

Comparación de Metodologías para la construcción del modelo de requerimiento de una WebApps. Un caso de Uso.

Lic. Rivero Fabio Nicolas
Lic. en Sistemas de Información
ICTIC – UCP
fnrivero@hotmail.com

Mgter. Vallejos Oscar
Magister en Informática y Computación
ICTIC -UCP
oval_lejos@hotmail.com

Unidad Ejecutora: Miembros del programa AINTA
Facultad: ICTIC – Instituto de Computación Tecnológica de la Información y del Conocimiento
Área: Asignaturas: Ingeniería del Software I, II y III; Programación I, II y III
Laboratorio UCP
Universidad de la Cuenca del Plata
Corrientes – Argentina

Resumen

La ingeniería web es una nueva línea de investigación en ingeniería de software que cubre la definición de los procesos, técnicas y modelos adecuados para el entorno web con el fin de garantizar la calidad de los resultados y la obtención de aplicaciones eficientes y robustas.

La mayoría de las propuestas metodológicas que existen para desarrollar aplicaciones Web orientan al grupo de desarrolladores a través de un conjunto de fases predefinidos que en su mayoría abarcan la etapa de diseño e implementación, descuidando, la etapa de requerimientos.

Este artículo presenta un análisis de un caso de uso aplicando dos metodologías seleccionadas de una lista de veinte y seis, que si bien no son estándares establecidos, son referentes en el ámbito de los desarrolladores de WebApps. Estas metodologías ofrecen modelos, procesos y técnicas para trabajar con sistemas de información web y un buen porcentaje de ellas contempla las etapas a transitar en la ingeniería de requerimiento.

Al no existir una convención en el uso de la aplicación de las mismas, vemos como necesario efectuar un estudio detallado de aquellas que tienen mayor ascendencia en el mercado internacional a los efectos de poder concluir en el mejor comportamiento de cada una de ellas, estableciendo el contexto donde se aprovecha y se hace uso de su mejor performance.

Palabras claves: Ingeniería de requerimientos; análisis y diseño de webApps; performance aplicaciones web.

Contexto

El proyecto promoverá la investigación aplicada, la innovación y el empleo eficiente de las tecnologías emergentes en tecnologías de información en los distintos campos sociales, productivos, culturales y educativos, priorizando, este último, a fin de participar activamente en el desarrollo software.

Para ello se estudiarán diversas metodologías, tecnologías y herramientas de actualidad, referentes al diseño, desarrollo e implementación de sistemas de información construidos y accesibles no solo vía plataforma Web, sino también de otros tipos.

Se aplicaran modelos y métricas para la medición de los distintos aspectos que afectan el proceso de desarrollo y el producto software.

Se prestará especial interés a los sistemas que puedan ser de utilidad para la gestión de las actividades propias de la Institución Educativa. Se enfatizará en aquellas con características de portabilidad a múltiples plataformas.

El proceso de acreditación implica la evaluación (3) externa a la UCP. Una vez acreditado, es financiado por la Universidad de la Cuenca del Plata.

Introducción

Para desarrollar aplicaciones Web demanda muchos conocimientos técnicos y mucha experiencia. En la actualidad el desarrollo de este proceso no está "muy" acompañado de métodos adecuados que garanticen la calidad en las aplicaciones.

Básicamente un método para desarrollar aplicaciones Web está conformado por un modelo de producto (diseño, implementación y testeo) y un modelo de proceso (requerimiento + diseño) que describe las tareas que se deben realizar.

Las metodologías existentes en la actualidad sopesan uno de los modelos sobre el otro y generalmente, referencian más la solución producto y no la forma de obtenerlo.

El propósito de este artículo es comparar un par de metodologías seleccionadas en el contexto de la educación de requisitos (ingeniería de requerimiento) a fin de determinar debilidades y fortalezas.

El resto del artículo está estructurado como sigue. La sección 2 describe el desarrollo de aplicaciones Web y enmarcar las actividades propias del modelo de requerimiento. La sección 3 refiere a la importancia de la aplicación de metodologías para el desarrollo de

aplicaciones. En la sección 4 se presenta el experimento, seleccionando las metodologías a aplicar en el caso de uso y su aplicación.

