

TITULO: Importancia de la Ingeniería de Software en la producción de software.

TITLE: The importance of Software Engineering in software production.

AUTORES:

Ing. Jenny Ruiz de la Peña

Ing. Oscar Aguilera Cruz

PAÍS: Cuba

RESUMEN:

Aborda la importancia de la Ingeniería de Software. Hace referencia a los actuales problemas que presenta su utilización y en muchos casos, su empleo totalmente empírico. Establece las principales situaciones como la incomprensión entre desarrolladores y usuarios o clientes, el largo tiempo de desarrollo, la corrección de fallas innecesariamente. Propone la enseñanza de la Ingeniería de Software aparejada con la de la programación como vía para eliminar la causa principal de lo planteado. Ejemplifica las ventajas del desarrollo de productos informáticos mediante el uso de una metodología de Ingeniería de Software con el Sistema de Negociación Comercial y el Sistema de Administración de las Capacidades. Hace especial énfasis en cómo pueden desarrollarse de forma simultánea dos sistemas que tienen intercambio de información por medio de la utilización de la Ingeniería de Software.

PALABRAS CLAVES: INGENIERÍA DE SOFTWARE, NEGOCIACIÓN COMERCIAL, CAPACIDAD DE PRODUCCIÓN, INFORMÁTICA.

ABSTRACT:

This work is about the importance of Software Engineering. It makes reference to the current problems presents in its use and, in many cases, their completely empiric utilization. It establishes the main situations like the incomprehension between developers and users or clients, the long time of development, the unnecessarily correction of problems. It proposes the teaching of the Software Engineering at the same time with the programming to eliminate the main cause of the problem. In this article the advantages of the development of computer products are exemplified by the use of a methodology of Software Engineering with the System of Commercial Negotiation and the Capacity Managed System. It makes special emphasis in how they can be developed simultaneously two systems with the use of the Software Engineering.

KEY WORDS: SOFTWARE ENGINEERING, COMMERCIAL NEGOTIATION, PRODUCTION CAPACITY, COMPUTER SCIENCE.

INTRODUCCIÓN

Debido a la introducción de las Tecnologías de la Informática y las Comunicaciones en las diferentes áreas de la vida cotidiana y empresarial, la

mayoría de los países del mundo, en los últimos años, ha aumentado considerablemente la producción o consumo de productos informáticos. Esto ha implicado que la industria de software crezca considerablemente y al mismo tiempo mejoren indicadores en la producción.

Con el gran auge del uso de las potencialidades computacionales, todos los sectores de la economía cubana han incluido de una forma u otra las computadoras en su desarrollo cotidiano. Junto a las computadoras las empresas necesitan software, ya sea desarrollado a la medida o no, que pueda soportar su quehacer diario.

El objetivo de este trabajo mostrar la importancia de la Ingeniería de Software y hacer referencia a los actuales problemas que presenta su utilización y en muchos casos, su empleo totalmente empírico.

MATERIALES Y METODOS.

Los estudios preliminares estuvieron destinados a la constatación de la situación problémica y recogida de datos iniciales. Se realizó una revisión bibliográfica para el estudio del desarrollo de productos informáticos y sus características.

Se utilizaron métodos empíricos y teóricos. La observación y la entrevista se emplearon como puntos de partida para la investigación y su posterior seguimiento. La modelación permitió determinar una abstracción de la realidad para agilizar el desarrollo del trabajo. El método histórico lógico dio las pautas para la evolución y el desarrollo de la investigación.

RESULTADOS DEL TRABAJO

El software en su desarrollo pasa por varias etapas que se pueden agrupar en estos cuatro grandes grupos: Concepción, Desarrollo, Prueba y Explotación.

Sin duda alguna las empresas desarrolladoras de software pasan de una forma u otra por estas etapas. Sin embargo, cabe preguntar cuántas de las empresas y equipos de desarrollo siguen metodologías capaces de administrar y documentar todo el ciclo de vida de un producto informático.

Desde la Concepción del software el equipo de desarrollo tiene que empezar a documentar cada decisión que se toma, en aras de poder satisfacer el futuro desarrollo, pruebas y evolución del sistema.

Muchas veces el desarrollo de software se hace de forma empírica. Si bien se atraviesan los cuatro grandes grupos de Concepción, Desarrollo, Prueba y Explotación, se cometen errores, principalmente durante la Concepción y el Desarrollo. Las consecuencias de no elegir una metodología o modelo de Ingeniería de Software hacen que los desarrolladores no se comuniquen del todo bien con los clientes y usuarios. Esto provoca que la captura de requerimientos, tan importante durante la Concepción, por su papel iniciador del proceso, no siempre se complete y comprenda a su debido tiempo.

