

FACTORES IMPORTANTES EN EL DESARROLLO DE UN PROYECTO DE SOFTWARE



ABSTRACT

To this article, it is shown the importance that has the project management in the development of software so much in the aspect of the conformation of equipments (teams), for taking a successful execution of the project; since (as, like) in the different processes that can exist in the same one, facilitating the detection of possible mistakes in the production (elaboration) and advices (councils) to anticipate (prepare) them managing an index of high quality of the product to present.

There is important to have clearly the model who will be chosen for the development of a process of software, since from this one, there decide the stages in which the equipment (team) of work will distribute the pertinent tasks to expire with the aims (lenses), the tools and methods that this one will have to use for the product development and the methodologies used to obtain the awaited results.

KEYWORDS

Project management, methodology rup, unified process, quality product, language UML

PALABRAS CLAVES

Lenguaje UML, gestión de proyectos, metodología rup, proceso unificado, producto de calidad.

RESUMEN

Con el presente artículo, se presenta la importancia de utilizar la gestión de proyectos en el desarrollo de software, tanto en el aspecto de la conformación de equipos de trabajo, para llevar una ejecución exitosa del proyecto, como en los diferentes procesos que pueden existir en el mismo, facilitando la detección de posibles errores en la elaboración y consejos para prevenirlos y alcanzando un índice de alta calidad del producto por presentar.

Es importante tener claro el modelo que se elegirá para el desarrollo de un proceso de software, ya que, a partir de este, se determinan las etapas en las cuales el equipo de trabajo distribuirá las tareas pertinentes para cumplir con los objetivos, las herramientas y métodos que éste deberá utilizar para el desarrollo del producto y las metodologías utilizadas para obtener los resultados esperados.

INTRODUCCIÓN

Cuando se decide emprender el camino del diseño y la construcción de software, la primera pregunta que nos invade es ¿cómo empezar? y detrás de esta vienen otras preguntas como ¿qué herramientas?, ¿Qué métodos? ¿Qué procesos? ¿Qué tareas? ¿Quiénes conformarían el equipo de trabajo? y una serie de interrogantes que llevan a una incertidumbre sobre el futuro de nuestro proyecto y la calidad del producto por construir.

Entre 1960 y 1985 se definía al éxito de un proyecto sólo con base en su calidad [1]. Es decir, un proyecto que cumpliera con los objetivos de calidad preestablecidos se definía como exitoso.

Luego, entre 1985 y 1993, se definió un proyecto como exitoso cuando, además de cumplir con la calidad, cumplía con los plazos y presupuesto definidos en el plan del proyecto [1].

En la actualidad, no basta con cumplir la calidad, plazos y presupuesto para el éxito de un proyecto, sino que, además de estos objetivos, es necesario que el proyecto cumpla con la satisfacción del cliente. ¿De qué serviría un proyecto de una calidad excepcional, que se finalizó en el plazo previsto, utilizando los recursos preestablecidos, si luego nadie compra los productos de ese emprendimiento?

DESARROLLO

"Los sistemas de software son creaciones complejas, realizan muchas funciones, están contruidos para lograr objetivos diferentes y con frecuencia conflictivos, comprenden muchos componentes, muchos participantes de disciplinas diferentes intervienen en el desarrollo de estos componentes, el proceso de desarrollo y el ciclo de vida del software a menudo abarcan muchos años y por último, los sistemas complejos son tan difíciles de comprender, incluso durante su fase de desarrollo, que algunas veces nunca llegan a ser terminados. Además, los proyectos de desarrollo de software están sujetos a cambios constantes, debido a que los requerimientos son complejos; necesitan ser actualizados cuando se descubren errores y cuando los desarrolladores tienen una mejor comprensión de la aplicación" [2].

En todo proyecto de software existe la necesidad de contar con personal capacitado, seleccionar el mejor proceso de acuerdo con problema que se vaya a tratar y por supuesto, una excelente

planificación, con el fin de obtener un producto a tiempo y de calidad. Todo esto conforma lo que se conoce como gestión de los proyectos.