A modo de corolario, en la sección 5 se presentan las conclusiones y se describen posibles líneas futuras para luego finalizar con la sección 6 donde se referencia la bibliografía utilizada.

Desarrollo de aplicaciones Web e introducción al modelo de requerimiento

La web, se ha convertido en el principal servicio de Internet. Ha dejado de ser un repositorio de páginas estáticas para convertirse en un servicio que permite acceder a multitud de prestaciones y funciones, así como a infinidad de servicios, programas, etc.

Este éxito se en el protocolo HTTP y el lenguaje HTML. Uno permite una implementación simple y sencilla de un sistema de comunicaciones que nos permite enviar cualquier tipo de ficheros de una forma fácil, simplificando el funcionamiento del servidor y permitiendo que servidores poco potentes atiendan miles de peticiones y reduzcan los costes de despliegue. El otro proporciona un mecanismo de composición de páginas enlazadas simple, fácil, eficiente y de uso muy simple.

Inicialmente la web era una colección de páginas estáticas que podían consultarse o descargarse.

Luego se le incluyo un método para confeccionar páginas dinámicas que permitieron que lo que se mostraba fuese dinámico (CGI -*common Gateway interface*) y definía un mecanismo mediante el cual podíamos pasar información entre el servidor HTTP y programas externos. Y con la libertad a la hora de escoger el lenguaje de programación para desarrollarlos. Ante la deficiencia de los CGI (carga de maquina: cada vez que se recibe una petición, el servidor web lanza un proceso que ejecuta el programa CGI), se empiezan a desarrollar alternativas a los CGI.

Por un lado se diseñan sistemas de ejecución de módulos más integrados con el servidor, que evitan que éste tenga que instanciar y ejecutar multitud de programas. La otra vía consiste en dotar al servidor de un intérprete de algún lenguaje de programación (RXML, PHP, VBScript, etc.) que nos permita incluir las páginas en el código de manera que el servidor sea quien lo ejecute, reduciendo así el tiempo de respuesta.

A partir de allí se incrementa el número de arquitecturas y lenguajes de programación que nos permiten desarrollar aplicaciones web.

Una de las mayores deficiencias en la práctica de construcción de software es la poca atención que se presta a la discusión del problema. En general los desarrolladores se centran en la solución dejando el problema inexplorado. El problema a resolver debe ser deducido a partir de su solución [14].

De acuerdo a estudios realizados se observó que los proyectos han fallado porque sus *requerimientos no tuvieron una correcta descripción* y fueron inadecuadamente explorados lo cual derivó al incremento de tiempos y costos iniciales del proyecto.

Los *requerimientos* se ubican en el *dominio de la aplicación* donde está el problema. Se debe definir el problema mediante una precisa y explícita descripción.

Las dificultad que se plantean al obtener los requerimientos de un sistema de Información (SI), fundamentalmente radican en la adquisición del conocimiento del usuario y los problemas que se presentan en la captura de los datos. Normalmente, los requerimientos tienen solamente un sentido en el contexto organizacional y generalmente son el producto de la interacción usuario – técnico.

Existe una falta de consenso entre los usuarios, la disponibilidad de tiempo de parte de los usuarios y los problemas marcado de la falta de comunicación entre usuarios y desarrolladores (lenguaje técnico, diferente formación profesional, etc.).

Los beneficios de los buenos requerimientos incluyen acuerdos entre los stakeholders de la tarea a ejecutar y del criterio de aceptación del sistema del sistema concluido. Se puede llegar a la obtención de una base para estimaciones de recursos y tiempo propendiendo a las mejoras en la usabilidad y mantenibilidad.

Como corolario podemos coincidir que un Requerimiento es una condición o capacidad que necesita un usuario para resolver un problema o alcanzar un objetivo, o una condición o capacidad que se debe cumplir o poseer un sistema para satisfacer un contrato, norma, especificación, u otros documentos formalmente impuestos [1].

Por otra parte, se define a la Ingeniería de Requerimientos como el proceso sistemático de desarrollar requerimientos a través de un proceso cooperativo e interactivo de analizar el problema, documentar las observaciones resultantes en una variedad de formatos de representación, y chequear la precisión de la comprensión obtenida [2].

