

CONTROLA: HERRAMIENTA DE APOYO AL PROCESO DE DESARROLLO DE SOFTWARE EN LAS PEQUEÑAS COMPAÑÍAS

*Clayton Vieira Fraga Filho {claytonfraga@gmail.com}; José Maurício dos Reis {mauricio1968@gmail.com}
Faculdade de Viçosa - FDV
Viçosa-MG - Brasil*

RESUMEN

El presente trabajo tiene como propósito presentar CONTROLA, una herramienta de apoyo al proceso de desarrollo de Software para las pequeñas compañías, con enfoque en la administración de requisitos. Permite la identificación de los requisitos cerca del *stakeholder* (cliente/usuario del sistema), sus detalles y la administración de los cambios a través de rastreabilidad y control de versiones. La herramienta esta siendo usada por los usuarios a lo largo de Brasil, entre ellos los estudiantes de graduación y postgrado, los maestros y pequeñas compañías de desarrollo del software.

PALABRAS-CLAVE

Ingeniería de Software; Procesos de Desarrollo de Software; Gerencia de Requisitos; Gerencia de Cambios; Controla.

INTRODUCCIÓN

Con la globalización, los mercados se pusieron más competitivos y la demanda para los productos y servicios de buena calidad aparecen como el factor decisivo en el momento de escoger un compañero de negocios por los días de hoy. En el ámbito del desarrollo de Software no es diferente, el cliente, cada vez más exigente, demanda por un sistema dentro del la marca pedida y que atienda sus necesidades, desarrollado en el tiempo más pequeño posible.

Actualmente, las compañías no sólo buscan poner en el mercado los productos que atiendan la demanda de sus clientes, sino que también activar su producción. Esa optimización tiene como punto principal el proceso de producción inherente a la compañía.

En la industria, en general, la calidad de un producto depende directamente de los procesos, materiales y técnicas usadas, por consiguiente, es importante que las compañías desarrolladoras de Software tengan en enfoque que la definición de un proceso de desarrollo y el uso de herramientas y técnicas apropiadas que garantizan un mejor funcionamiento y calidad del producto final y, por consiguiente, la mayor satisfacción del cliente.[KAN 2002].

La construcción e implantación de un proceso de desarrollo de Software en las pequeñas compañías desarrolladoras no es algo común, porque el enfoque normalmente está en el desarrollo por la asistencia inmediata del problema del cliente. A los requisitos están asociados los problemas principales del desarrollo del software. [BLASCHEK 2002], siendo así, la administración de los requisitos y de los cambios a lo largo del proyecto se vuelve un proceso bastante costoso, dado que no hay documentación formal y

estandarizado de las necesidades del cliente. Ese cuadro puede empeorarse por la gran rotación de personal (*turnover*). Frecuentemente encontramos compañías con problemas serios en atender a sus clientes en el periodo indicado y cumpliendo con el propósito del producto, porque en el transcurso del proyecto uno de los miembros del equipo deja la compañía, llevando consigo el conocimiento de las reglas comerciales y de la especificación del Software. El tiempo y los costes para la recuperación de ese conocimiento pueden ser muy altos dependiendo de la fase en que el proyecto se encuentra.

Además de los asuntos arriba expresados, la falta de registro formal y estandarizado de requisitos afecta el proceso de mantenimiento adaptativo o evolutivo del Software, impidiendo la identificación de los impactos del cambio de algún requisito en el desarrollo del sistema, aumentando la ocurrencia de errores.

OBJETIVO

Pretendemos en este trabajo, presentar una herramienta para que las compañías tengan un referencial inicial que puede ser adaptado a su realidad y lo bastante robusto para proporcionar mayor calidad y reducir el descontento del cliente por atrasos y errores de especificaciones. El Software denominado Controla presenta un grupo de indicadores que hacen posible a un equipo de proyecto tomar decisiones junto al cliente para que el proyecto sea conducido con mayor control, reduciendo focos de problemas. Se llevaron a cabo dos modelos de estimativa, uno para la priorización de los requisitos junto a los clientes del sistema [WIEGERS 2003] y otro ideal para el dimensionar esfuerzo y tamaño de proyectos basado en Casos de Uso. Como objetivos específicos:

- 1) Hacer posible la definición de un Plan de Proyecto, con informaciones importantes, entre ellas: metas gerenciales, restricciones, equipo del proyecto y expectativas de los usuarios finales;
- 2) Permitir la administración de requisitos, desde su definición hasta el control de versiones, control de cambios, rastreabilidad con los otros artefactos del proyecto y la identificación de sus proponentes;
- 3) hacer posible la identificación de los Casos de Uso, sus actores, los flujos principales, alternativos, de error, pre y post-condiciones;
- 4) Permitir la administración de implementaciones identificadas para un sistema;
- 5) apoyar la identificación de *stakeholders* del sistema, así como su jerarquía para solicitud y aprobación de requisitos;
- 6) Apoyar la rastreabilidad entre los Requisitos a través de una Matriz de Rastreabilidad;
- 7) Ofrecer una herramienta de estimación de tamaño y esfuerzo del software;
- 8) Suministrar mecanismos para la priorización de las necesidades de los usuarios;
- 9) Administrar los estados de los Requisitos;
- 10) Permitir el registro y planificación de test para el proyecto de software;
- 11) Proporcionar las estadísticas (métricas) para un proyecto de software;
- 12) Generar documentos de descripción de los Casos de Uso, especificación de requisitos, casos de tests;
- 13) Facilitar la comunicación entre clientes, analistas y equipo de desarrollo.

MATRIZ DE RASTREABILIDAD

“Existen muchas relaciones entre los requisitos y otros requisitos, y entre el proyecto del sistema. Hay también dependencias entre los requisitos y las razones básicas de la proposición de esos requisitos”. [SOMMERVILLE 2003].

Una parte crítica de la administración de proyectos de Software son las alteraciones y la evaluación del impacto del cambio durante el ciclo de desarrollo. Cuando la alteración se propone mientras los requisitos están siendo reunidos, debe identificarse cómo la alteración afecta a otros requisitos. Si la alteración se propone mientras el sistema está en el desarrollo, el impacto de la alteración involucra verificar como la alteración afecta los requisitos, el proyecto del sistema y su implementación. Si la alteración se propone después de que el sistema se puso en producción, también debe haber una comprobación adicional para identificar como todos los *stakeholders* del sistema pueden ser afectados por la alteración.

HERRAMIENTAS EXISTENTES

Con el aumento de las necesidades en administrar requisitos, algunas compañías desarrollaron herramientas para ayudar al mercado.

Analizamos algunas de ellas con el objetivo de ayudar en el momento de la definición del propósito de la herramienta presentada. Evaluamos 2 herramientas existentes en el mercado para administración de requisitos, presentadas en seguida:

CALIBRE DE BORLAND - RM

Herramienta para administración de requisitos, permite que sean definidas matrices y rastreabilidad. Ofrece como recurso el priorización y acompañamiento de los requisitos durante todo el ciclo de vida de la aplicación. [CALIBER 2005]

IBM RATIONAL REQUISITEPRO

Entre las funcionalidades principales observadas en RequisitePro, podemos mencionar la posibilidad de personalizar un grupo de artefactos para determinada organización. También hace posible la integración total con Microsoft Word; controlando la modificación y creación de requisitos. Además el ofrece soporte a Oracle y Microsoft SQL. [REQUISTEPRO 2005]

METODOLOGÍA

La metodología aplicada en el desarrollo del Controla resulta de estudios comparativos realizados en las metodologías principales hoy existentes como Praxis [PAULA FILHO 2002], recomendaciones de PMBoK [PMI STANDARD COMMITTEE 2005] y SWEBoK [ABRAN 2004], siempre tiendo como enfoque, las pequeñas compañías de desarrollo de software. Estas organizaciones no poseen recursos para invertir en la implantación de un proceso de desarrollo complejo y las herramientas de apoyo disponibles en el mercado. Procuramos definir un grupo básico de procedimientos para ser adaptado a la realidad de esas organizaciones.

Un grupo básico de artefactos fue definido, contemplando el proceso de la identificación de las expectativas de los usuarios, identificación y especificación de requisitos, proyecto, testes, identificación de errores, administración de las implementaciones, contemplando así todo el ciclo de desarrollo de software, de la identificación de la marca, proyecto, implementación, tests, pasando por el plan de liberación del software, administración de cambios, permitiendo la rastreabilidad entre todos los artefactos presentes en este ciclo. La Matriz de rastreabilidad es una herramienta importante para el equipo de proyecto, porque permite visualizar de una manera gráfica la relación entre los artefactos y las sospechas eventuales que pueden aparecer durante el desarrollo de un proyecto de software.