Todo proyecto debe tener consigo una planificación previa; no se debe aventurar al éxito sin antes conocer los beneficios, contras y coste del proyecto. Para ello es necesario que se pongan en funcionamiento cuatro características muy importantes en la gestión: personal, producto, proceso y proyecto [2].

Muchas veces el éxito o fracaso de los proyectos depende fundamentalmente del equipo humano con el que se trabaje, de los procesos válidos y que realmente le sirvan a su proyecto. La ejecución de las cuatro características marcará el rumbo del proyecto.

El factor humano es muy importante en el desarrollo de software, muchos empresarios famosos, líderes de empresas tecnológicas, coinciden en que el éxito que han alcanzado sus empresas no se debe a las herramientas que utilizan, es la gente y el trabajo en equipo. Todos los miembros del equipo deben tenerse confianza y distribuir la carga de trabajo según el problema que se esté tratando.

Un ejemplo de ello se ve en la Universidad Francisco de Paula Santander Ocaña, institución que implementa dentro del programa del plan de estudios de ingeniería de sistemas, áreas donde los estudiantes desarrollan proyectos de software orientados a solucionar necesidades tanto de la institución como de diferentes entes de la provincia de Ocaña. En estas áreas se forman equipos de trabajo, los cuales tienen la responsabilidad de desarrollar durante cierto tiempo pequeños proyectos de software.

La figura 1 representa un grupo de estudiantes de la Universidad Francisco de Paula Santander Ocaña, los cuales conforman varios equipos de trabajo de diferentes proyectos de software.



Figura 1. Equipos de trabajo de diferentes proyectos, pertenecientes a las áreas de proyectos de ingeniería de software de la UFPSO.

La pregunta ¿cuánto me va a costar? es común escucharla por parte del cliente, cuando pide que le construyan una solución. Pues bien, todo producto requiere estimaciones cuantitativas y una adecuada planificación. Una apropiada recolección de información y un análisis detallado de los requerimientos proporciona la información necesaria para dar una estimación del costo del producto.

Antes de planear un proyecto, se deben establecer los objetivos y el alcance que tendrá el proyecto, además de sus restricciones técnicas y de gestión. Con una buena planificación se puede estimar el tiempo que tomará desarrollar o construir el producto y redimensionar el valor cuantitativo del producto. Para lograr rapidez en la construcción del producto, se debe dividir la carga de trabajo

entre el equipo de desarrollo, es decir, dividir el problema. Esto con el fin de desarrollar el producto con mayor eficiencia, eficacia y en el tiempo acordado con el cliente.

El proceso del software proporciona un marco de trabajo de las tareas que se requieren para construir software de alta calidad. El equipo de desarrollo debe elegir el proceso adecuado, que le permita obtener una solución o producto que satisfaga las necesidades o requerimientos del cliente [1].

El gestor del proyecto debe elegir el modelo de procesos adecuado para ser aplicado en la construcción del software, el apropiado para los clientes que han solicitado el producto y el personal que hará el trabajo, las características del producto, el ambiente del proyecto en el que trabajará el equipo de desarrollo del software [3].

La elección del modelo depende de la naturaleza del proyecto y de la aplicación, de los métodos y herramientas por utilizar y de los controles y entregas que se requieren.

Seleccionado el modelo de procesos, se desarrolla una planeación preliminar del proyecto, basada en las actividades del marco de trabajo. Cuando el equipo de desarrollo de software ha definido correctamente el modelo proceso, este debe ser flexible y adecuado para el proyecto. El proceso se puede descomponer para lograr ejecutar correctamente las actividades y tareas del marco de trabajo. En la figura 2, se muestra una relación entre los elementos del proceso de software.