Existe consenso respecto a que la Ingeniería de Requerimientos es un proceso iterativo y tiene fases que pueden ejecutarse en forma concurrente, sin embargo hay distintos enfoques de los subprocesos a realizar.

Por ejemplo, Davis [3] propone que la fase de requerimientos consta de dos actividades principales: Análisis del Problema (delinear refinar y negociar las restricciones) para comprender el problema y expandir la información y la Descripción del Problema que incluye el chequear y consolidar la información.

Otro enfoque es el de Jitnah [4], quien considera. que el proceso de Ingeniería de Requerimientos consta de las actividades de elicitación, análisis y especificación de requerimientos, siendo estas tareas no necesariamente ejecutadas respetando algún orden o periodo de tiempo. En [5] Kotonya y Sommerville sugieren un proceso iterativo que se puede llevar a cabo durante todo el proceso de desarrollo del software. Este proceso establece cuatro actividades principales que son: elicitación (donde se captura, descubre y adquiere el conocimiento), análisis, especificación y validación de requerimientos. Además establece una actividad de Gestión de Requerimientos para administrar los cambios, el mantenimiento y la trazabilidad de los requerimientos.

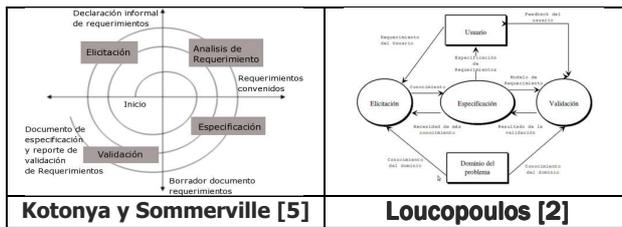


Fig. 1: Modelo del Proceso de Ingeniería de Requerimientos

Por otro lado Van Lamsweerde [6] asegura que las actividades realizadas en el proceso de la Ingeniería de Requerimientos son análisis del dominio, elicitación, negociación y acuerdo, especificación, análisis de la especificación, documentación, y por último la evolución.

Una de las propuestas más destacadas y de mayor consenso es la presentada por Loucopoulos [2]. Básicamente, Loucopoulos propone comprender y describir el problema alcanzando un acuerdo sobre la naturaleza del problema (elicitación, especificación y validación). Ver figura 1.

Hay muchos retos para satisfacer estos criterios mencionados. Uno de los principales problemas que existe es la brecha de comunicación entre el cliente y los desarrolladores [¡Error! No se encuentra el origen de la referencia.].

Importancia del uso de una metodología

Las metodologías son herramientas de soporte garantizando el proceso de desarrollo establecido en pasos con la obtención concreta de los resultados esperados.

Además son una técnica para la construcción, el análisis y la transformación de los SI. Todas ellas ofrecen una sintaxis de los documentos, tipos de diagramas y una semántica preestablecida.

Metodologías existentes

Según varios estudios comparativos realizados [10], [11], [12], [13] sobre las técnicas actualmente disponibles se pueden concluir que la mayoría está centrada en el análisis y el diseño.

Las técnicas presentadas por cada metodología cabe aclarar que si bien prestan bastante atención a la captura y definición de requerimientos prestan poca importancia a la validación de requerimientos.

Detalle del caso de uso

Descripción de las metodologías utilizadas en el experimento

De los estudios comparativos presentados anteriormente podemos seleccionar 2 propuestas para aplicar a un caso de uso específico. Las metodologías seleccionadas son UWE y NDT. El motivo de la selección de las mismas es porque brindan y completan una gran variedad de técnicas en las etapas de captura, definición y validación.

