

Mantenimiento Perfectivo en un Gestor de Contenidos: Una Experiencia para la Especificación de Consultas Dinámicas

Ana Gomez Codutti, Sonia I. Mariño, Pedro Alfonzo, Maria V. Godoy

Departamento de Informática, Facultad de Ciencias Exactas y Naturales y Agrimensura
Universidad Nacional del Nordeste. Corrientes. Argentina
simarinio@yahoo.com

Resumen—Los Sistemas Gestores de Contenidos o CMS son ampliamente aplicados en una diversidad de soluciones para mediar la comunicación con los usuarios finales. En este trabajo se expone una experiencia de personalización de un CMS de libre distribución considerada ésta como un mantenimiento perfectivo, abordando las actividades que propone estándar IEEE 1219 para gestionar dicho proceso.

Palabras clave—Ciclo de vida, gestión del mantenimiento, metodologías, gestores de contenidos.

I. INTRODUCCIÓN

Una de las nueve disciplinas de la Ciencia Informática constituye la Ingeniería del Software (IS) [1]. Se coincide con [2] quien expone que la IS es una disciplina de la ingeniería que abarca todos los aspectos de la producción de software desde las etapas preliminares de la especificación del sistema, hasta el mantenimiento de éste después de que se lo implementa.

El conjunto de procesos, actividades y tareas intervinientes en el desarrollo, explotación y mantenimiento de un producto software, desde su etapa inicial de formulación de requerimientos hasta que su utilización finaliza, se denomina ciclo de vida del software [3]. De todas estas actividades en las que se descompone el ciclo de vida del software, las últimas en realizarse, pero no menos importantes son las tareas de mantenimiento [4].

Según [5], el mantenimiento del software (MS) es impulsado por el cambio asociado a la corrección de errores, a la adaptación del software a un nuevo entorno, a las solicitudes de nuevas características o funciones por parte del cliente y a las mejoras que se deban introducir en una aplicación para ofrecer beneficio en un contexto moderno. Asimismo, [2] considera al mantenimiento como “el proceso general de cambiar un sistema después de que éste ha sido entregado”. Esta última definición señala que dichas actividades comienzan después que el producto se encuentra en funcionamiento. Sin embargo, usualmente, algunas pueden comenzar antes de la entrega del producto.

De este modo, el proceso de desarrollo de software continúa durante su tiempo de vida, aun cuando es entregado. Esto se debe a que los sistemas construidos experimentan cambios para su permanencia y utilidad. Una vez que el software comienza a emplearse, surgen nuevos requerimientos y los existentes cambian. Algunas partes del mismo tienen que modificarse para corregir errores detectados en su funcionamiento, adaptarlo a una nueva plataforma, mejorar su rendimiento, entre otras características no funcionales [2].

El proceso de mantenimiento, proporciona las acciones necesarias y detalladas de las entradas y salidas a esas actividades, las que se describen en las normas de mantenimiento de software IEEE 1219 e ISO/IEC 14764 ([6]).

IEEE STD 1219 [7] define cambios en un producto software a través de un proceso dividido en fases. Éste es iterativo y en cascada, con una gran semejanza al ciclo de vida del desarrollo clásico. Las fases se detallan a lo largo del estándar, indicando en cada una los elementos de los que se dispone al empezar, por ejemplo la documentación, las tareas a realizar, la manera de seguir fielmente los estándares, y por último el resultado de la misma (documentación generada, código, entre otras). Se distinguen las siguientes: i) identificación y clasificación del problema o de la modificación; ii) análisis, iii) diseño, iv) implementación, v) pruebas del sistema, vi) pruebas de aceptación, vii) liberación del producto.

Continuando con lo señalado anteriormente [2] distingue tres tipos de mantenimiento: i) Mantenimiento correctivo: realizar cambios en el software para reparar defectos; ii) Mantenimiento adaptativo: modificar el software para adaptarlo a un nuevo entorno o nuevos requerimientos; iii) Mantenimiento perfectivo: perfeccionar el software implementando nuevos requerimientos o bien, mantener la funcionalidad del sistema pero mejorando su estructura y su rendimiento. Por otra parte, [8] considera otro tipo de mantenimiento, definido cómo preventivo, el cual consiste en modificar un producto de software después de entregado para detectar y corregir fallas latentes antes de que se conviertan en fallas funcionales.

