MODELO PARA EL ANÁLISIS DE LA NEGOCIACIÓN EN LOS PROYECTOS DE SISTEMAS EN UNA FÁBRICA DE SOFTWARE BASADO EN PUNTOS DE CASOS DE USO

MODEL FOR ANALYSIS OF BARGAINING SYSTEMS PROJECTS IN A SOFTWARE FACTORY BASED ON USE CASE POINTS

Paulo Henrique de Lima Maciel, Jandecy Cabral Leite, Adelson Bezerra de Medeiros

Universidade Federal do Pará / Instituto de Tecnologia Galileo da Amazônia – ITEGAM, Brasil, phlmaciel@yahoo.com.br

Instituto de Tecnologia Galileo da Amazônia – ITEGAM, Brasil, jandecy.cabral@itegam.org.br

Universidade Federal do Pará, medeiros@ufpa.br

RESUMEN

El modelado de sistemas se están aplicando cada vez más en los círculos académicos para diferentes propósitos, incluyendo el área de Diseño de Sistemas, con el fin de establecer el número de personas en el equipo, analizar el esfuerzo, el tamaño del software y costos totales del proyecto. Este artículo tiene como objetivo desarrollar un modelo para apoyar el análisis basado en casos de uso Puntos (UCP). Para ello, utilizamos varios métodos de investigación, incluyendo estudios de casos sobre los sistemas que utilizan indicadores tales como líneas de código (LOC), constructivo modelo de costes (COCOMO), Puntos de Función y Puntos de Casos de Uso (PCU). A lo largo del estudio se confirmó mediante la técnica de Puntos de Casos de Uso (PCU), y por consiguiente de los diagramas de casos, usted puede calcular automáticamente los cambios en las estimaciones de cada sistema a las necesidades cambiantes.

Palabras clave: Modelado de Sistemas, Diseño de Sistemas, Software Factories.

ABSTRACT

Modeling systems are being used increasingly in academic circles for different purposes, including the area of systems design, in order to establish the number of people on the team, analyze the effort, software size and total costs the project. This article aims to develop a model to support analysis based on use case points (UCP). To do this, we use several methods of research, including studies on systems using indicators such as lines of code (LOC), constructive cost model (COCOMO), Function Points and Use Case Points (UCP). Throughout the study was confirmed by the technique of Use Case Points (UCP), and therefore case diagrams, you can automatically calculate the changes in estimates of each system to changing needs.

Keywords: Systems Modeling, System Design, Software Factories.

1. INTRODUCCIÓN

Putin (citado Mertins, 2004), afirma que el 26% de los proyectos de software son exitosos. En consecuencia, el 74% de ellos son defectuosos en algún momento, completando su pensamiento, que los proyectos de software se completan con eficacia tienen algunas características comunes. Un problema se produce cuando los requisitos realmente reflejan las necesidades del cliente.

Así, para un proyecto de software para aumentar las posibilidades de ser completado con éxito, debe tener éxito en sus primeras etapas,

incluyendo la fase de negociación. El cliente y el desarrollador debe llegar a un punto común donde ambos conocen sus necesidades y lograr el objetivo deseado, la generación de una negociación en la que ambas partes lograr resultados satisfactorios.

No siempre el cliente y el desarrollador puede entender, por lo que la negociación de un proceso complicado. El director del proyecto puede utilizar algunas métricas de tamaño del software por su ayuda, podemos mencionar entre estos parámetros la métrica de las líneas de código (LOC), constructivo modelo de costes (COCOMO), Puntos de Función (FP) y Puntos de Caso Uso (PCU).

2. JUSTIFICACIÓN

JUNIOR (2007) se centra en el mundo de los negocios hoy en día es extremadamente rápido y competitivo. decisiones a tiempo son la savia de cualquier negocio. Tienen que ser rápido y eficaz, que la declaración se debe al hecho de que el creciente flujo de información en los últimos años, generado principalmente a través de Internet. Las empresas están en constantes cambios en el modo de gestión de la información que puede determinar el éxito o el fracaso de una negociación.

Siguiendo este pensamiento, Paula (2007) afirma que hay dos tipos de negociación: competitiva y comparativa primero es clasificado como un ganar-perder, donde los resultados de un partido tienen prejuicios contra el otro mientras que el segundo se clasifica como un ganar-ganar, donde los resultados son positivos para ambas partes.

En el software comercial no es diferente, los gerentes deben determinar un tiempo aceptable en el ámbito del proyecto, los indicadores existentes para este tamaño de software que ayudan al gerente en la toma de la misma, entre ellos el caso de uso Puntos (PCU.