UWE: UML-Based Web Engineering

UML-Based Web Engineering (UWE) es una propuesta metodológica basada en el Proceso Unificado (Jacobson, Booch & Rumbaugh, 1999) y UML para el desarrollo de aplicaciones web (Hennicker & Koch, 2000, Koch, 2001). UWE cubre todo el ciclo de vida de este tipo de aplicaciones, centrandose además su

atención en aplicaciones personalizadas (adaptivas). Para este trabajo, nos interesa principalmente analizar la propuesta de captura de requisitos de UWE. Esta metodología distingue entre la tarea de elicitar requisitos, definir y validar los requisitos. El resultado final de la captura de requisitos en UWE es un modelo de casos de uso acompañado de documentación que describe los usuarios del sistema, las reglas de adaptación, los casos de uso y la interfaz.

UWE clasifica los Requerimientos en dos grandes grupos: funcionales y no funcionales.

Los funcionales son los relacionados con el contenido, la estructura, la presentación, la adaptación y los usuarios.

Además, UWE propone como técnicas apropiadas para la captura de los requisitos de un sistema web las entrevistas, los cuestionarios y los checklists y los casos de uso, los escenarios y el glosario para la definición de requisitos. Para la validación propone walk-throughs, auditorías y prototipos.

NDT - Navigational Development Techniques

NDT (Navigational Development Techniques) (Escalona, Torres & Mejías, 2002) es una técnica para especificar, analizar y diseñar el aspecto de la navegación en aplicaciones web. Para este trabajo, solo es relevante la propuesta que ofrece para la definición y captura de requisitos. El flujo de especificación de requisitos de NDT comienza con la fase de captura de requisitos y estudio del entorno. Para ello, plantea el uso de técnicas como las entrevistas o el brainstorming y JAD. Tras esta fase, se propone la definición de los objetivos del sistema. En base a estos objetivos, el proceso continúa definiendo los requisitos que el sistema debe cumplir para cubrir los objetivos marcados. NDT clasifica los requisitos en:

- Requerimientos de almacenamiento de información
- Requerimientos de almacenamiento de información de actores
- Requerimientos de almacenamiento de información funcionales
- Requerimientos de interacción, representados mediante:
 - Frases, que recogen cómo se va a recuperar la información del sistema utilizando un lenguaje especial denominado BNL (Bounded Natural Language) (Brisaboa, Penabad, Places & Rodríguez, 2001).
 - Prototipos de visualización, que representan la navegación del sistema, la visualización de los datos y la interacción con el usuario.
- Requerimientos no funcionales

Todo el proceso de definición y captura de requisitos y objetivos que propone NDT se basa principalmente en plantillas o patrones, pero también hace uso de otras técnicas de definición de requisitos como son los casos de uso y los glosarios. La propuesta ofrece una plantilla para cada tipo de requisito, lo que permite describir los requisitos y objetivos de una forma estructurada y detallada. Algunos de los campos de los patrones son cerrados, es decir, solo pueden tomar valores

predeterminados. Estos campos permiten que en el resto del proceso del ciclo de vida de NDT se puedan conseguir resultados de forma sistemática.

Dominio de Aplicación

Dominio: El dominio de la aplicación es un Sistema de Gestión de Trámites de Financiación de Deuda. Los actores que intervienen: a) Asesor Comercial: Es el responsable del primer contacto con el cliente y de la carga del trámite de Financiación de deuda; b) Responsable Backoffice Financiación: es el responsable de la distribución de los trámites de su sector; c) Asesor BackOffice Financiación: es el responsable de la gestión del trámite de financiación.

El asesor comercial ante una llamada del cliente solicitando un plan de pago por facturas adeudadas, carga el trámite de financiación en el sistema especificando, entre otros: número de teléfono, nombre del cliente, número de cliente, domicilio de facturación, deuda Financiada, anticipo, cantidad de facturas, cantidad de cuotas, envió de comprobante a domicilio.

El responsable de Financiación lista los trámites de financiación por antigüedad ingresados a su sector y los distribuye a los asesores disponibles.

Los tipos de trámites que el asesor puede ingresar son Findeuda, FindeudaAnticipo, FinDeudaEspecial.

Tipos de Segmento de clientes (Residencial, empresas, grandes clientes)

El asesor de backoffice gestiona los trámites de financiación que le fueron asignados. Ingresando un plan de financiación a la línea del cliente y cerrando el trámite o cerrando el trámite de no poderse realizar la gestión de financiación de deuda

El trámite cuando es cargado por el asesor comercial está en estado registrado, cuando el asesor de backoffice lo toma pasa a estado pendiente, una vez que lo trabaja para a estado cerrado.

