

Justificación de las metodologías ágiles en el desarrollo software

M^a Nieves Carralero Colmenar

Profesora de Educación Secundaria

IES Pedro Mercedes. Cuenca

Un proceso de Ingeniería de Software se define como: “un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad” (Jacobson, 1999). En este trabajo se hace una introducción a los procesos clásicos para justificar la necesidad de las metodologías ágiles.

1. Procesos de desarrollo software

Un proceso de Ingeniería de Software se define como: “un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad” (Jacobson, 1999). A este proceso también se le llama el ciclo de vida del software que comprende cuatro grandes fases: concepción, elaboración, construcción y transición, esto siempre desde el punto de vista de una metodología llamada tradicional, clásica o predictiva.

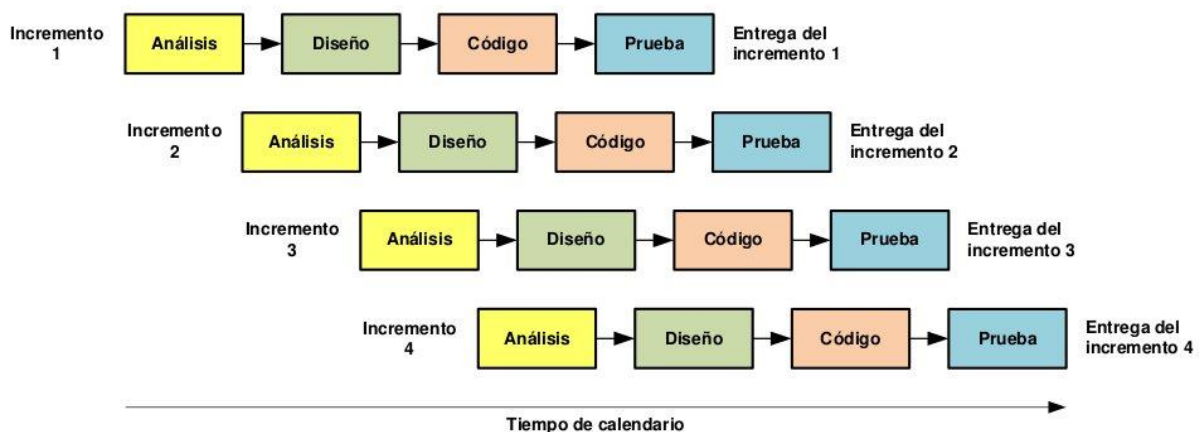
- Concepción: Define el alcance del proyecto y desarrolla un caso de negocio.
- Elaboración: Define un plan del proyecto, especifica las características y fundamenta la arquitectura.
- Construcción: Crea el producto.
- Transición: Transfiere el producto a los usuarios.

Para llevar a cabo este ciclo de vida de desarrollo software en el que se tienen definidas las fases de desarrollo, es necesario establecer unos modelos que permitan, en función de cada tipo de proyecto, utilizar aquel que más se ajuste. Aquí veremos algunos tipos de modelos de desarrollo (Pressman, 2006):

Modelo Cascada: Ciclo de vida lineal, aunque admite iteraciones. Se define como una secuencia de fases en las que al final de cada una de ellas se reúne la documentación para garantizar que cumple las especificaciones y requisitos para pasar a la fase siguiente.