Los Sistemas Gestores de Contenidos o CMS son ampliamente aplicados en una diversidad de soluciones para mediar la comunicación con los usuarios finales. Existen numerosas herramientas comprendidas en esta categoría del software, de libre distribución y propietarias, así como diversas experiencias exitosas de su uso para la gestión y diseminación de contenidos en distintos dominios del conocimiento.

Por otra parte, cabe destacar que el mantenimiento careció del mismo grado de atención, en comparación con otras fases del proceso como es el desarrollo, a pesar de ser parte integral del ciclo de vida [6]. Desde la Ingeniería del Software y particularmente desde el mantenimiento como parte integral del ciclo de vida es relevante conservar al software operativo el mayor tiempo posible, maximizando las inversiones de las organizaciones. Sin embargo, no todos los CMS como soluciones configurables son adecuadas a los usuarios finales.

En este trabajo se expone una experiencia de personalización de un CMS de libre distribución, a los efectos de implementar mejoras en su rendimiento, considerada ésta como un mantenimiento perfecto, abordando las actividades que propone la IEEE 1219 para gestionar dicho proceso.

II. METODOLOGÍA DE TRABAJO

La metodología aplicada constó de las siguientes fases:

- Revisión de bibliografía sobre experiencias concretas y antecedentes de mantenimiento del software.
- Recopilación, selección, estudio de metodologías orientadas al proceso de mantenimiento. Se optó por abordar las actividades mencionadas por el estándar IEEE 1219, dado que este explicita “qué” hacer, sin dar cuenta del “cómo”.
- Selección de un CMS de libre distribución. Se optó por Drupal dada su amplia difusión en el mercado.
- Selección de un módulo y su adecuación a fin de ilustrar la aplicación del mantenimiento perfecto.

En la Tabla 1, se exponen las fases explicitadas por el estándar seleccionado, se describe una visión general del proceso de mantenimiento y se enuncian las actividades implementadas y asociadas a cada una de ellas.

TABLA I. FASES Y ACTIVIDADES RELACIONADAS CON ESTÁNDAR IEEE 1219

Fases	Actividades
Identificación y clasificación del problema o de la modificación.	<ul style="list-style-type: none"> ▪ Identificar el problema. Se planteó la necesidad de diseñar interfaces que faciliten el despliegue de información específica. ▪ Clasificar el problema por tipo de mantenimiento. Se definió como tipo el mantenimiento perfecto ▪ Asignar prioridad. Se estableció como primacía la redefinición de la interfaz inicial y la localización de información específica. ▪ Obtener aprobación de la solicitud de modificación y las tareas a llevar a cabo. Se analizó la solicitud y se procedió a su aceptación. ▪ Estimar inicialmente los recursos necesarios para modificar el sistema existente. Básicamente se asociaron a la disponibilidad de una computadora personal y conexión a la web para su despliegue en la mencionada interfaz.
Análisis.	<p>Se identificaron los elementos a modificar. A través de prototipos se plantearon soluciones alternativas y como estos cambios impactan en la información contenida en la base de datos y en el funcionamiento del sistema.</p> <ul style="list-style-type: none"> ▪ Evaluar el impacto ▪ Evaluar los costos ▪ Estudiar la viabilidad y el alcance de las modificaciones ▪ Desarrollar un plan preliminar de diseño, implementación, pruebas y liberación del software. ▪ Desarrollar estrategias de pruebas
Diseño	Se diseñaron los elementos de interfaces que facilitan la selección de información contenida en la base de datos. A fin de validar las modificaciones realizadas se generaron y ejecutaron las pruebas previamente planteadas. Además, los cambios realizados se documentaron para su futura divulgación y constituirse en información de retroalimentación.
Implementación	<ul style="list-style-type: none"> ▪ Desarrollar y probar las modificaciones realizadas ▪ Codificar y generar pruebas unitarias. ▪ Integrar el software modificado con el sistema existente. ▪ Analizar el riesgo. ▪ Revisar la preparación para las pruebas.
Pruebas del Sistema	<ul style="list-style-type: none"> ▪ Realizar pruebas sobre el sistema modificado ▪ Revisar integridad. ▪ Obtener aprobación.