Definición de actores: Todos los actores poseen un usuario y un perfil.

Definición de Procesos: El Asesor comercial realizara la consulta de deuda del cliente para informarle la deuda pendiente. En base a la deuda pendiente le especificara las pautas de financiación vigentes en la compañía.

Si el cliente es del segmento residencial el asesor comercial debe ingresar un tramite findeudaanticipo, si es empresas o grandes clientes un tramite findeuda. El tramite FinDeudaEspecial debe ingresarlo cuando el cliente solicita un plan fuera de pautas corporativas

Pautas tramites findeudaAnticipo:

Tipo de Segmento: Residencial

Anticipo: 25% de la deuda

Cantidad de Cuotas: 6

Bloqueos: BPS (Bloqueo de salida parcial)

Pautas tramites findeuda

Tipo de Segmento: Empresas o grandes clientes

Anticipo: 50% de la deuda

Cantidad de Cuotas: 10

Bloqueos: BPS (Bloqueo de salida parcial) o BCC (Bloqueo a Celulares)

Pautas tramites findeudaEspecial

Tipo de Segmento: Todos

Anticipo: 50% de la deuda

Cantidad de Cuotas: más de 10

Bloqueos: BPS (Bloqueo de salida parcial) o BCC (Bloqueo a Celulares) o sin bloqueo de línea

El asesor de backoffice en base al tipo de trámite revisa las pautas de financiación y realiza la carga o no del plan de financiación en el sistema y cierra el trámite con su respectivo comentario.

Resultados obtenidos

Aplicación de NDT al dominio del problema: Se obtuvieron los Documentos de requerimientos tal cual lo propone NDT (participantes, Definición de Objetivos, identificar Requerimientos de Almacenamiento de Información, Identificar y definir los actores y sus relaciones, Identificar y definir los requerimientos funcionales, Requerimientos de interacción: Definición de Frases y Prototipos de visualización).

Aplicación de UWE al dominio del problema: Se obtuvieron los Documentos de requerimientos tal cual lo propone UWE (Definición de Casos de Uso con estereotipos, Diagrama de Actividades con estereotipos).

Conclusiones

UWE propone una fase de definición de requisitos totalmente grafica a través de Casos de Uso, Diagramas de Actividades y Metamodelos. Por el contrario NDT propone una representación más textual partiendo de objetivos y llegando a patrones de visualización.

Las herramientas que proponen ambos métodos, UWE con Argo UWE que generalmente trabaja como Plugin de otra herramienta por ejemplo el Magic Draw es muy completo a nivel de diseño, la herramienta que propone NDT (NDT-Tool) es más aproximado para la realización de requerimientos.

NDT es una herramienta apropiada para la comunicación Cliente – Equipo de Proyecto, en cambio UWE es más apropiada para el equipo del proyecto.

Ambos métodos son flexibles y adaptables para ser utilizados en diferentes dominios de problemas. Los dos presentan una estructura de proceso bien definida que facilita el proceso de desarrollo para la construcción del modelo de requerimiento de la aplicación web. Es un modelo claro y explícito que indica las características del producto a desarrollar y la calidad de la aplicación que se producirá.

Línea futura

Con la comparación de los dos métodos seleccionados se cubrió los objetivos de la presente investigación: enmarcar deficiencias y fortalezas de ambos métodos en la etapa de educación de requisitos.

Como trabajo futuro nos planteamos la necesidad de evaluar las distintas metodologías en las etapas de diseño y testeo y concretar con una propuesta de adopción de una de ellas.

Líneas de investigación y desarrollo

- Ingeniería de construcción de aplicaciones informáticas y aquellas orientadas, específicamente, a plataformas web.

- Ingeniería de construcción de aplicaciones en tiempo real.
- Minería de datos aplicada en la inteligencia de negocios.
- Metodologías para evaluar la calidad de productos software.
- Promover actividades de I+D relacionados con las tecnologías emergentes.
- Investigar y comparar modelos de sistemas inteligentes.