Muchas veces se procede a implementar los requerimientos de software directamente, aún antes de haber realizado un análisis y un diseño profundo de estos. Por consiguiente, no se hace una buena elección de los lenguajes de programación o de la plataforma de explotación. Lo más usual radica en que no se determinan inicialmente los vínculos existentes entre los requerimientos de software. No se puede por tanto definir correctamente cuáles son las prioridades reales desde el punto de vista de los clientes y usuarios, y por supuesto, desde el de los desarrolladores, quienes deberían determinar qué se necesita primero para cumplir con lo solicitado.

La estimación del tiempo de desarrollo o costos se ven afectadas si no se comprende el real alcance de lo que se realiza. Aumentan, por tanto, no se satisfacen los convenios realizados, conllevando a desilusiones e informalidades. Al realizar las pruebas de sistema se encuentran más problemas, ya sean de programación o de requerimientos mal implementados o incorrectamente comprendidos por los desarrolladores. Se debe entonces retroceder para corregirlos con un gran aumento de tiempo. En ocasiones, lo que se realiza debe ser totalmente cambiado para poder entregar un producto que finalmente satisfaga lo planteado al inicio.

Se adolece de falta de documentación en cada una de las etapas o de documentación realizada en distintos formatos, cada uno de los cuales elegido por los desarrolladores a su conveniencia o entendimiento. El paso de una etapa a la siguiente puede verse afectado por esto, pues los desarrolladores no comprenden la documentación entregada por otros. Es por esta razón que se hace tan necesaria la utilización de estándares para evitar estas situaciones.

Existen diferentes metodologías o aproximaciones a estas en el mundo, y evidentemente todas tienen sus ventajas y desventajas. A pesar de existir varias metodologías es sumamente importante seleccionar una que se adapte a las características propias del software a desarrollar.

Cuando se va a seleccionar una metodología es necesario tener en cuenta un conjunto de características que permitirán al equipo de desarrollo aprovechar al máximo las facilidades y potencialidades de la metodología seleccionada.

La metodología debe adaptarse fácilmente al tipo de sistema a desarrollar. Existen metodologías que se ajustan a varios tipos de sistemas mientras que otras han sido creadas para algo en particular. Se debe hacer un análisis de varias y elegir una de acuerdo con las características esenciales de lo que se desea entregar.

Debe facilitar la entrega de documentación de una etapa a otra y evitar el exceso de información que provoque confusión. Se hace vital que contenga formatos comprensibles para los clientes y usuarios. Con esto se puede garantizar en las etapas iniciales que los requerimientos capturados sean los correctos y entendidos en su totalidad.

Por otra parte es muy importante que favorezca el desarrollo progresivo del producto informático. De esta forma en breve tiempo los usuarios podrán

comenzar a utilizarlo, encontrar las fallas en un momento en que no tenga que afectar grandemente el desarrollo posterior de otros requerimientos. Al mismo tiempo, esto favorece la disminución del rechazo al cambio de los usuarios.

La principal causa de los problemas mencionados anteriormente en la aplicación de metodologías o modelos de Ingeniería de Software radica en que se aprende primero a programar. Se separa la implementación, como si en lugar de formar parte indivisible de la Ingeniería de Software fuera algo totalmente ajeno a ella. Luego de haber realizado productos sencillos o incluso ejercicios de menor complejidad para aprender a programar, se comienza la enseñanza de la Ingeniería de Software. Cuesta mucho cambiar la forma de pensar y entender qué significa realmente desarrollar un producto informático, cuáles son sus etapas y cómo debe realizarse de forma tal que se obtenga con buena calidad y sea sostenible en el tiempo. La enseñanza de la Ingeniería de Software debe ir aparejada de la enseñanza de la programación para evitar que ocurra lo anterior.

Ejemplo de utilización de la Ingeniería de Software en dos sistemas de forma simultánea.

Es también digno de destacar que la utilización de la Ingeniería de Software puede favorecer la unión de distintos productos informáticos para utilizar de cada uno lo que se necesita. Así puede agilizarse la producción y la utilización de estos sistemas. Un ejemplo de esto es la realización y la utilización del Sistema de Negociación Comercial y el Sistema de Administración de las Capacidades.