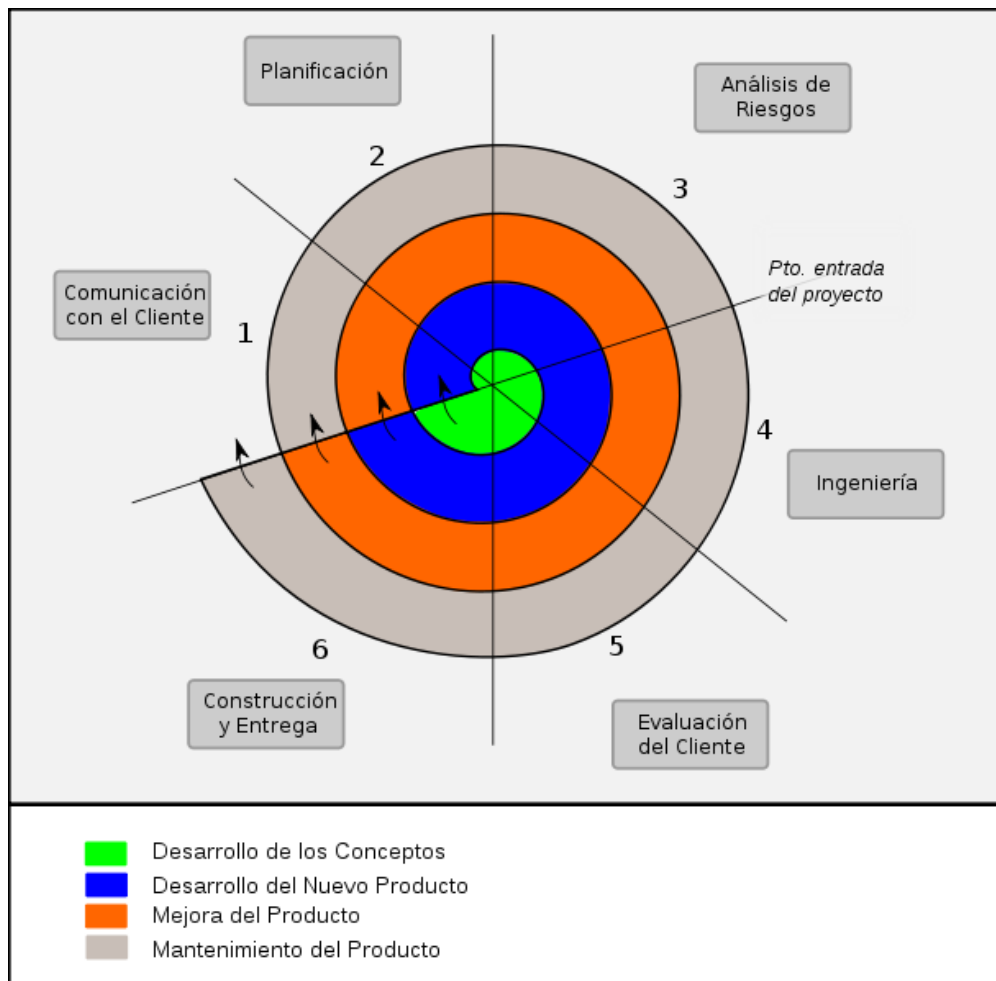
Prototipado: Se modela sobre el producto final y permite realizar un test sobre determinados atributos del mismo sin necesidad de que éste esté terminado. Puede ser utilizado en cualquier etapa de desarrollo. A medida que el proceso avanza el producto se completa, de forma que el prototipo vaya adquiriendo las características del producto final. Los prototipos tienen una doble función y es que el cliente ve el producto y refina sus requisitos, y que el desarrollador comprende mejor lo que necesita hacer. Por tanto, se asegura la reducción de costos y aumenta la probabilidad de éxito.

Iterativo e incremental: Este modelo disminuye riesgos ya que se construye a partir de un diseño preliminar, que va siendo completado con nuevos incrementos conforme el cliente va teniendo nuevos requisitos. El modelo iterativo: mejora cada versión la funcionalidad que tiene dicha versión.



El modelo incremental: mantiene la función anterior y aumenta otra, ya que puede ser que el primer incremento no hubiera tenido todos los requisitos que necesitaba el proyecto.

Modelo en Espiral: Combina la construcción de prototipos con los aspectos controlados del modelo lineal secuencial visto anteriormente. Las actividades de este modelo se conforman en una espiral, en la que cada bucle o iteración representa un conjunto de actividades, las cuales no están fijadas a priori, sino que se eligen en función del riesgo comenzando por el bucle interior. Durante las primeras iteraciones la versión incremental podría ser un modelo en papel o un prototipo. Durante las últimas iteraciones, se producen versiones cada vez más completas del sistema diseñado.



2. Tendencia actual.

A la hora de afrontar el proyecto, se ha trabajado sobre las diferentes tendencias metodológicas existentes en la actualidad, resultando especialmente interesante la evolución que expresa Juan Palacio en su libro (Palacio, 2007). En él se ofrece una visión general de la manera en la que nacen las metodologías ágiles.

La gestión de proyectos predictiva (o clásica, tal y como se indica en la sección anterior) define proyecto como: “conjunto único de actividades necesarias para producir un resultado previamente definido, en un rango de fechas determinado y con una asignación específica de recursos” (Palacio, 2007). Es decir, un proyecto sería aquel realizado por personas, ejecutado con unos recursos limitados, llevado a cabo siguiendo una estrategia de actuación, produciendo un resultado único y desarrollándose en un marco temporal pre-establecido.

