



UNIVERSIDAD NACIONAL DE ASUNCIÓN

FACULTAD POLITÉCNICA

Especialización en Tecnologías de la Información y la Comunicación

Ingeniería de Software

**Tema:**

**IMPLEMENTACIÓN DE LA METODOLOGÍA ÁGIL DE  
DESARROLLO DE SOFTWARE EN EL DEPARTAMENTO DE  
INFORMÁTICA DEL MINISTERIO DE HACIENDA.**

---

**Tutor:** Dr. Daniel Romero

**Autor:** Mario Asunción Cañiza Páez  
mario.caniza@gmail.com

Abril - 2.016

# Índice

<b>1. INTRODUCCIÓN</b>	<b>5</b>
<b>2. HIPÓTESIS</b>	<b>6</b>
<b>3. OBJETIVOS</b>	<b>7</b>
3.1. Objetivo General . . . . .	7
3.2. Objetivos Específicos . . . . .	7
<b>4. HISTORIA DE LOS PROCESOS DE DESARROLLO</b>	<b>8</b>
<b>5. DESARROLLO ÁGIL</b>	<b>9</b>
5.1. ¿Que es? . . . . .	9
5.2. ¿Quien lo hace? . . . . .	9
5.3. ¿Porque es importante? . . . . .	9
5.4. ¿Cuales son los pasos? . . . . .	9
5.5. ¿Cual es el producto obtenido? . . . . .	10
5.6. ¿Como puedo estar seguro de lo que he hecho correctamente.? . . . .	10
5.7. ¿Que es la agilidad? . . . . .	10
<b>6. METODOLOGÍAS ÁGILES</b>	<b>11</b>
<b>7. Conclusión</b>	<b>13</b>

## Índice de figuras

1.	Ciclo de entrega ágil . . . . .	12
----	---------------------------------	----

## Índice de cuadros

1.	Principios de la alianza ágil. . . . .	10
2.	Causas principales de fracasos. . . . .	11

## Resumen

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, documentación y planificación, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros muchos. Una posible mejora es incluir en los procesos de desarrollo más actividades, más artefactos y más restricciones, basándose en los puntos débiles detectados. Sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto. Otra aproximación es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software. Esta es la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Las metodologías ágiles están revolucionando la manera de producir software, y a la vez generando un amplio debate entre sus seguidores y quienes por escepticismo o convencimiento no las ven como alternativa para las metodologías tradicionales. En este trabajo se presenta resumidamente el contexto en el que surgen las metodologías ágiles, sus valores, principios y comparación con las metodologías tradicionales actualmente utilizado por el área de Desarrollo de Sistemas del Ministerio de Hacienda. Entre otras cosas se describen brevemente las principales propuestas, especialmente Programación Extrema (eXtreme Programming, XP) la metodología ágil más popular en la actualidad.

**Palabras clave:** Procesos de Software, Desarrollo incremental, Metodologías Ágiles, eXtreme Programming.

## 1. INTRODUCCIÓN

## 2. HIPÓTESIS

vamos a probar la hipo

### **3. OBJETIVOS**

#### **3.1. Objetivo General**

Implementación de la metodología de desarrollo

#### **3.2. Objetivos Específicos**

- Los productos de software es presentado fuera del plazo establecido.
- bla bla bla hola mundo



## 4. HISTORIA DE LOS PROCESOS DE DESARROLLO

Uno de los grandes pasos dados en la industria del software fue aquel en que se plasmó el denominado modelo en cascada. Dicho modelo sirvió como base para la formulación del análisis estructurado, el cual fue uno de los precursores en este camino hacia la aplicación de prácticas estandarizadas dentro de la ingeniería de software. Propuesto por Winston Royce en un controvertido paper llamado "Managing the Development of Large Software Systems" [Royce, 1970] este modelo intentaba proponer una analogía de línea de ensamblaje de manufactura para el proceso de desarrollo de software, queriendo forzar predictibilidad en una entidad como el software que como fue mencionado es algo complejo y que esta más relacionado con el desarrollo de nuevos productos.

## 5. DESARROLLO ÁGIL

En el 2001, Kent Beck y otros 16 notables desarrolladores, escritores y consultores (conocidos como la alianza ágil) firmaron el Manifiesto para el desarrollo ágil de software, el cual establecía:

Hemos descubierto mejores formas de desarrollar software al construirlo por nuestra cuenta y ayudar a otros a hacerlo. Por medio de este trabajo hemos llegado a valorar:

- A los individuos y sus interacciones sobre los procesos y las herramientas.
- Al software en funcionamiento sobre la documentación extensa.
- A la colaboración del cliente sobre la negociación del contrato.
- A la respuesta al cambio sobre el seguimiento de un plan.

Esto es, aunque los términos a la derecha tiene valor, nosotros valoramos más los aspectos de la izquierda. Un manifiesto se asocia por lo general con un movimiento político emergente: aquel

que ataca a la vieja vanguardia y sugiere un cambio revolucionario (en el mejor de los casos para mejorar).