Pruebas de Aceptación	<ul style="list-style-type: none"> ▪ Realizar pruebas sobre el sistema completamente integrado ▪ Estas pruebas se han desarrollado incrementalmente y en paralelo al mantenimiento perfecto aplicado en el módulo seleccionado.
Liberación del Producto	<ul style="list-style-type: none"> ▪ Desarrollar un plan. ▪ Notificar a los usuarios. ▪ Realizar una copia de seguridad de la versión del sistema. ▪ Realizar la instalación y capacitar a los usuarios. ▪ Actualmente se elabora un plan destinado a la liberación del producto. En referencia a su difusión, se entiende este trabajo como un producto derivado de actividades de I+D y se planifica un curso de capacitación en la temática.

A continuación se presenta como ejemplo ilustrativo la aplicación de mantenimiento perfecto al diseño estándar propuesto por Drupal a fin de mejorar las funcionalidades del sitio. Particularmente se mostrarán las modificaciones efectuadas respecto a la localización de contenidos.

En la Fig. 1, se observa el formulario de búsqueda estándar. A partir de éste, se consideró necesario la incorporación de determinados criterios de filtrado que facilitarían y agilizarán la búsqueda de contenidos a los usuarios. Para ello, se creó una “Vista”, que constituye un constructor de consultas [9] a través de la cual se despliegan los informes que permite mostrar datos, seleccionando en la configuración su origen, campos que se visualizarán, criterios de filtrado y de ordenación, entre otros elementos.

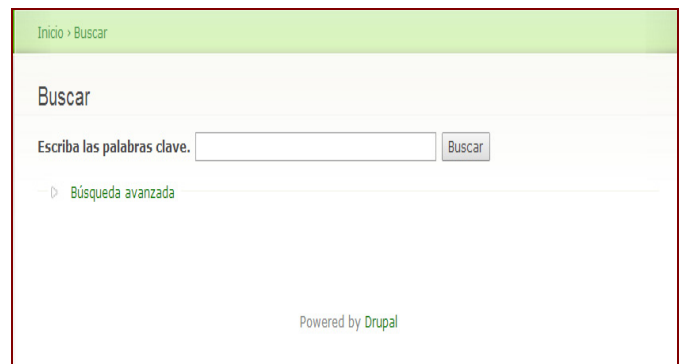


Fig. 1. Formulario de búsqueda estándar

En la Fig. 2 se visualiza la configuración correspondiente al nuevo formulario de búsqueda.

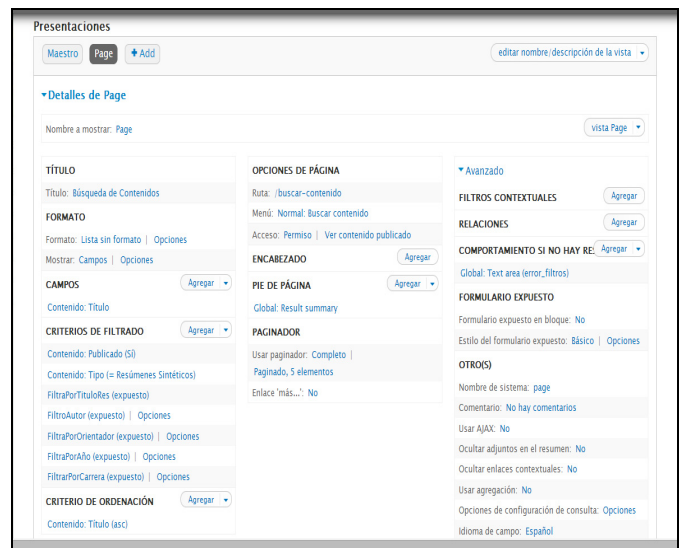


Fig. 2. Configuración del nuevo formulario de búsqueda

De este modo, en la Fig. 3, se observa el nuevo formulario obtenido.

Fig.3. Formulario de búsqueda personalizada

III. CONCLUSIÓN Y TRABAJOS FUTUROS

El crecimiento de la Industria del Software y la connotación asociada a la generación de aplicaciones web ha favorecido la evolución de tecnologías y herramientas para la difusión de la información. El mantenimiento del software se considera como una solución a la implementación de procesos más rápidos, flexibles y eficientes para el desarrollo de software.