Resultados y Objetivos

En estos primeros meses del proyecto (8): a) se investigo sobre las metodologías a utilizar; b) se elaboro la pagina web del grupo de investigación; c) se elaboro el presente artículo (comparación de metodologías de IR); d) se está realizando un estudio sobre la aceptación y uso del espacio virtual por parte de la comunidad educativa; y un estudio sobre la inserción de los egresados de la UCP en área laboral de la región. En ambas líneas se trabajara con minería de datos y de donde se pretende la obtención de otro artículo; e) se está preparando un seminario de minería de datos con Weka cuyo destinatario serán alumnos avanzados de la carrera y; otro seminario de desarrollo de aplicaciones Web dinámicas.

Recurso Humano

Integrantes del Proyecto: Director: Mgter, Vallejos, Oscar A.; Investigadores: Lic. Soler, Alejandro y Lic. Ayala, Raúl; personal de apoyo: Lic. Rivero, Fabio. Debido a los pocos meses de implementación del proyecto (8) no se ha realizado ninguna tesis de posgrado ni de grado que se relacionan directamente con la línea de I/D presentada, si bien esta previsto dentro de los objetivos del mismo.

Referencias

- 1- The Institute of Electrical and Electronics Engineers, "IEEE Standard Glossary of Software Engineering Terminology", New York, Std.610.12-1990, 1990
- 1- Loucopoulos P. Karakostas V., "System Requirements Engineering", 1995, London
- 3- Davis Alan M., "Software Requirements, Objetos, Funciones y Estados", Editorial Prentice Hall, ISBN 0-13-805763-X, 1993.
- 4- Jitnah, D., Han, J., Steele, P., "Software Requirements Engineering: An Overview", Technical Report 95-04, Peninsula School of Computing and Information Technology, Monash University, Melbourne, Australia, September (1995).
- 5- Katonya, G., Sommerville, "Requirements Engineering Processes and Techniques", Chichester, John Wiley & Sons (1998)
- 6- Van Lamsweerde, A., "Requirements engineering in the year 00: a research perspective", International Conference on Software Engineering archive, Proceedings of the 22nd international conference on Software engineering, Limerick, Ireland, ACM Press, ISBN:1-58113-206-9, (2000), 5 - 19
- 7- Borland®, "Mitigating Risk With Effective Requirements Engineering", Abril 2005
- 8- Angela Tuffley, "CIT3190 IT Project Course. Requirements Elicitation & Management", 2005
- 9- M. Stephens, P. Bates, "Controlling Prototyping and Evolutionary Development", Proceedings 4th International Workshop on Rapid System Prototyping: Shortening the Path From Specification To Prototype - Junio 1993
- 10- Barry, C. & Lang, M. (2001) A Survey of Multimedia and Web Development Techniques and Methodology Usage. IEEE Multimedia. April-June 2001, 52-60.
- 11- Escalona, M.J., Mejías, M., Torres, J. (2002). Methodologies to develop Web Information Systems and Comparative Analysis. Informatik/Informatique. núm. 2/2002 de I/I.
- 12- Escalona, M.J., Torres, J., Mejías, M. (2002). Requirements Capture Workflow in Global Information Systems. Proceedings of OOIS. Springer-Verlag. Montpellier, France.
- 13- Koch, N. (2001). Software Engineering for Adaptive Hypermedia Applications. Ph. Thesis, FAST Reihe Softwaretechnik Vol(12), Uni-Druck, Munich, Germany.
- 14- Jackson, Michael, Requerimientos de Software y Su Especificacion, Addison- Wesley, ACM Press, 1995.
- 15- [IEEE, 1993] [IEEE 1993] IEEE. IEEE La Práctica recomendada para las Especificaciones de Requerimientos de Software. IEEE/ANSI Standard 830-1993, El instituto de Eléctrico e Ingenieros de la Electrónica, 1993.
- 16- [POH, 1997] K. Pohl. La Ingeniería de requerimientos: Una Apreciación global. La enciclopedia de Informática y Tecnología, 36, 1997. Disponible en <http://sunsite.informatik.rwth-aachen.de/CREWS/reports96.htm>.