Ambos sistemas responden a la necesidad de mejorar el proceso de negociación comercial en las empresas. La negociación es un proceso de construcción de acuerdos que ha trascendido el acotado mundo de las relaciones comerciales, penetrando en todas las dimensiones de la vida. Los empresarios, gerentes y profesionales del siglo XXI requieren estar preparados para manejar este proceso. Los negociadores están encargados de responder a sus intereses y deben tener una visión global de la misma.

En el proceso de compra-venta de recursos o productos las empresas se enfrentan entre sí por medio de los negociadores que pretenden lograr el mejor negocio para su empresa. Las negociaciones se hacen a largo plazo por lo que los negociadores deben conocer cuáles son las posibilidades reales para satisfacer pedidos. No resulta sencillo controlar cómo se comportará el mercado en fechas futuras o si ya se tienen compromisos que no permitan contratar los nuevos que se contraen. Los negociadores tienen que ser capaces de, dominando esta información, dar una respuesta rápida a los negociadores con que tratan y que, efectivamente, sea la mejor para la empresa.

La capacidad de producción de una empresa que necesitan conocer los negociadores está relacionada directamente con la producción de los bienes o servicios que tenga. Una empresa puede tener una o varias líneas de

producción con características propias que habrá que analizar para poder determinar la capacidad de producción de cada una de las líneas.

El cálculo de las capacidades, debido a la gran variedad de posibles combinaciones entre las diferentes líneas de producción y a la gran variedad de almacenes que pueden existir en una organización, para dar respuesta a una negociación comercial, puede verse afectado, fundamentalmente por el tiempo que demora determinar las capacidades.

Actualmente resulta complejo realizar las negociaciones y se hacen intuitivamente. Los negociadores no son capaces de valorar de forma dinámica las propuestas previendo las consecuencias que puedan traer las alternativas elegidas. En la mayoría de las negociaciones comerciales en que sí se realizan los cálculos de las capacidades, esto se hace en varias ocasiones. Sumando el tiempo que se emplea en su determinación, pueden contribuir a que la negociación comercial no se desarrolle con la calidad que se requiere.

El desarrollo de un sistema de apoyo a la toma de decisiones en el proceso de negociación comercial brinda utilidad a infinidad de empresas que actualmente no cuentan con una herramienta con estas características. Empresas estatales o privadas necesitan contar con una herramienta de este tipo que los apoye en sus decisiones para que, durante el período posterior a la negociación, realicen su objetivo general y las metas sociales que persiguen.

El Sistema de Negociación Comercial se ha desarrollado con el propósito de planificar las negociaciones comerciales. Para ello es necesario sopesar una infinidad de variantes en un corto tiempo teniendo en cuenta fechas de entrega o recepción, productos o materias primas, cantidades, precios de oferta de contra oferta y fundamentalmente las capacidades reales de la empresa de satisfacer pedidos.

Por su parte el Sistema de Administración de las Capacidades tiene vida propia como su nombre lo indica. Su objetivo es determinar qué capacidad tiene la empresa en determinado momento y planificar cómo serán utilizados sus medios de producción teniendo en cuenta las distintas líneas y estaciones de trabajo con las que cuente así como también con las materias primas disponibles. Por lo que el Sistema de Negociación Comercial puede utilizar la información que brinda en su propio beneficio.

La unión de estos dos sistemas elimina la pérdida de orientación en el proceso de negociación que hace que los empresarios tomen decisiones desacertadas de acuerdo al momento en que se esté negociando. Como consecuencia, permite liberar al negociador de la incertidumbre en materia de toma de decisiones en el proceso de negociación comercial. La información del estado de los indicadores de resultado y proceso es conocida por el negociador de forma muy rápida.

Para poder desarrollar de forma simultánea ambos sistemas se necesitó desde un inicio definir qué alcance tendría cada uno. La Ingeniería de Software fue

utilizada desde que se comenzaron a comprender los procesos implicados y cómo se relacionaban unos con otros.

Para su desarrollo se utilizó la metodología Proceso Unificado de Desarrollo de Software (RUP, por sus siglas en inglés). Esta metodología utiliza el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés). RUP se adapta perfectamente a los sistemas propuestos de Negociación Comercial y de Administración de las Capacidades. Permitió un acercamiento iterativo e incremental de los productos. Gracias a ella fue posible establecer una única forma de trabajo para los sistemas que mediara entre los desarrolladores, clientes y usuarios de ambos trabajos.