La ilustración de la implementación de la propuesta metodológica expuesta en el estándar IEEE 1219 para lograr un mantenimiento perfecto en un módulo de la plataforma Drupal, ha permitido:

- proponer un proceso de mantenimiento ágil y perfecto, diseñando y desarrollando versiones frecuentes hasta lograr la aceptación del Project Manager.
- mejorar el conocimiento y comprensión del código del mencionado CMS, además colaborar y comunicar con la comunidad de desarrolladores que operan como un equipo de mantenimiento.
- flexibilizar los requerimientos de modificación, con la incorporación de nuevas solicitudes y la asignación de nuevas prioridades.
- disminuir los costos vinculados con el mantenimiento perfecto.

A partir de lo expuesto, se propone profundizar en esta línea de I+D+I, brindando y difundiendo datos empíricos que sustenten experiencias relacionadas con el sector de Servicios y Sistemas Informáticos, y contribuir con el mantenimiento perfecto de herramientas software disponibles para todos los usuarios.

REFERENCIAS

[1] REDUNCI. Carreras de Grado en Informática. Propuesta de Currícula, 2006. [En Línea]. Disponible en: <http://redunci.info.unlp.edu.ar/docs/propuesta.doc>. [Accedido: 2-jul-2014].

[2] I. Sommerville, “Ingeniería del Software”.7ª Edición. Ed. Pearson, 2005.

[3] ISO/IEC 12207:1995. Information Technology - Software life cycle processes.

[4] O. G. Reyes Pupo, “Mantenimiento del Sistema para la Tramitación de Interrupciones en los Sistemas de Informática y Comunicaciones (TI)”. Ciencias Holguín, Revista Trimestral, enero-marzo 2010. [En Línea]. Disponible en:

<http://www.ciencias.holguin.cu/index.php/cienciasholguin/articulo/view/542/414>. [Accedido: 10-jul-2014].

[5] R. Pressman, “Ingeniería del software. Un enfoque práctico”. 7ª Edición. Mexico: Ed. Mc Graw-Hill Interamericana, 2010.

[6] Swebok. “Guide to the Software Engineering Body of Knowledge”, 2004. [En línea]. Disponible en: <http://www.computer.org/portal/web/swebok>. [Accedido: 10-jul-2014].

[7] IEEE STD 1219:1993. Standard for Software Maintenance. IEEE Computer Society Press. USA, 1993.

[8] ISO/IEC Standard 14764: 2006. International Organisation for Standardization. Software Engineering – Software Life Cycle Processes - Maintenance.

[9] P. Cambra, “Módulo Views: Qué es, cómo se instala y cómo crear nuestra primera vista”, 2014. [En línea]. Disponible en: <http://cambrico.net/drupal/modulo-views-que-es-como-se-instala-y-como-crear-nuestra-primer-a- vista>. [Accedido: 15-jul-2014].



Ana Gomez Codutti. Alumna avanzada de la carrera Licenciatura en Sistemas de Información. Becaria EVC-CIN, Universidad Nacional del Nordeste.



Sonia I. Mariño. Docente–Investigadora, Profesora Titular, Dedicación Exclusiva, del Departamento de Informática de la Facultad de Ciencias Exactas de la Universidad Nacional del Nordeste. Licenciada en Sistemas. Es Magíster en Informática y Computación. (UNNE - Universidad de Cantabria - España), Magíster en Epistemología y Metodología de la Investigación Científica (Facultad de Humanidades - UNNE). Cursa el Doctorado en “Ciencias Cognitivas”, Facultad de Humanidades (UNNE).



Pedro L. Alfonzo. Docente-Investigador. Experto en Estadística y Computación (Facultad de Ciencias Exactas y Naturales y Agrimensura – UNNE). Especialista en Ingeniería de Software (Universidad Nacional de la Plata). Magíster en Ingeniería de Software (Universidad Nacional de La Plata).



María V. Godoy. Docente–Investigadora, Profesora Titular, Dedicación Exclusiva, del Departamento de Informática de la Facultad de Ciencias Exactas de la Universidad Nacional del Nordeste. Experta en Estadística y Computación, y Licenciada en Sistemas. Es Magíster en Informática y Computación. (UNNE - Universidad de Cantabria - España). Cursa el Doctorado en “Ciencias Cognitivas”, Facultad de Humanidades (UNNE).