El proyecto a desarrollar en este TFG no se ciñe a las características de una gestión de proyectos predictiva por las siguientes razones: en primer lugar los resultados que se desean obtener no se encuentran previamente definidos, ya que el resultado final será una consecución de cambios y adaptaciones sugeridos por el cliente. En segundo lugar, no va a ser ejecutado en base a unos recursos específicos sino que en función de las necesidades se irán adquiriendo. Y por último, no existe una única estrategia de actuación sino que a partir de los cambios demandados, se actuará de la mejor forma para resolverlos. Por tanto, este proyecto podría servir de ejemplo de que una metodología predictiva no es la más aconsejable para su desarrollo.

A menudo, las organizaciones que gestionan los proyectos de una manera predictiva tienen tan definidas e institucionalizadas las prácticas de trabajo que a veces puede ocurrir que se desvirtúe el objetivo principal de la gestión. Muchas veces el gestor del proyecto se centra tanto en cumplir los pasos establecidos que deja de lado la gestión en un enfoque amplio del propio proyecto.

Dos variables importantes han sido las que han ido moldeando el nuevo escenario de desarrollo de productos y por consiguiente aplicaciones informáticas a finales del siglo pasado: velocidad de desarrollo e incertidumbre (Kniberg, 2007).

Velocidad en el desarrollo: Se refiere a que el mercado ha cambiado y ahora las empresas facturan mucho más por productos novedosos que por productos en catálogo. En este momento un producto que sale al mercado en relativamente poco tiempo queda descatalogado (Palacio, 2007). Por ejemplo: En 6 años, Apple ha lanzado al mercado 5 modelos de iPod, y ha creado las líneas iPod mini e iPod Shuffle.

Incetidumbre: En un escenario rápido e inestable, encajan mal las estrategias de negocio predictivas, que diseñan un producto y trazan un plan de negocio. En entornos estables la

previsión es una estrategia válida, pero no es realista para entornos rápidos. La incertidumbre es una consecuencia de la velocidad. En los sectores sometidos a una evolución muy rápida, no es posible, ni aconsejable, trazar un plan de negocio para un producto en el momento de su diseño.

Debido a estas dos variables entre otros factores, surgió la necesidad de cambiar de modelo metodológico a la hora del desarrollo de proyectos en general, y más particular en el desarrollo de proyectos informáticos. Por lo que en marzo de 2001 surgió el manifiesto ágil con idea de ofrecer una alternativa a las metodologías tradicionales (Sims et al., 2011)

3. Metodologías ágiles (Kniberg, 2007).

Dynamic Systems Development Method (DSDM): Metodología ágil más veterana y la que más se aproxima a los métodos tradicionales, su implantación incluso permitiría alcanzar un nivel 2 de madurez según CMMI.

Extreme Programming (XP): La metodología ágil más radical y popular. Se centra en el ciclo de vida del desarrollo de software.

Agile Modeling: Metodología para el modelado y la generación de documentación que se encuentra alineado con los principios del desarrollo ágil. Aplicar valores, principios y prácticas “ágiles” a otras metodologías de desarrollo de software, a fin de complementarlas.

Feature Driven Development (FDD): Metodología de desarrollo de software orientada a la generación de valor para el cliente.

Scrum: probablemente la metodología más conocida actualmente, está basada en un proceso iterativo e incremental (como el descrito al principio de esta sección) con entregas constantes y ciclos de desarrollo solapados entre sí, enfocados al cliente y a los resultados.

Bibliografía

Alvarez, M. A. (2010). Desarrollo agil. <http://www.desarrolloweb.com/articulos/392.php>

Jacobson, I., Booch, G. y Rumbaugh, J. (1999). El proceso unificado de desarrollo de software. Addison-Wesley

Kniberg, H. (2007). Scrum y XP desde las trincheras. Cómo hacemos Scrum. Info-Q. Enterprise Software Development Series. <http://infoQ.com/minibooks>.

Palacio, J. (2007). En Flexibilidad con Scrum. http://www.navegapolis.net/files/Flexibilidad_con_Scrum.pdf

Pressman, R. S. (2006). Ingeniería del Software: Un enfoque Práctico. Mc- Graw-Hill.

Sims, C. y Johnson, H. L. (2011) The elements of Scrum. Dymaxicon

SOCIEDAD DE LA INFORMACION

www.sociedadelainformacion.com

Edita:



Director: José Ángel Ruiz Felipe

Jefe de publicaciones: Antero Soria Luján

D.L.: AB 293-2001

ISSN: 1578-326x