Durante la captura de requerimientos y el análisis se determinaron los puntos de intercambio entre los sistemas. Así se diseñaron posteriormente las clases que garantizarían el flujo de información entre uno y otro. RUP indica que se deben priorizar los procesos más importantes para comenzar por ellos y no por otros. Esto fue de gran ayuda para definir qué se podía hacer de forma concurrente en el Sistema de Negociación Comercial y el Sistema de Administración de las Capacidades. También influyó en gran medida para poder discernir en los aspectos más importantes que se necesitaban para que los dos sistemas pudieran ser utilizados tal y como se deseaba.

El Sistema de Negociación Comercial fue desarrollado con tecnología Web.

RUP utiliza las extensiones de UML para este tipo de aplicaciones. El Sistema de Administración de las Capacidades fue desarrollado con un lenguaje de programación orientado a objetos. RUP fue diseñada especialmente para eso.

De esta forma, aunque los dos sistemas se comunican entre sí y no tienen las mismas características de implementación, utilizan una misma metodología que ayudó a obtener un producto de calidad en el menor tiempo posible.

La documentación requerida en cada etapa por la metodología fue llevada a cabo. Esto permitirá en un futuro realizar mantenimientos ya sean correctivos, adaptativos, preventivos o de mejora.

CONCLUSIONES

Como se puede apreciar, es posible desarrollar productos informáticos de pequeña, mediana y gran envergadura con calidad. La Ingeniería de Software siempre juega un papel sumamente importante en cada uno de ellos, incluso en los más sencillos.

Es necesario cambiar la forma de desarrollar software empírico y pasar a software con la utilización de métodos de ingeniería que hagan progresar el actual desarrollo de software en cualquier entidad. Sin importar dónde se desarrolla y qué complejidad tengan, las metodologías de desarrollo de software deberán estar presentes. El objetivo es contar con software

sostenible, con larga vida de explotación, fácil de utilizar y que sean desarrollados en breve tiempo. Sin la Ingeniería de Software no será posible llevar de la mano la calidad del producto y el poco tiempo de desarrollo.

BIBLIOGRAFÍA

1. The Art of Negotiating Software. [documento en línea] <http://www.notjustsurveys.com/HR/neg.htm> [consultado: 7 jun. 2002]
2. Bases de datos. [documento en línea] <http://www.prematuros.cl/tallermarzo2003/basesdatos.htm> [consultado: 31 mar. 2003]
3. Conallen, Jim. Building web applications with UML. 2. ed. Boston : Addison Wesley, 2003. 458 p.
4. Jacobson, Ivar. El Proceso Unificado de Desarrollo de Software / Ivar Jacobson, Grady Booch, James Rumbaugh. Madrid: Addison Wesley, 2000. 438 p.
5. Larman, Craig. UML y patrones. 2. ed. Madrid: Prentice Hall, 2003. 590 p.
6. Matos, Rosa María. Diseño de bases de datos. La Habana : CUJAE, 1997. 30 p.
7. Negociación Comercial Avanzada. [documento en línea] <http://www.louisville.edu> [consultado: 2 feb. 2005]
8. Software to help you prepare for any negotiation. [documento en línea] <http://www.negotiatorpro.com/negprosoft.html> [consultado: 7 feb. 2006]
9. Taller de Negociación Comercial. [documento en línea] <http://www.isede.edu.uy/cursos/index.php> [consultado: 1 may. 2005]
10. Técnicas de negociación comercial. [documento en línea]. <http://www.hipermarketing.com/nuevo%204/columnas/habib/nivel4contratacion.html> [consultado: 1 may. 2005]
11. Win Squared Persuasion, Sales and Negotiation Software. [documento en línea] <http://www.winxwin.com/left2.htm> [consultado 22 feb. 2006]

DATOS DE LOS AUTORES

Nombre:

Ing. Jenny Ruiz de la Peña
Ing. Oscar Aguilera Cruz

Correo:

jruizp@facinf.uho.edu.cu
oscar@facinf.uho.edu.cu

Centro de trabajo:

Universidad de Holguín "Oscar Lucero Moya". Facultad de Ingeniería Informática. Departamento de Informática. Ave XX Aniversario Gaveta Postal 57 Holguín 80100 Cuba. Teléfono: (+53) (24) 48 2675, Fax: (+53) (24) 46 8050.

© Centro de Información y Gestión Tecnológica (CIGET), 1995. Todos los derechos reservados Última actualización:
29 de Marzo del 2010