

Metodologías actuales de desarrollo de software

RIVAS, Carlos Ignacio*†, CORONA, Verónica Paola, GUTIÉRREZ, José Fructuoso y HERNÁNDEZ, Lizeth

Instituto Tecnológico de Pachuca. Felipe Angeles Km. 84.5, Venta Prieta, 42083 Pachuca de Soto, Hgo., México

Recibido 5 de Julio, 2015; Aceptado 24 de Noviembre, 2015

Resumen

Las metodologías de desarrollo de software son indispensables para crear o actualizar software de calidad que cumpla con los requisitos de los usuarios; son una parte fundamental de la Ingeniería de software la cual denomina metodología a un conjunto de métodos coherentes y relacionados por unos principios comunes. El objetivo del artículo es brindarle al lector un panorama general de las que existen agrupándolas, de acuerdo a su evolución, al tipo de software por desarrollar, a la forma de generarlo y a su agilidad y prontitud para adaptarse a los cambios tecnológicos. Una contribución del artículo al conocimiento de los desarrolladores de software, se presenta en los resultados, donde se dan recomendaciones para seleccionar la metodología más apropiada. El artículo lo integran tres partes; la primera es la introducción donde se plantea la rápida evolución del software, la enorme demanda de este y la justificación de emplear metodologías de desarrollo del software. La segunda muestra un panorama general de las metodologías existentes y en la tercera están los resultados donde se dan recomendaciones para seleccionar la adecuada.

Ingeniería de software, metodologías de desarrollo de software

Abstract

The software development methodologies are essential to create or update quality software that meets the requirements of users; they are an essential part of software engineering methodology which called a coherent set of methods and related by common principles. The objective of this article is to give the reader an overview of the existing grouping them according to their evolution, the type of software to develop, and how to create agility and readiness to adapt to technological changes. Item contribution to the knowledge of software developers is presented in the results, where recommendations are given for selecting the most appropriate methodology. The article was composed of three parts; The first is the introduction where the rapidly changing software arises, the huge demand for and the justification of using software development methodologies. The second shows an overview of existing methodologies and the third are the results where recommendations are given for selecting the right.

Software engineering, software development methodologies.

Citación: RIVAS, Carlos Ignacio, CORONA, Verónica Paola, GUTIÉRREZ, José Fructuoso y HERNÁNDEZ, Lizeth. Metodologías actuales de desarrollo de software. Revista de Tecnología e Innovación 2015, 2-5: 980-986

* Correspondencia al Autor (Correo Electrónico: crivaspalacios@yahoo.com.mx)

† Investigador contribuyendo como primer autor.

Introducción

Desde hace cinco décadas (principios de los 60), la tecnología computacional e informática ha evolucionado a pasos agigantados en el hardware, que son los componentes físicos y tangibles de los sistemas de cómputo (procesador, memoria RAM, monitor, teclado, disco duro, etcétera), y más aún en el software, que es el conjunto de programas, procedimientos y documentación relacionada que asocia un sistema computacional, específicamente la parte lógica de la computadora (McIver, 2011).

Por lo que se refiere al hardware, los avances son palpables; nos damos cuenta al observar y operar las máquinas y dispositivos computacionales que están a nuestro alcance, como computadoras, tabletas, teléfonos móviles, televisores, cámaras fotográficas, computadoras de automóviles, computadoras de videojuegos y muchos otros. En 1965, y de acuerdo al vertiginoso desarrollo del hardware, el ejecutivo Gordon Moore, de la empresa fabricante de chips y microprocesadores Intel, observó que cada nuevo chip (monocristal semiconductor que contiene un circuito integrado) de procesador tenía aproximadamente el doble de la capacidad de su predecesor, y que cada nuevo chip, salía al mercado en un plazo de 18 a 24 meses; esto ahora se conoce como la ley de Moore. La tendencia de duplicar la capacidad de procesamiento de cómputo cada dos años continúa en nuestros días y es extraordinariamente precisa, además de que constituye la base para predicciones en la industria de fabricación de procesadores y sistemas computacionales (McIver, 2011).

Si la evolución del hardware es muy acelerada y la cantidad que se fabrica en la actualidad es enorme, mayor aún es la demanda de software, ya que, cada computadora y dispositivo computacional requiere muchos programas para funcionar.

Asimismo una computadora, ya sea multiusuario, de red o personal, puede atender muchos usuarios, al mismo tiempo que utilizan múltiples aplicaciones de software. De igual forma, hay infinidad de usuarios de empresas, fábricas, instituciones, negocios, gobierno y otros, que emplean software de todo tipo, ya sea empresarial, de propósito general, de propósito específico y particular según sus necesidades. También, la demanda de software crece enormemente porque se requiere para profesionistas de diversas disciplinas tales como ingenieros, actuarios, matemáticos, abogados, contadores, comunicadores, médicos, estudiantes de diferentes carreras y para actividades como la educación, la aviación, las ciencias, las finanzas, la cultura y las artes, la medicina, la astronomía, la gastronomía, la hotelería, el gobierno, el transporte, en fin la demanda de software es mucha y diversa.