3. PROBLEMA ESPECIFICACIÓN

No siempre en una negociación ambas partes están satisfechas. Una alternativa ofrecida por un administrador de software en la mayoría de los casos no es aceptado por el cliente. La falta de un modelo hace que la negociación más difícil e imprecisa, ya que el gerente debe estimar los valores de costo y tiempo de forma rápida y segura, lo que no ocurrió con métodos manuales.

Se observó que muchos administradores de software no tienen las herramientas adecuadas, tanto para realizar el alcance de la negociación con sus clientes y aplicar técnicas para la estimación de software. En base a este principio, los administradores de terminar la negociación y aplicación indebida de las técnicas en un manual, que por lo tanto generar varios problemas para la empresa, tales como retraso en la entrega del sistema, generando mayores costos, las estimaciones poco realistas, entre otros .

Frente a este problema, se cree que una posible solución consiste en crear una herramienta para apoyar el gerente, tanto en el momento de la negociación, como en la definición de costo y tiempo del software.

4. NEGOCIACIÓN

Las industrias en general, tienen un proceso continuo, que, según la traducción del Project Management Body of Knowledge (PMBOK), los proyectos no pueden ser considerados como las principales características de un proyecto se les da tiempo, los servicios o productos únicos y los resultados de desarrollo progresivo . Por lo tanto, un proyecto caracterizado por un esfuerzo temporal dirigido a la creación de un producto, servicio o resultado. Tales características no se encuentran en estos procesos de fabricación repetitiva.

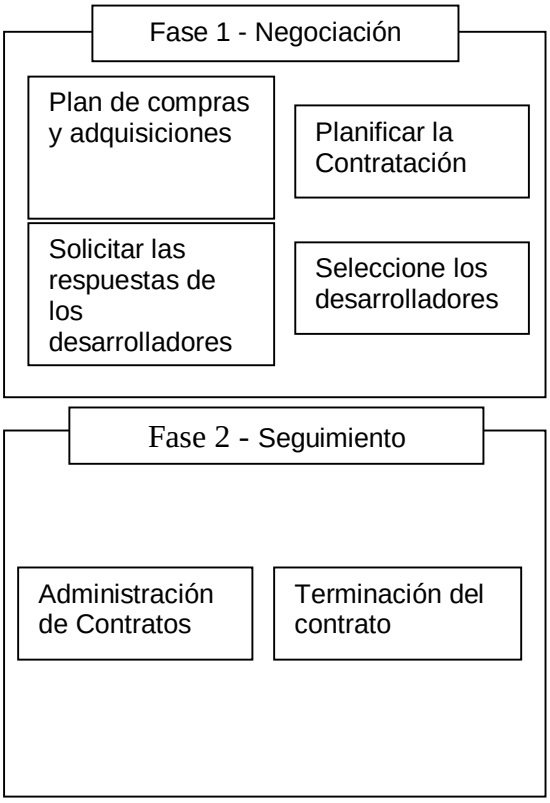
Sin embargo, Aaen, Bottcher y Mathiassen (citado Castor, 2007) sostienen que las fábricas de software cuentan con una estrategia que funciona con dos tipos de servicios: el aumento de la calidad y la productividad en el desarrollo y mantenimiento de software. El primer tipo de servicio se refiere a la creación de nuevos productos, ya sea genérico o la costumbre, los proyectos de software llamada. Este producto cumple con todas las características de un proyecto, por lo que las fábricas de software tienden a utilizar las prácticas en el PMBOK, que según Ávila (2007), es una guía para el conjunto de conocimientos en gestión de proyectos. El segundo tipo es similar a la desarrollada en las industrias, ya que tiene un final definido los resultados y no exclusiva, centrarse exclusivamente en la mejora o reparación de un producto existente.

Como el foco de este trabajo es la negociación entre el desarrollador de software y el cliente, es necesario detallar el área de proceso de

gestión de las adquisiciones, que se divide en dos fases principales que tienen seis subprocesos que figuran en:

- Plan de compras y adquisiciones;
- Planificar la Contratación;
- Solicitar las respuestas de los desarrolladores;
- Seleccione los desarrolladores;
- Administración de Contratos;
- Terminación del contrato.

Os subprocessos podem ser visualizados na figura abaixo.