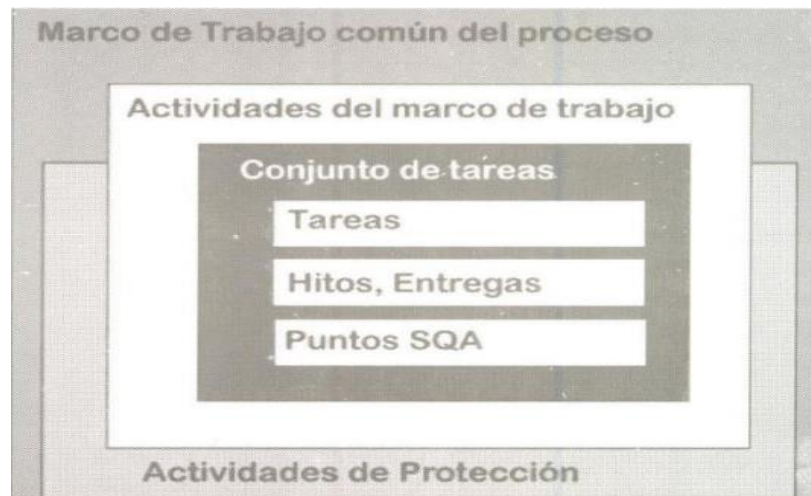


Figura2. Relación entre los elementos del proceso de software

Todo el desarrollo del software se puede caracterizar como bucle de resolución de problemas, en la cual se encuentran cuatro etapas distintas: el estado actual, que representa el comportamiento del sistema actual; la definición de problemas, la cual identifica el problema específico por resolverse; el desarrollo técnico, que presenta alternativas de solución para dicho problema, a través de la aplicación de algunas tecnologías y, finalmente, la integración de soluciones, la cual ofrece los resultados.[3]

Existen diferentes tipos de modelos, cada uno de ellos se enfoca en un aspecto del sistema y revela un tipo de información. El modelo o el conjunto de modelos por utilizar en la solución de un problema

deben seleccionarse cuidadosamente, pues de igual manera que un modelo adecuado puede ayudar a comprender el sistema e identificar detalles importantes para la solución, un modelo inapropiado puede generar confusión. [4]

En el campo del desarrollo de software, se han propuesto varios paradigmas; cada uno tiene sus propios modelos así como sus procedimientos y metodologías. Cada uno de ellos puede conducir a una solución del problema, aunque con diferentes costos y beneficios. Si el proyecto se enfoca bajo el paradigma Orientado a Objetos es conveniente utilizar un modelado orientado a objetos en el que se utilicen concepto, mecanismos y artefactos propios de este enfoque.

El Proceso Racional Unificado (Rational Unified Process en inglés, RUP) es un proceso de desarrollo de software y, junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.[2]

El Proceso Unificado es un proceso de software genérico que puede ser utilizado para una gran cantidad de tipos de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de competencia y diferentes tamaños de proyectos. Suministra un enfoque disciplinado en la asignación de tareas y responsabilidades, dentro de una organización de desarrollo. Su meta es asegurar la producción de software de muy alta calidad que satisfaga las necesidades de los usuarios finales, dentro de un calendario y presupuesto predecibles.

El proceso unificado tiene dos dimensiones:

Un eje horizontal que representa el tiempo y muestra los aspectos del ciclo de vida del proceso, a lo largo de su desenvolvimiento.

Un eje vertical que representa las disciplinas, las cuales agrupan actividades de una manera lógica de acuerdo con su naturaleza.

La primera dimensión representa el aspecto dinámico del proceso conforme se va desarrollando; se expresa en términos de fases, iteraciones e hitos (milestones).

La segunda dimensión representa el aspecto estático del proceso: cómo es descrito en términos de componentes del proceso, disciplinas, actividades, flujos de trabajo, artefactos y roles [4]. En la figura 3 se muestran las disciplinas y fases que componen el modelo de desarrollo RUP.