Ahora bien, ¿quién desarrolla o fabrica el software? Los ingenieros en Sistemas Computacionales, los licenciados en Informática y en general los profesionales de desarrollo de software. Estos deben desarrollar software de calidad que atienda las necesidades y cumpla con los requisitos que los usuarios demandan, y que además que sea amigable, es decir, fácil de usar. Pero crear software es algo muy complejo, sobre todo cumplir con los atributos de calidad que los usuarios (personas, empresas, instituciones) requieren; debido a ello, desde que se inició la fabricación de computadoras y hubo la necesidad de programas para que funcionaran (década de los 60), surgieron también las metodologías de desarrollo de software (MDS), que es el tema central de este artículo.

Las MDS son parte esencial de la ingeniería de software (IS), que es la disciplina profesional que trata fundamentalmente de las actividades llevadas a cabo por personas que producen.

Usan o modifican artefactos de software (un artefacto es algo tangible creado con un propósito práctico) (Sánchez, 2012).

Las MDS son indispensables para crear, o modificar software de calidad que cumpla con los requisitos de los usuarios, ya que si no se utiliza la metodología apropiada, seguramente no se alcanzará el objetivo.

El problema actual es que de las diversas MDS que existen no se selecciona la adecuada, y en el peor de los casos no se emplea ninguna, para desarrollar el software que se requiere. Para dar una solución a lo anterior, en este artículo se presenta un panorama general de las metodologías que se pueden utilizar, agrupándolas por tipos de aplicaciones particulares; asimismo, a manera de resultados, se proporcionan algunas sugerencias para seleccionar la adecuada.

La metodología de investigación para obtener los resultados y conclusiones de este artículo consistió en: seleccionar un tema interesante y útil para los profesionistas, académicos y estudiantes del desarrollo de software, investigar cuáles existen, y se agruparon de acuerdo a su evolución, tipos de aplicaciones, prontitud y adaptabilidad de desarrollo, y al final se presentan los resultados de un análisis sencillo para elegir una MDS apropiada.

¿Qué son las metodologías de desarrollo de software?

Inicialmente, es importante conocer la definición de metodología y desarrollo. Metodología es una palabra compuesta por tres vocablos griegos: metá (“más allá”), odós (“camino”) y logos (“estudio”); considerando lo anterior, la definición de metodología son los métodos para luego determinar cuál es el más adecuado.

El concepto de metodología es “conjunto de métodos coherentes y relacionados por unos principios comunes”. El concepto de desarrollo, está vinculado a la acción de desarrollar o a las consecuencias de este accionar, por lo tanto es necesario, rastrear el significado del verbo desarrollar: se trata de incrementar, agrandar, extender, ampliar o aumentar alguna característica de algo físico (concreto) o intelectual (abstracto) [1]. Por lo anterior, se concluye que metodología de desarrollo es: el estudio y determinación de cuál es el método más adecuado para dar incremento a algo en este caso al software.

Actualmente el término desarrollo es el más utilizado para referirse a las actividades que involucran la creación, fabricación, actualización o modificación de software.

¿Cuáles metodologías existen y cómo se pueden agrupar?

Con base en la información de los cursos de IS impartidos por el autor, en el Instituto Tecnológico de Pachuca, la recopilada por alumnos de la materia, los textos de ingeniería de software y la investigación en internet sobre el tema, a continuación se presentan los nombres de las metodologías que existen y una forma de agruparlas.

Metodologías clásicas

De acuerdo con Pressman (2010), las MDS clásicas son llamadas también modelos de proceso prescriptivo, y fueron propuestas originalmente para poner orden en el caos del desarrollo de software que existía cuando se empezó a generar masivamente. La historia indica que estos modelos tradicionales, propuestos en la década del 60, han dado cierta estructura útil al trabajo de IS y constituyen un mapa razonablemente eficaz para los equipos de software. Estas MDS son:

- Ciclo de Vida o Cascada
- Incremental
- Evolutivo
- Espiral
- Prototipos
- Desarrollo basado en componentes

En la Figura 1 se muestra un diagrama de la metodología de Ciclo de vida o Cascada