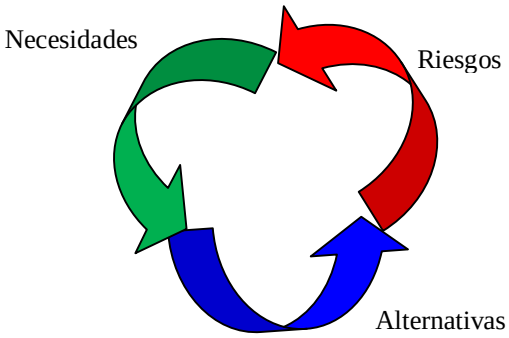
Fases de la gestión de las adquisiciones.

De la investigación en varios concursos públicos, se observó que éstos tienen un elemento común que hace que sea comprensible requisitos entre las partes, es decir: el desarrollador y el cliente. De acuerdo a la Ley Federal N º 8666 del 21 de junio de 1993, art. 40, los anuncios deben obligatoriamente algunos aspectos, entre ellos el objeto de la licitación en un breve y claro.

En los anuncios y las solicitudes de requerimientos de productos de software no es diferente. También necesitan un lenguaje común. Según Vasconcelos (2007), las medidas del tamaño del software son el número más común de líneas de código, el número de puntos de función y el número de puntos de caso de uso. Antes de discutir las medidas comunes de software o en su relación con las etapas del proceso de negociación, habrá una reflexión sobre los conceptos de la negociación.

Por lo tanto, Lewicki y literaria (citado Ramires, 2004) define la negociación como un proceso social básico se utiliza para resolver conflictos. Para CARVALHAL (2004) La negociación es un proceso en el cual dos o más partes con intereses comunes y oposición se reúnen para confrontar y discutir las propuestas con el objetivo explícito de llegar a un acuerdo. Por lo tanto, se observa que la negociación se inicia en diferentes formas a fin de terminar de una manera convergente.

Que complementa esta visión, RIELLO (2007) hace hincapié en que la negociación es un proceso de consecución de objetivos a través de un acuerdo en situaciones donde hay intereses comunes, los opuestos complementarios. Camargo (2007) ilustra el ciclo de la negociación se muestra en la figura siguiente.



CAMARGO (2007).

El ciclo de negociación comienza con la identificación de las necesidades. Luego viene la

fase de las alternativas, que se comparan las ventajas y características de cada uno. En este punto, Camargo (2007) considera que durante la fase de las alternativas aún no están en el punto de decidir, pero se necesita información y argumentos que permitan la comparación y la formación de un juicio crítico sobre cada opción, lo que hace un análisis crítico de cada alternativa.

Después de examinar las alternativas, identificar los riesgos involucrados en cada uno. Sin el balance de riesgos, la negociación se deteriora y el ciclo no está completo. Para Junqueira (citado Picolo, 2004), el proceso de negociación es muy similar al proceso de toma de decisiones. En cierto sentido, se puede decir que el comercio no es más que una toma de decisiones.

Ante estos hechos, nos encontramos con que el comercio debe ser visto como una manera de beneficiar a las partes involucradas, porque no se negocia en contra de alguien, pero en compañía de alguien. Al analizar las alternativas que participan, no es la existencia de elementos comunes. PICOLO (2004) sostiene que con el fin de negociar un acuerdo o un callejón sin salida, sin embargo, los resultados se consideran eficaces cuando no hay otra solución.

Después de la planificación de las compras y adquisiciones, se procede a la planificación

contratación que se define en la visión de la traducción del PMBOK (2004)

requisitos de documentación de productos, servicios y resultados e identificar posibles

los desarrolladores.

En la Fase 1, los procesos de trabajo en conjunto, cada uno de ellos implica el esfuerzo de una o más personas sobre la base de las necesidades del proyecto. Un ejemplo del uso de estos procesos sería una subasta, donde se ejecutan los dos procesos de planificación y, posteriormente, existe un estudio de mercado, en busca de una compañía que proporciona el servicio requerido en la más viable. Una vez seleccionado el desarrollador, se empiece a operar.

El resultado de los desarrolladores de paso seleccionar una base para la implementación de la Fase 2, que se refiere al proceso de seguimiento. La salida de información de este paso como PMBOK (2004), se seleccionan los desarrolladores; Contrato del Plan de Gestión de Contratos, la disponibilidad de recursos, el plan de solicitud de gestión (actualización), y pidió cambios.

Fase 2, el proceso de monitoreo se pueden dividir en: Administración de Contratos y Cierre de contrato. El manual de PMBOK (2004) señala al primero como subproceso de gestión de contratos y la relación entre el comprador y el desarrollador. En esta etapa, hacemos un análisis sobre el actual contrato y no es su progreso.