### 5.1. ¿Que es?

La ingeniería de software ágil combina una filosofía y un conjunto de directrices de desarrollo. La filosofía busca la satisfacción del cliente y la entrega temprana del software incremental; equipos de proyecto pequeños y con alta motivación; métodos informales; un mínimo de productos de trabajo de la ingeniería del software; y una simplicidad general del desarrollo. Las directrices de desarrollo resaltan la entrega sobre el análisis y el diseño (aunque estas actividades no se descargan), y la comunicación activa y continua entre los desarrolladores y los clientes.

### 5.2. ¿Quien lo hace?

Los ingenieros de software y otros participantes del proyectos (gerentes, clientes y usuarios finales) trabajan juntos en un equipo ágil: un equipo con organización propia y que controla su propio destino. Un equipo ágil fomenta la comunicación y la colaboración entre todos los que trabajan en él.

### 5.3. ¿Porque es importante?

El ambiente moderno de los negocios ocasiona que los sistemas basados en computadoras y los productos de software estén acelerados y en cambio continuo. La ingeniería del software ágil representa una opción razonable a la ingeniería convencional para ciertas clases de software y ciertos tipos de proyectos de software. Ha demostrado su utilidad al entregar sistemas exitosos con rapidez.

### 5.4. ¿Cuales son los pasos?

El desarrollo ágil podría llamarse con mayor precisión "ingeniería del software ligera". Las actividades básicas del marco de trabajo - comunicación con el cliente, planeación, modelado, construcción, entrega y evolución - se conservan, pero estas se conforman como un conjunto mínimo

de tareas que empuja al equipo de proyecto hacia la construcción y la entrega (habrán quienes argumenten que esto se hace a costa del análisis del problema y el diseño de la solución).

### 5.5. ¿Cual es el producto obtenido?

Los clientes e ingenieros de software que han adoptado la filosofía ágil tienen la misma visión: el único producto de trabajo realmente importante es un incremento de software.<sup>en</sup> funcionamiento, el cual se entrega al cliente en una fecha prometida.

### 5.6. ¿Como puedo estar seguro de lo que he hecho correctamente.?

Si el equipo de software esta de acuerdo en que el proceso funciona y dicho equipo produce incrementos de software entregables que satisfacen al cliente.<sup>1</sup>

### 5.7. ¿Que es la agilidad?

Agilidad se ha convertido en la palabra de moda en cuanto se describe un moderno proceso de software. Cualquiera es ágil. Un equipo ágil es un equipo rápido que responde de manera apropiada a los cambios. Estos son en gran parte, la materia del desarrollo de software. Cambios en el software que se va a construir, cambios entre los miembros del equipo, cambios debidos a las nuevas tecnologías, cambios de todo tipo que pueden incidir en el producto que se construye o en el producto que crea el producto. En cualquier actividad de software se debe incluir un soporte para los cambios, esto es algo que adoptamos porque es el alma y el corazón del software. Un equipo ágil reconoce que el software lo desarrollan individuos que trabajan en equipo y que las actitudes de esta gente, y su capacidad para colaborar, son esenciales para el éxito del proyecto. De acuerdo con la visión de Jacobson, la insistencia en el cambio es el conductor primordial hacia la agilidad. Los ingenieros de software deben tener pies veloces si quieren ajustarse a los cambios rápido que describe Jacobson.

”La agilidad es dinámica, con contenido específico, ajustable al cambio de manera dinámica y orientada al crecimiento.”

**Steven Goldman et al.**

La alianza ágil define 12 principios para quienes quieren alcanzar la agilidad:

Ítem	Principios
1	Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software
2	Bienvenidos los requisitos cambiantes, incluso es fases tardías del desarrollo. La estructura de los pro
3	París

Cuadro 1: Principios de la alianza ágil.

<sup>1</sup>Ingeniería del Software. Roger Pressman. Página 78, 79

## 6. METODOLOGÍAS ÁGILES

En febrero de 2001, tras una reunión celebrada en Utah - EEUU, nace el término “ágil” aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas.

Tras esta reunión se creó The Agile Alliance <sup>3</sup>, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida es fue el Manifiesto Ágil, un documento que resume la filosofía “ágil”.

Según el informe de Standish, las diez causas principales de los fracasos, por orden de importancia, son:

Ítem	Causas de fracaso
1	Escasa participación de los usuarios
2	Requerimientos y especificaciones incompletas
3	Cambios frecuentes en los requerimientos y especificaciones
4	Falta de soporte ejecutivo
5	Incompetencia tecnológica
6	Falta de recursos
7	Expectativas no realistas
8	Objetivos poco claros
9	Cronogramas irreales
10	Nuevas tecnologías

Cuadro 2: Causas principales de fracasos.



Figura 1: Ciclo de entrega ágil

## 7. Conclusión

bla  
bla  
lba

en fin ya damos por concluida este trabajo gracias totales!!!

## Referencias

- [1] PRESSMAN, Roger, *Ingeniería del software un enfoque práctico*, México. Mac Graw Hill. 5ta. Edición, (2002)
- [2] SOMMERVILLE, Ian. Ingeniería del Software. Madrid. Person Educación, S.A. Séptima Edición, (2005)
- [3] Letelier, P., Proyecto Docente e Investigador, DSIC, 2003.
- [4] Blé Jurado, Carlos., Diseño Ágil con TDD, SEFE Creative, Primera Edición, 2010.