Wesley, 1999.

ANEXO 1

Resumen de la aplicación

1. Oración que defina el sistema.
2. Clasificación de la aplicación (intranet, cliente servidor, hipermedia u otra).
3. Facilidades de que consta la aplicación. Cada una se describe mediante una oración.
4. Categorías de los usuarios de la aplicación, definiendo qué facilidades (o partes dentro de las facilidades) de la aplicación pueden utilizar.
5. Interfaz y facilidad de traducción.

Configuración general

- ¿Qué puede hacer el administrador general del sistema?
- ¿Qué pueden hacer en la aplicación los distintos niveles de administradores del sistema? Detalle para cada uno de ellos.
- ¿Cuál URL emplea? Si hay distintos países a tratar, diga todas las URL.
- ¿Se usará conexión ODBC? Indíquelas
- ¿Se usará servidor de correo? Indique dirección IP y SMTP
- ¿Se usará servidor de páginas Web? Indique dirección IP y SMTP.
- ¿Cuál directorio utilizará la aplicación?

Instalación del sistema

- ¿Se utilizará llave de hardware?
 - ¿Qué tipo de protección tendrá por software?
 - ¿Cuáles son los requisitos para instalar el software que se tendrán disponibles? Precise software y hardware.
- Una vez entrado al sistema, especifique las opciones que tendrá disponible la aplicación.

Facilidades y detalle de los requerimientos asociados

Una facilidad se corresponde con una opción general del sistema y se describe por los requerimientos que le dan un mayor nivel de detalle.

La facilidad se describe con una oración simple de carácter general del tipo: Herramienta para ... o Tratamiento de Los requerimientos deben especificar los datos involucrados en cada una de las operaciones, apareciendo operaciones, suboperaciones y por último, los datos involucrados

- <Facilidad 1>
- <Requerimiento 1.1>
- <Requerimiento 1.11>
- <Requerimiento 1.12>
- <Requerimiento 1.2>.

<...>

Glosario de términos de la aplicación