Por último, es el cierre del contrato que se caracteriza por completar y aprobar cada contrato, o rescindir los contratos existentes que no tienen pendiente.

5. MÉTRICAS

En una fábrica de software, un enfoque en las rutas comerciales con el sistema de información, que de acuerdo a la función Punto de Brasil del Grupo de Usuarios (BFPUG) (2007) tiene su tamaño funcional medido por indicadores, incluyendo la métrica de Puntos Caso de Uso (PCU). Al igual que los ingenieros civiles que hacen sus presupuestos sobre la base de pies cuadrados.

En vista de Pressman (1995), la medición es fundamental para cualquier actividad de ingeniería y de ingeniería de software no es diferente. Por otra parte, hacemos hincapié en que la medida permite obtener la comprensión de un mecanismo de evaluación objetiva.

Con los años, muchas métricas fueron creados con el objetivo de proporcionar el jefe de proyecto con una herramienta que apoya la determinación de los esfuerzos para implementar un proyecto, entre los más conocidos son los modelos para la estimación de la línea de código (LOC), la Modelo de Costos de modo constructivo (COCOMO), el análisis de los puntos de función (FP) y, recientemente, de Puntos de Casos de Uso (PCU).

Uno de los primeros modelos que surgió fue una idea original de Boehm COCOMO desde 1970. El COCOMO tiene varias versiones, la más reciente titulado COCOMO II utiliza líneas de código, puntos de función o puntos objeto de calcular sus estimaciones.

Como HARMAN (2007), un problema que enfrenta el COCOMO es el uso de líneas de código como una medida de su tamaño. Así que ha sido

citado como un ejemplo, de 1000 líneas de código escrito en Java que en teoría representa la funcionalidad mucho más que un hecho en el código ensamblador. Por lo tanto, la dificultad de que representa el tamaño funcional del software por el número de líneas codificadas generado la aparición de nuevas métricas, incluyendo el día de hoy más utilizado, denominado Puntos de Función (FP).

AGUIAR (1998) sostiene que la PF tuvo sus conceptos métricas introducidas por Allan Albrecht en una conferencia de GUIA (Grupo de usuarios de IBM) en 1979. La métrica se caracteriza por la medición de la funcionalidad proporcionada por el software desde el punto de vista de su usuario, es decir, la métrica PF trata de medir lo que hace el software y no la forma en que fue construido.

FATT (2007) subraya que el objetivo de la métrica PF es encontrar una técnica para estimar el esfuerzo de desarrollo de software que es independiente del lenguaje de programación utilizado. En la actualidad, la técnica es gestionada por el Organismo Internacional de Función Punto User Group (IFPUG), que tiene una extensión llamada BFPUG brasileño.

A falta de un indicador simple y el surgimiento de nuevos paradigmas de lenguaje de programación, otras técnicas han surgido. Entre estos paradigmas, la programación orientada a objetos está ganando importancia y cada vez más integrados en las organizaciones de desarrollo de software en Brasil. Con que incluía nuevas métricas derivadas de ella, entre ellos la métrica de Puntos de Casos de Uso (PCU).

6. CONCLUSIÓN

A lo largo del estudio se confirmó mediante la técnica de la UCP, y por lo tanto de los diagramas de casos de uso, se puede calcular automáticamente los cambios en las estimaciones del tamaño de los requisitos del sistema para cada pequeño cambio, sólo rehacer algunos cálculos.

La UCP técnica, como una técnica formal y han mostrado resultados satisfactorios en esta investigación, proporcionando una alternativa para las empresas que no tienen definido métricas y para las empresas no ha dado lugar a resultados satisfactorios con las métricas que utilizan hoy en día.

En la preparación de los diagramas de la metodología se utilizó el sistema de Iconix, fue elegida porque, como la UCP métricas, que también se basa en utilizar los diagramas de casos, lo que facilita la construcción de artefactos. Una de las ventajas de la metodología de Iconix es la facilidad para supervisar todas las etapas para asegurar que todos sus requisitos se cumplen, esto ocurre a través de una característica única llamada Trazabilidad de Requisitos.

Durante la realización de diagramas, la metodología Iconix demostrado ser un sólido y continuo, ya que cada artefacto construido sirve como entrada a la siguiente, con la metodología que se reúnan todas las expectativas que tiene.

En el escenario de la negociación de proyectos de software, el modelo propuesto mostró un ejemplo de cálculo de las estimaciones de horas de programación. Esta función puede acelerar una negociación futura, ya que los valores se encuentran una manera más rápida de dar una idea del tamaño de la propuesta y por lo tanto el tiempo de desarrollo viable.