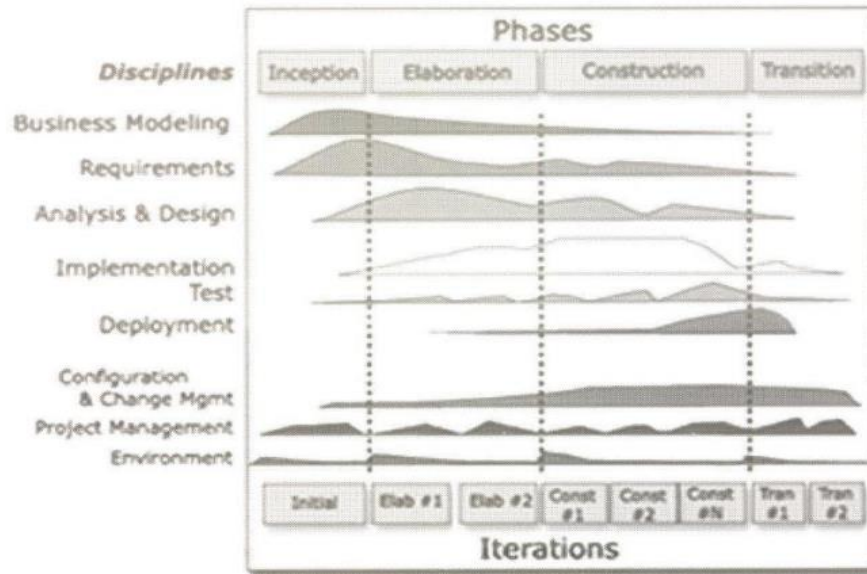


Figura 3. Disciplinas y fases que componen el modelo de desarrollo RUP.

El proceso unificado se basa en componentes, lo que significa que el sistema en construcción está hecho de componentes de software interconectados por medio de interfaces bien definidas.

El proceso unificado usa el Lenguaje de Modelado Unificado (UML) en la preparación de todos los planos del sistema, debido a que tiene una semántica bien definida y que permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlás a otras personas. Esta ventaja radica en que el UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Es importante destacar que un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema.

Actualmente es la herramienta de modelado más utilizada por las empresas diseñadoras de software. También es empleada por diversas universidades, que encuentran en ella una herramienta útil para desarrollar proyectos de investigación y de ingeniería con sus estudiantes.

La universidad Francisco de Paula Santander Ocaña no es ajena a esta innovación, ya que desde hace tiempo viene implementando dicha herramienta para la creación de sus proyectos en las diferentes líneas de investigación. Es el caso de las asignaturas de proyectos I,II,III, análisis y diseño de sistemas e ingeniería de software, en las cuales se han desarrollado las diferentes etapas que conforman la metodología RUP, empleando el lenguaje UML.

Se citan algunos de los proyectos en los cuales se ha tomado como herramienta para el análisis y diseño de software el lenguaje de modelado UML:

El caso por ejemplo del diseño de un software de registro productivo bovino para la granja experimental de la UFPSO, el cual está a cargo de los estudiantes Diana Sepúlveda, Aldemar Ramírez y Alfredo Guevara y que en el momento se encuentra en la etapa de implementación y

pruebas. La interfaz gráfica del software para la selección de bovinos de la granja experimental de la UFPSO y que fue elaborado por estudiantes de proyectos, se presenta en la figura 4.