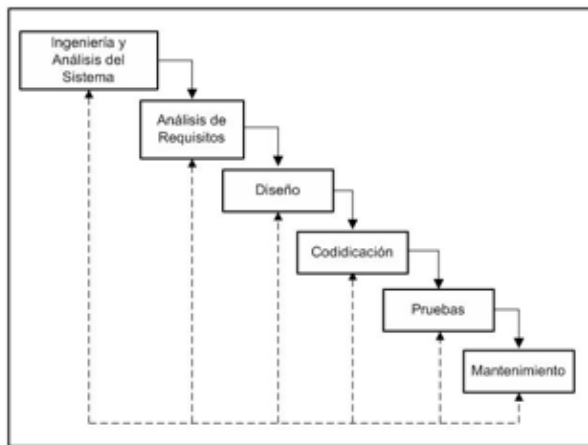


Figura 1 Diagrama de metodología Ciclo de vida o Cascada

Metodologías orientadas a objetos

En los años finales del siglo XX se publicaron centenares de artículos y libros que proponían distintas metodologías, técnicas y notaciones para el desarrollo orientado al objeto. En cuanto al proceso de desarrollo de software, se distinguen tres grandes corrientes:

Metodologías dirigidas por los datos (data-driven), que se basan en la parte estructural de los objetos y son una extensión del modelo conceptual en el modelo Entidad/Relación. Estas son:

- Fusión
- Object Modelling Technique (OMT)

Metodologías dirigidas por las responsabilidades (responsability-driven), que representan el enfoque más purista de la orientación al objeto centrándose en las “responsabilidades” de los objetos, esto es, las acciones que puede llevar a cabo un objeto. Dos de estas son:

- Object Management Facility (OMF)
- Object Management System (OMS)

Proceso de unificado de desarrollo de software (USDP Unified Software Development Process): se deriva de la metodología Objectory, de Jacobson; la metodología de Booch; y la técnica de modelado de objetos, de Rumbaugh. (Piattini, 2000).

- Unified Process (UP)

En la figura 2 se muestra un diagrama del Proceso unificado de desarrollo de software.



Figura 3 Diagrama de metodología Proceso Unificado de desarrollo de software

Metodologías ágiles

Actualmente, las empresas operan en un entorno global que cambia rápidamente; en ese sentido, deben responder a nuevas oportunidades y mercados, al cambio de las condiciones económicas así, como al surgimiento de productos y servicios nuevos y competitivos. Para ello es necesario emplear computadoras y dispositivos computacionales, por lo que el software es partícipe de casi todas las operaciones empresariales, de modo que debe desarrollarse de manera ágil para responder con oportunidad y calidad a todo lo necesario. Estas MDS son:

- Programación extrema (XP), es de las más exitosas y se considera también emergente
- Mobile-D (ágil y extrema para móviles)
- Scrum
- Crystal
- Evolutionary Project Management (Evo)
- Feature Driven Development (FDD)
- Adaptive Software Development (ASD)
- Lean Development

En la Figura 3 se muestra un diagrama de procesos de metodología ágil de desarrollo.



Figura 3 Diagrama de metodología Ágil de desarrollo

Metodologías formales

Los métodos formales son soluciones matemáticas para resolver problemas de software y hardware a nivel de requisitos, especificación y diseño.

Generalmente, se puede utilizar la teoría de autómatas para aumentar y validar el comportamiento de la aplicación diseñando un sistema de autómata finito. Los métodos formales suelen aplicarse en software de aviación, especialmente si es progradológica de seguridad crítico (Pressman, 1997).

- Red de Petri
- RAISE
- Vienna Development Method (VDM)

Metodologías para la web

El crecimiento desenfrenado que está teniendo la web está ocasionando un impacto en la sociedad, y el nuevo manejo de información en las diferentes áreas ha hecho que las personas tiendan a realizar sus actividades por esta vía. La ingeniería y las metodologías web están relacionadas con el establecimiento y utilización de principios científicos, de ingeniería y gestión, y con enfoques sistemáticos y disciplinados del éxito y desarrollo.

Empleo y mantenimiento de sistemas y aplicaciones basados en la World Wide Web de alta calidad. (Pressman, 2010). A continuación se presentan algunas MDS para web:

- Ingeniería web
- Diseño de webapps
- Método de diseño de hipermédios orientados a objetos (MDHOO)

En la figura 4 se muestra un diagrama de metodología de desarrollo para sistemas web.