La herramienta ofrece la posibilidad de:

1. Levantamiento y registro de las expectativas (visión) y requisitos de los usuarios;
2. Proyecto y especificación del sistema a través de los Casos de Uso
3. Control y seguimiento de las implementaciones identificadas y registradas para un sistema.
4. Registro de los usuarios, miembros de equipo de desarrollo y clientes del sistema, así como su jerarquía para solicitud y aprobación de requisitos;
5. Levantamiento de las reglas comerciales, para que puedan ser comprobables;
6. Validación de los requisitos junto al cliente;
7. Herramientas de estimación de plazos del proyecto basadas en los Casos de Uso.
8. Herramienta de Priorización de Requisitos fundamentada en la participación activa del usuario;
9. Especificación y Detalles de los requisitos;
10. Relaciones de dependencia entre los requisitos para los diferentes grupos de artefactos a través de las Matrices de Rastreabilidad;
11. Definición de los Requisitos non-funcionales;
12. Creación y Registro de los Casos de Tests para un Caso de Uso;
13. Relación de los Casos de Uso que atienden cierto requisito;
14. Registro de los resultados de los tests y correcciones para errores encontrados en las fases de tests o funcionamiento del sistema.
15. Registro del informe de alteraciones en los requisitos;
16. Evaluación del impacto de alteraciones en el sistema a través de la rastreabilidad;
17. Definición de tipo de implementaciones;
18. El registro de Implementaciones para los Casos de Uso de Software;
19. Visualización de artículos sospechosos en las matrices;
20. Consulta, impresión y visualización de Indicadores (estadísticas) para los artefactos del software;

La relación entre los artefactos del sistema puede visualizarse en la ilustración 1.

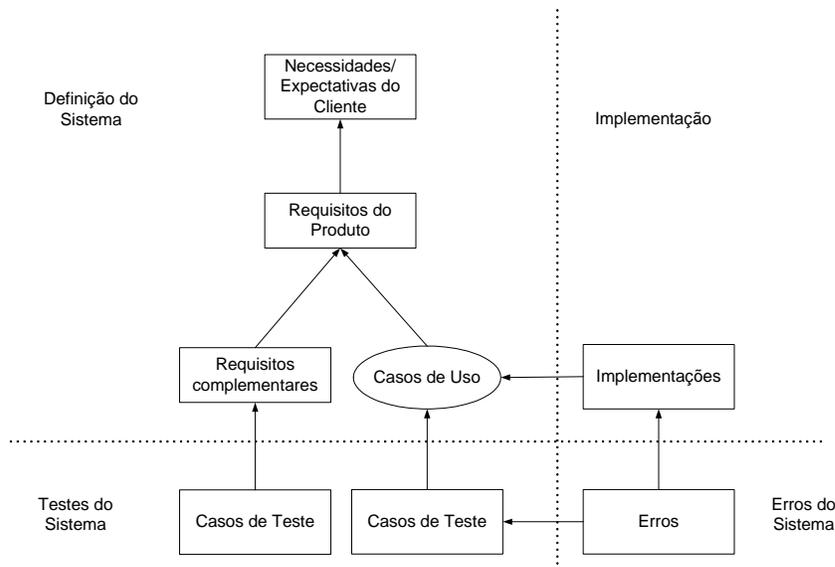


Ilustración 1 - Relación entre los artefactos del sistema en su ambiente [LEFFINGWELL e WIDRIG 2003]

La herramienta todavía ofrece al equipo de proyecto la posibilidad de crear escenarios usando la herramienta de simulación, permitiendo la manipulación de las variables, impresión y cambio de archivos con otros equipos de proyecto. Los modelos implementados son 2, el primero es el Método de priorización de requisitos basado en valor, costo y riesgo [WIEGERS 2003]. El otro modelo se destina a la estimación de esfuerzo y tamaño de proyectos, denominado Método de Puntos por Caso de Uso (UCPM) [KUSUMOTO 2005]. Los dos son útiles, porque ellos ofrecen mayor grado de certeza en los procedimientos realizados en la estimativa y administración del proyecto.