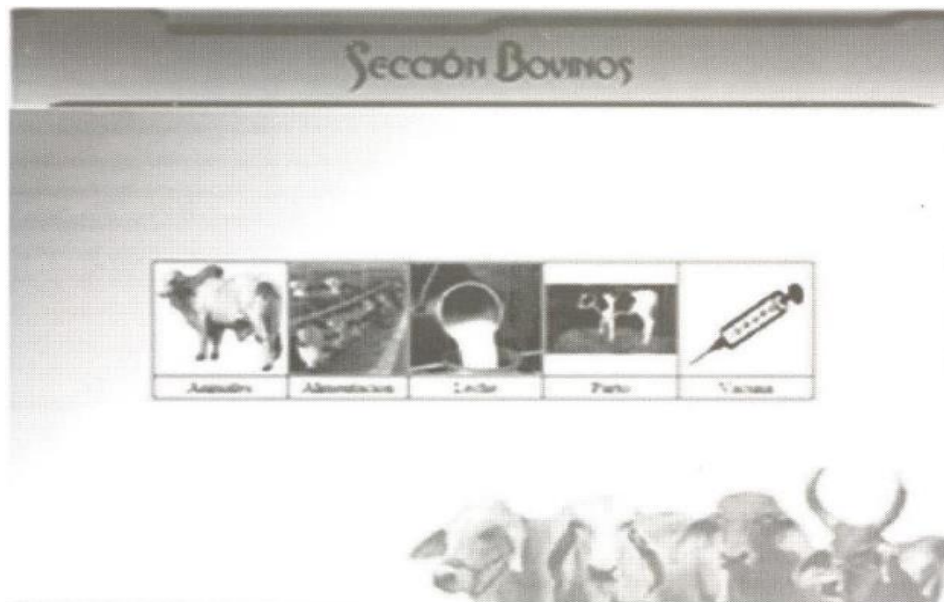


Figura 4. Ejemplo de producto: interfaz gráfica del software para la sección Bovinos de la granja experimental de la UFPSO, elaborado por estudiantes de proyectos.

Como es también el caso del estudiante Fernando Páez, el cual se encuentra en la etapa de implementación y pruebas de su proyecto de sistematización de historias clínicas y asignación de citas para una EPS.

Y el proyecto de los estudiantes, Luis Eduardo Hernández, Luis Ernesto López y Freyman Lozano, titulado CEISVIRTUAL, que consiste en la creación de un portal Web, donde se enseñará a los estudiantes todo lo relacionado con la programación Web y que en la actualidad se encuentra en la etapa de pruebas.

Así como estos, existe un gran número de proyectos que se encuentran en diferentes etapas pero, sobre todo, que han analizado y diseñado el dominio de información que les permitió tanto conocer a cabalidad las necesidades y requisitos del cliente como proponer un sistema que permitiera de alguna u otra forma hallar la solución.

En todos los proyectos de ingeniería la buena calidad se adquiere mediante un buen diseño, pero en el caso del software, la etapa de construcción incide en su calidad.

No se puede gestionar un proyecto de desarrollo de software como si se tratara de un proyecto de fabricación. Es por eso que en el desarrollo de software donde no existe una gestión adecuada podemos encontrar varios problemas tales como:

- Requerimientos incorrectos e incompletos, muchas especificaciones de requerimientos son inestables y sujetas a cambios mayores; la planificación no se lleva a cabo por la creencia errónea de que es una pérdida de tiempo y los planes cambiarán de todos modos; es difícil estimar el tamaño y complejidad del proyecto de software de modo de realizar una estimación de costos y plazos realista, no se manejan factores de riesgo, la mayoría de las organizaciones de desarrollo de software no recolectan datos de proyectos pasados, las compañías no establecen políticas o procesos de desarrollo de software.
- La gestión del proyecto de software es el primer nivel del proceso de ingeniería de software, que nos permite hacer frente a los problemas anteriormente mencionados, porque cubre todo el proceso de desarrollo.
- Para conseguir un proyecto de software fructífero, se debe comprender el ámbito del trabajo por realizar, los riesgos en los que se puede incurrir, los recursos requeridos, las tareas por llevar a cabo, el esfuerzo (costo) por consumir y el plan a seguir.
- En el presente artículo se detallan los elementos más importantes y relevantes de la gestión de proyectos de software, de modo de proveer las bases conceptuales necesarias para llevar a cabo un proyecto exitoso.

CONCLUSIONES

Se han mostrado los diferentes factores y elementos necesarios en un proceso de desarrollo de proyectos de software, elementos indispensables para obtener un producto de calidad.

Serían muchos los problemas que se presentarían al no realizar una gestión adecuada del proyecto por realizar.

Emplear una herramienta de modelado, como por ejemplo UML, garantiza la comprensión por parte, no solo del equipo de desarrollo, sino también por parte del cliente y usuario final.

BIBLIOGRAFÍA

- [1] Schach Stephen Ingeniería de software clásica y orientada a objetos (6° ed). (2006).
- [2] Bruegge, Bernard y Dutoit, Allen, Ingeniería de software orientado a objetos
- [3] Pressman, Roger. Ingeniería de software un enfoque práctico (5° ed). Prentice Hall, México, 2001
- [4] Bennet, Simon, McRobb, Steve y Farmer, Ray. Análisis y diseño orientado a objetos de sistemas, (3° ed). (2006)