

# Adaptacion o Muerte del Proceso Unificado

Rodrigo Rivera

Junio de 2014

## Abstract

El presente trabajo habla acerca de el uso de los procesos ágiles, que de cierta forma están dejando atrás a uno de los procesos mas tradicionales y conocidos como es el Proceso Unificado, enfatizado en las características que convierten a los procesos ágiles como los mas usados por los equipos de desarrollo en la actualidad

## 1 Introducción

En este artículo, se presentara una comparación entre procesos de desarrollo de software, mostrando características de los mismos; comenzaremos hablando de el Proceso Unificado, el primer libro que habla acerca de este proceso, fue publicado en 1999 por Ivar Jacobson, Grady Booch y James Rumbaugh, que llevo por título "El Proceso Unificado de Desarrollo de Software"; las características principales del mismo son:

- Dirigido por casos de uso, los casos de uso son utilizados para capturar los requerimientos del cliente, y son estos los que definen que se desarrollara.
- Iterativo Incremental, toma un conjunto de casos de uso y se los desarrolla, una vez concluido, se toma otro grupo, y así hasta cubrir todos los casos de uso.
- Centrado en la arquitectura, UP asume que no existe un único modelo que cubra todos los aspectos del sistema, por ello, realiza varias vistas(modelos) del mismo.
- Utiliza el Lenguaje Unificado de Modelado

Fue este proceso, catalogado como muy riguroso, lo que le dio la certificación equivalente al ISO 9000, además de permitir conocer y ayudar a entender a aquellas personas que no conocen mucho de área del desarrollo del software el como se podría planificar proyectos grandes, dando tareas bastante específicas a los desarrolladores, haciendo creer que programar es cuestión de seguir instrucciones.

It fit perfectly with corporate and government guilt to become CMM certified and provided what appeared to be the software equivalent of TQM/Six Sigma and ISO 9000. At last there was a prescriptive process to build software. All the developers had to do was follow instructions! If only software development was so simple! [1]

Ahora toca el turno de los procesos denominados "Agiles", del primero que se hablara es de SCRUM, este proceso fue aplicado y documentado por primera vez por Jeff Sutherland, John Scummiontales y Jeff McKenna en 1993, y fue en 1995 cuando fue formalizado por Ken Schwaber; este proceso tiene las siguientes características:

- Define roles para cada "personaje" involucrado en el desarrollo de software, "stake holder"(cliente), "product owner"(contratista que posee contacto con el stake holder y el equipo de desarrollo), "scrum master"(líder del equipo), y el equipo de desarrollo.
- Maneja Historias de Usuario, las cuales son los requerimientos del cliente(o stakeholders), además que posee dos valores, el "User Value"(el pago que hará el stake holder por la culminación de esa historia) y una complejidad(asignado por el equipo).
- Es iterativo incremental, se divide en partes, cuando se completa una de esas partes, se pasa a la siguiente.
- El equipo es autogestionable, es decir el equipo toma las decisiones(define el valor de complejidad).
- Presenta "sprints"(lapso de tiempo en el cual cada programador da todo de si para desarrollar una o mas historias de usuario).

## 2 Desarrollo

Como estos procesos existen varios mas, pero dado que este articulo solo se cubrirá estos dos tipos de procesos, que si bien tienen cosas en común, también tienen características que los hace totalmente diferente; existen tantos procesos como tipos de programadores existen, por ello es que existe un proceso de desarrollo de software adecuado para cada uno, algunos se sienten muy cómodos siguiendo ordenes, bajo un patrón muy rígido, y que da la idea de "control total" de lo que pasa, esta idea brinda confianza, y por lo tanto crea un ambiente de trabajo mucho mas cómodo para este tipo de programadores; por otro lado, existen otros programadores, que no se sienten cómodos con esta forma de trabajo, puesto que tiende a limitarlos de cierta forma, además que bajo un esquema desarrollado por "expertos" en el área de planificación que poco o nada conocen del desarrollo de software, puede ser muy peligroso, puesto que podrían darte plazos muy cortos para tareas grandes, o plazos grandes para tareas pequeñas, este uno de los problemas de Proceso Unificado, y aquí cito el comentario de un docente:

Cuando trabajaba en un proyecto, teníamos un encargado, y cada día teníamos una reunión, donde se le preguntaba a cada uno los avances de las tareas asignadas a cada miembro del grupo de trabajo, la mayoría siempre cumplía con las tareas del día; pero en mi caso era todo lo contrario, puesto que mis tareas requerían de mayor tiempo, y aunque trabajase todo el día, me era muy complicado cubrir mi cuota del día. [2]

Por esta clase de situaciones es que los procesos ágiles son bastante apreciados por los programadores, puesto que ellos son tomados en cuenta a la hora

de poner plazos de entrega, claro estos no pueden ser excesivos, además que tienen mayor contacto con el cliente (o al menos eso es lo ideal) y esto permite que el software sea producido con mayor velocidad, sin contar que toda la documentación que se genera o se debe tener para comenzar a programar es mínima, y no es tan rigurosa en su forma de ser escrita. Este último punto de la documentación también tiene ciertos casos puede llegar a ser una desventaja para los procesos ágiles, puesto que muchas empresas que no son desarrolladoras de software deben de justificar gastos de manera muy clara con documentación rigurosa para el uso de fondos económicos (el caso de una alcaldía por ejemplo).