Todas las pruebas realizadas en el sistema tuvo resultados positivos, incluyendo comparaciones con trabajos anteriores.

7. AGRADECIMIENTOS

Ao Instituto Galileo de Tecnología de la Amazonia - ITEGAM por el acuerdo firmado con la Universidad Federal de Pará - UFPA, Master Profesional en Ingeniería de Procesos Industriales / CMPPI. En la Fundación de Apoyo a la Investigación del Estado de Amazonas - Fapeam por el apoyo financiero del PPGE / UFPA.

8. REFERENCIAS BIBLIOGRÁFICAS

AGUIAR, M. O que é um Ponto de Função. Disponível em: http://www.bfpug.com.br/Artigos/Dekkers-PontosDeFuncaoEMedidas.htm

ÁVILA, M. d' PMBoK e Gerenciamento de Projetos http://www.mhavila.com.br/topicos/gestao/PMBoK.html

CAMARGO, R.J.F. A NEGOCIAÇÃO EFICAZ.
Disponível em: http://www.iep.org.br/lit/neg.html

CARVALHAL, E. Negociação. Rio de Janeiro: Vision, 2004.

CASTOR, M.E. de Fábricas de Software: Passado,Presente e Futuro.
Disponível em:
http://www.unibratec.com.br/revistacientifica/diretorio/edicao1/artigo_fabricasw_tecnologus_ final.pdf

FATTO. Consultoria e Sistemas.
Disponível em: http://www.fattocs.com.br/faq.asp

FREIRE, H. Calculando Estimativas: o Método de Pontos de Caso de Uso.
Disponível em:
http://www.cnnt.com.br/files/usecasepoints.pdf

HARMAN, M. How long is this going to take?
Disponível em:
http://www.dcs.kcl.ac.uk/staff/mark/exe11.html

HEIMBERG, V; GRAHL, E.A. Estudo de Caso de Aplicação da Métrica de Pontos de Casos de Uso numa Empresa de Software
Disponível em:
http://www.inf.furb.br/seminco/2005/artigos/130-vf.pdf

JUNIOR, A.P.S. Investigação concernente ao uso da videoconferência na negociação empresarial: um estudo de caso.
Disponível em:
http://64.233.169.104/search?q=cache:LCXmcK51RTcJ:www.ead.fea.usp.br/Semead/6semead/PGT/005PGT%2520-%2520Investiga%25E7ao%2520Concernente.doc

MELO, W. Qualidade+Testes de Softwares=Qualidade de Software.
Disponível em:
http://www.ogerente.com.br/qual/dt/qualidade-dt-ws-testes_software.htm

MONTEIRO, T. C; BELCHIOR, A. D; TUCP: Uma Extensão da Técnica UCP.
Disponível em:
http://www.inf.ufes.br/~sbqs2006/?q=node/8

OLIVEIRA, F. Capacitação em Técnicas de Teste de Software.
Disponível em:
http://www.pucrs.br/inf/eventos2003/workshop/arquivo/MiniCurso_Teste.pdf

PAULA, M.V. Um sistema de suporte a negociação cooperativa para destinação de excessos.
Disponível em:
http://cronos.cos.ufrj.br/publicacoes/reltec/es62203.pdf

PICOLO, A.A. Projeto de Software para Negociações.
Disponível em:
http://nourau.eadstrong.com.br/document/get.php/280/Picolo_AngeloA_TCCI2005.pdf

PRESSMAN, R. S. Engenharia de Software. 2.ed. São Paulo:Makron Books,1995.

RAMIRES, V.C.J.J. Negociação de requisitos no processo de desenvolvimento de software.
Disponível em:
http://www.di.fc.ul.pt/sobre/documentos/tech-reports/04-19.pdf

RIELLI, E.M. A ARTE DE NEGOCIAR.
Disponível em:
http://www.bioenergetica.com.br/novo/artigo.asp?cod_artigo=44

[VASCONCELOS, A. Introdução a Métricas de Software.
Disponível em:
http://www.cin.ufpe.br/~if720/slides/introducao-a-metricas-de-software.ppt

VIJAYKUMAR, N. L. Introdução aos testes de Software.
Disponível em:
http://www.lac.inpe.br/~vijay/download/ELAC2007/Vijay_Testes_De_Software_E_Avaliacao_De_Desempenho_PARTE_II.pdf

WAZLAWICK, S. R. Análise e projeto de sistemas de informação orientados a objetos. Rio de Janeiro: Campus, 2004.