
Prólogo

La ingeniería de software afecta a la economía y a las sociedades de muchas formas ya que contribuye al incremento de la productividad. Socialmente cambia la cultura del mundo debido al extendido uso de la computadora. El correo electrónico (e-mail), la WWW (World Wide Web) y la mensajería instantánea permiten a la gente interactuar en nuevas formas.

El software baja el costo, mejora la calidad de los servicios de salud, los departamentos de bomberos, las dependencias gubernamentales y otros servicios sociales. Los proyectos exitosos donde se han usado métodos de ingeniería de software incluyen a Linux son, por ejemplo, el software del trasbordador espacial, los cajeros automáticos, entre otros.

En este sentido la ingeniería aplicada al software se basa en herramientas preestablecidas de manera eficiente y óptima. No es sólo resolver problemas, sino, tener en cuenta varias soluciones y elegir la más apropiada. El objetivo desde hace tiempo, ha sido encontrar procesos o metodologías predecibles y repetibles que mejoren la productividad y la calidad.

La ingeniería de software requiere llevar a cabo muchas tareas como son: análisis de requisitos (extraer los requisitos del software a desarrollar), descripción (detallar el software a realizar), diseño (funcionamiento de forma general), programación (pasar el diseño a código), prueba (comprobar que el software realice correctamente las tareas indicadas), documentación (manual de usuario) y el mantenimiento (del software para su mejora).

Los factores más importantes que han llevado a la necesidad de tener software de calidad son: el aumento del poder de la computación en las empresas, el incremento en el número de usuarios de los sistemas, los usuarios que solicitan sistemas hechos a la medida y las nuevas tecnologías que surgen.

La complejidad del software producido y demandado, se incrementa constantemente. Las aplicaciones de software son complicadas porque modelan el mundo real. En estos días, las aplicaciones típicas son muy grandes y complejas para que un individuo las entienda, por ello, lleva gran tiempo implementar software.

En la actualidad la mayoría del software que se construye es riesgoso y difícil de controlar, muchos de éstos son construidos sin una metodología de por medio y por lo tanto el resultado son clientes insatisfechos. Esto resulta cuando no se toma en cuenta el emplear una metodología adecuada, sobre todo cuando se trata de proyectos pequeños que duran pocos meses. Lo que se hace con este tipo de proyectos es separar rápidamente en procesos, cada proceso en funciones y por cada función determinar un tiempo aproximado de desarrollo.

Cuando los proyectos que se van a desarrollar son de mayor amplitud, se debe tomar en cuenta una metodología de desarrollo buscando cuál sería la más apropiada. Lo cierto es que muchas veces no se encuentra la más adecuada y se termina por hacer o diseñar una metodología propia, algo que por supuesto no está mal, siempre y cuando cumpla con el objetivo.

Los problemas de desarrollo de software se deben en gran medida a la mala implementación de la metodología. Así, la clave para el éxito de un proyecto de desarrollo de software es la adecuada implementación de la metodología con la que se va a desarrollar, para obtener la satisfacción del usuario final.

Después de identificar las necesidades al desarrollar sistemas, se propone la metodología a utilizar, una de las mejores formas de desarrollar software es utilizar un modelo de desarrollo en espiral, este modelo está basado en repeticiones o bucles en las cuales cada repetición es llamada actividad, estas actividades no son fijas, son elegidas en función del análisis de riesgos, el cual debe comenzar por el bucle más interno. La metodología que cumple con este modelo es llamada RUP (Rational Unified Process) la cual es un proceso de desarrollo de software que 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.

RUP es una forma disciplinada de asignar tareas y responsabilidades (¿Quién hace? ¿Qué?, ¿Cuándo? y ¿Cómo?), que pretende implementar las mejores prácticas en ingeniería de software teniendo un desarrollo iterativo con una administración de requisitos, una arquitectura basada en componentes, un control de cambios, un modelado visual del software y la verificación de la calidad del software.

Se caracteriza por ser iterativo e incremental, está centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento y puede desempeñar distintos roles a lo largo del proceso).

Esta metodología divide el proceso de desarrollo en ciclos, teniendo un producto final de cada ciclo, éstos se dividen en fases:

- Fase de inicio: aquí se identifican los principales casos de uso y los riesgos.
- Fase de elaboración: es donde se completan los casos de uso y se eliminan los riesgos.
- Fase de construcción: donde se elabora un producto totalmente operativo y eficiente, además de elaborar los manuales de usuario y la fase de transición donde se implementa el producto terminado.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos. Los artefactos son los resultados parciales o finales que es producido y usado durante el proyecto, son las entradas y salidas de las actividades; éstos pueden ser un documento, un modelo o un elemento de modelo. Ésta es una de las características que hacen de RUP una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

En el siguiente trabajo se presentan las características de diversas metodologías, descripción de los procesos que éstas involucran, así como resultados, ventajas y desventajas de cada una de ellas. En el capítulo 1 se presenta el marco teórico de la ingeniería de software, las clasificaciones de las distintas metodologías de desarrollo y análisis, así como ventajas y desventajas de ellas; en el capítulo 2 se describe el proceso de cada una de estas metodologías; en el capítulo 3 se explica el paradigma orientado a objetos, las definiciones y conceptos que se involucran en éste; al llegar al capítulo 4 adquirimos los conocimientos necesarios para empezar con el modelado de sistemas, en éste se presentan los distintos diagramas que conforman el modelado mediante UML. Teniendo los conocimientos de las distintas metodologías y del modelado podemos iniciar con el diseño de sistemas de información más completos; en el capítulo 5 presentamos el proceso de diseño mediante RUP, así como todos los “artefactos” que este proceso contiene. Al llegar el término de este trabajo presentamos los resultados obtenidos mediante la aplicación de RUP y las conclusiones que se obtuvieron en la realización del presente proyecto.