RESULTADOS

El foco inicial del Control son las pequeñas compañías y los desarrolladores autónomos. Se conoce que este público, quizás por la carencia de recursos y del tiempo, no tienen condiciones de aumentar la calidad a su proceso productivo o adquirir herramientas que los ayuden, siendo así, el software contempla varias fases del proceso de calidad, ofreciendo un grupo de funcionalidades, entre ellas:

- Permite el catastro y mantenimiento de proyectos de software;
- Administración de los Requisitos catastrados para un proyecto;
- Administración de Casos de Uso;
- Administración de Casos de Teste y Errores;
- Planeamiento de Liberaciones de software para el cliente;
- Administración de Implementaciones;
- Controle de Dependencia entre implementaciones;
- Matriz de Rastreabilidad (Traceability Matrix) para los artefactos contemplados por el Controla;
- Herramienta de estimación de tamaño de software por Puntos de Casos de Uso;
- Herramienta para priorización de Requisitos [WIEGERS 2003];

- Emisión de documentos de Plan de Proyecto; Casos de Uso; Especificación de Requisitos y Exportación de datos;

Controla cuenta hoy con usuarios en todo Barisl. Aproximadamente 3000 copias fueran distribuidas después de amplia divulgación en importantes sitios especializados en la Web, periódicos y revistas electrónicas.

El resultado de la divulgación fue muy amplio. Investigadores, compañías, los órganos gubernamentales, estudiantes, maestros entraron en el contacto ofreciendo sugerencias y describiendo como el software está siendo útil.

Entre las aplicaciones principales del Controla, percibidas por el retorno de los usuarios, podemos mencionar principalmente:

- Como herramienta de enseñar de la Disciplina Ingeniería de Software;
- Como herramienta de Apoyo al Proceso de pequeñas y medianas compañías;
- Como herramienta de investigación: despertando el interés de los investigadores que ya están en contacto con Controla;

La ilustración 2 presenta la interface principal del Controla



Ilustración 2 - la Interface principal del software Controla.

Controla está disponible¹ en el URL: <http://baixaki.ig.com.br/site/detail34971.htm>

¹ Después de la instalación use usuario “controla” y contraseña “controla”

CONCLUSIONES

Controla está siendo utilizado como herramienta de apoyo al proceso de desarrollo del software en algunos proyectos e incorporando mejoras y ajustes continuos. Este año (2005) será usado como herramienta académica en la disciplina Ingeniería de Software en la Universidad de Viçosa – MG, Brasil. La versión 1.0 ofrece muchos recursos para pequeñas compañías y equipos que no poseen metodologías definidas o no tienen los recursos para adquirir las poderosas herramientas CASE disponibles en el mercado. Una versión multiusuario para ambiente Web está desarrollándose. Hará posible la colaboración on-line para los equipos de desarrollo y grupos de *stakeholders* en múltiples proyectos, incluso los recursos de foro de discusión, *workflow* y acompañamiento de métricas pre-establecidas.

REFERENCIAS

ABRAN, Alain et al, SWEBOK - Guide to the Software Engineering Body of Knowledge, IEEE Computer Society, 2004.

BLASCHEK, José Roberto, 2002, Gerência de Requisitos: O principal problema dos projetos de Software. Rio de Janeiro, Brasil, 4p.

CALIBERRM, Borland Collaborative requirements management system. Disponível por WWW em <http://www.borland.com.br/caliber/> (20/05/2005)

KAN, Stephen H., 2002, *Metrics and Models in Software Quality Engineering, Second Edition*, Addison Wesley, Boston, EUA.

KUSUMOTO, Shinji, Jean Jéferson, 2005, *Effort Estimation Tool Based on Use Case Points Method*, Disponível por WWW em <http://www.spars.info/~lab-db/betuzuri/archive/462/462.pdf>.

LEFFINGWELL, Dean; WIDRIG, Don; *Managing Software Requirements: A Use Case Approach, Second Edition*, Addison Wesley

PAULA FILHO, Wilson de Pádua. Engenharia de Software Fundamentos, Métodos e Padrões. LTC - Editora, 2002.

PMI STANDARDS COMMITTEE 2005, A Guide to the Project Management Body of Knowledge. Upper Darby, PA: . PMI Management Insitute, 1996. Disponível por WWW em: www.pmi.org/publicatn/pmbok. (06/04/2005).

REQUISTEPRO, Rational. Disponível por WWW em <http://www-306.ibm.com/software/awdtools/reqpro/> (20/05/2005)

SOMMERVILLE, Ian., 2003, *Engenharia de Software. 6.ed*, Addison Wesley, São Paulo-SP, BRASIL.

WIEGERS, Karl E., 2003, *Software Requirements, Second Edition*, Microsoft Press.