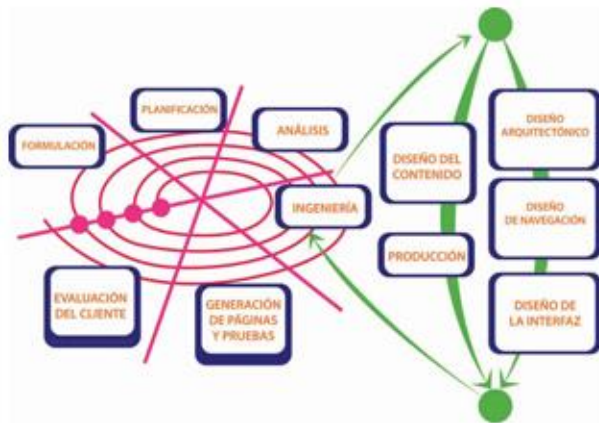


Figura 5 Metodología Ingeniería Web

Otras metodologías

Con base en las diferentes aplicaciones y tipos de software por desarrollar, otras metodologías son:

- Reingeniería
- Ganar-ganar
- Ingeniería de software distribuido
- Ingeniería para software educativo

En la figura 5 se muestra un diagrama de metodología de Reingeniería de software.

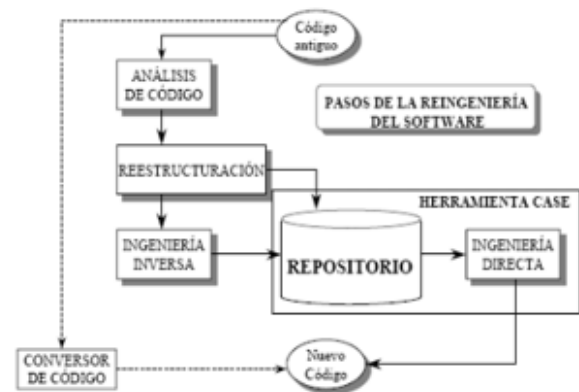


Figura 6 Diagrama de metodología de Reingeniería de software

Resultados

Los resultados de la investigación son seleccionar cuál MDS es la apropiada para desarrollar un producto de software. En el apartado anterior se observa que hay diversidad de MDS, por lo que para elegir la adecuada, debe considerarse el tipo de software a desarrollar (que puede ser de propósito general, propósito específico, educativo, tiempo real, etcétera), la premura y el entorno de globalización, los estándares que se emplean en el sitio de desarrollo (empresa consultora de software o lugar de trabajo), los dispositivos donde correrá el software (móviles) o si el software será para la web.

Por ejemplo, si el desarrollador trabaja por su cuenta, podría escoger la del Ciclo de Vida, que es la más sencilla y sigue todos los pasos formales para obtener un producto de calidad. Para el caso de productos muy grandes y complejos que requieren revisiones por etapas y que el cliente o usuario las apruebe progresivamente, se recomienda utilizar las MDS de Espiral o Evolutiva. Si es el caso de una empresa que ya cuenta con sistemas funcionales pero es necesario actualizarlos debido a nuevas tecnologías computacionales de hardware y software, entonces se recomienda la Reingeniería o el Desarrollo basado en componentes.

Si es necesario entregar los productos de software en tiempo breve y que se adapten de inmediato a los cambios de tecnologías, para lograrlo es necesario un equipo de trabajo conformado por individuos muy comprometidos, cuya capacidad y habilidad para colaborar es el fundamento para el éxito del proyecto, entonces es conveniente seleccionar MDS ágiles y extremas. Finalmente, si el objetivo es desarrollar un sistema sencillo o complejo para la web se cuenta, con metodologías altamente especializadas y apropiadas para este fin.

Conclusiones

Todas las metodologías tienen ventajas que se pueden aprovechar, dependiendo de las condiciones del software que se pretende desarrollar; de igual forma presentan desventajas cuando no se consideran todos los factores que intervienen al realizar el trabajo. Lo importante es utilizar siempre una MDS apropiada, para lo cual, si es la primera vez que se empleará, es necesario conseguir información y documentación sobre ella. En el caso de que ya se haya empleado, lo recomendable es actualizarse y adaptarla lo mejor posible para obtener un producto de calidad que cumpla con los requerimientos funcionales y no funcionales.

Referencias

McIver McHoes Ann y Flynn Ida (2011). *Sistemas Operativos*. México, CENAGE Learning. (6ª. ed.).

Piattini Mario, Calvo-Manzano José y Cervera Joaquín (2000). *Análisis y diseño detallado de aplicaciones informáticas de gestión*, México, Alfaomega Grupo Editor.

Pressman Roger S. (2010). *Ingeniería del software. Un enfoque práctico* (7ª. ed.). México: McGraw-Hill Interamericana

Sánchez Salvador, Sicilia Miguel Ángel y Rodríguez Daniel (2012). *Ingeniería del Software. Un enfoque desde la guía SWEBOK*, México, Alfaomega Grupo Editor.

Pressman, R. S. (1997). *Ingeniería del Software: Un enfoque práctico*. Mikel Angoar.

Disponible en:

<http://books.google.es>

[http://www.google.com.mx/definición\(Real Academia Española RDA\)](http://www.google.com.mx/definición(RealAcademiaEspañolaRDA))