Este problema era bastante común cuando antes las empresas tenían su propio equipo de "tecnología" los cuales estaban a cargo de todo lo relacionado al software; hoy en día, más y más son las empresas que se dedican netamente al desarrollo de software, haciendo que estos equipos de "tecnología" se vayan reduciendo y que pasen a ser consultores a la hora de adquirir algún programa de estas empresas de software. Casos como estos, son prueba clara de la tendencia actual de optar por servicios, principalmente porque te olvidas de todos los detalles técnicos del cómo se hacen las cosas, puesto que se te garantiza que lo que se va a utilizar es lo mejor, y uno ya no se preocupa de esos detalles; por ejemplo el uso de la "computación en la nube", permite que ya no sea necesario instalar un software en una computadora para que este funcione, solo se necesita una buena conexión a internet para y alguna página que provea ese servicio (el software requerido).

Esta tendencia de los servicios permite que aparezcan empresas especializadas en brindar un tipo de servicio, este es el caso de las empresas de desarrollo de software; en nuestro medio existen empresas como Jala Soft, Code Road, True Extend, entre otras, que solo se dedican al desarrollo de software; uno contrata a una para que desarrolle la aplicación que se desea, y estos te la entregaran, uno deberá pagar por la aplicación, y dependiendo del contrato, podrían brindar mantenimiento una vez terminada la misma. En muchas de estas empresas utilizan procesos ágiles de desarrollo, esto debido a que para la parte organizativa es bastante buena, muchos programadores se sienten cómodos en este esquema, motivos por los cuales la política de la empresa es utilizar este tipo de procesos. Algunas de las empresas más importantes que utilizan procesos ágiles como SCRUM son las que se pueden ver en la figura [1].

Entonces, si los procesos ágiles son tan utilizados, y queridos por los programadores, ¿qué pasa con el Proceso Unificado de Jacobsson, Booch y Rumbaugh? La respuesta es sencilla, si quiere más cabida en el mercado de trabajo, debe de transformarse, y Jacobsson ya encontró una forma, a este "cambio" o "mejora" le denomina ESSENTIAL UP, este proceso, se queda con las características esenciales del Proceso Unificado, entre ellas se destacan las siguientes:

- Permite el uso de varias prácticas de procesos y sus artefactos asociados para su descripción, organización y comunicación
- Se enfoca en la ingeniería social basado en los procesos de desarrollo ágil, que promueven las prácticas ágiles en esta nueva propuesta de desarrollo.

Sectores	Ejemplos de empresas que utilizan metodologías ágiles como Scrum
Media y Telcos	BBC, BellSouth, British Telecom, DoubleYou, Motorola, Nokia, Palm, Qualcomm, Schibsted, Sony/Ericsson, Telefonica I+D, TeleAtlas, Verizon
Software, Hardware	Adobe, Autentia, Biko2, Central Desktop, Citrix, Gailén, IBM, Intel, Microfocus, Microsoft, Novell, OpenView Labs, Plain Concepts, Primavera, Proyectalis, Softhouse, Valtech, VersionOne.
Internet	Amazon, Google, mySpace, Yahoo
ERP	SAP
Banca e Inversión	Bank of America, Barclays Global Investors, Key Bank, Merrill Lynch
Sanidad y Salud	Patientkeeper, Philips Medical
Defensa y Aeroespacial	Boeing, General Dynamics, Lockheed Martin
Juegos	Blizzard, High Moon Studios, Crytek, Ubisoft, Electronic Arts
Otros	3M, Bose, GE, UOC, Ferrari

Figure 1: Empresas que utilizan Scrum

- Proporciona referencias a las obras originales de la fundación de la comunidad de software, esto para mostrar que los profesionales tienen conocimiento esencial para poder llevar a cabo las prácticas de desarrollo de software

Al ver un poco estas características se observa que tiene una gran influencia de los procesos ágiles, sobre todo la segunda característica, la cual habla de la ingeniería social, hace referencia a la interacción entre sus componentes, es decir a la relación entre programadores, en el caso de los procesos ágiles, los desarrolladores poseen bastante contacto entre ellos, son realmente un grupo; caso contrario ocurre en el proceso unificado, puesto que cada uno tiene una tarea y se encarga de hacerlo bien, luego unirlos, pero no conversan(o al menos no es parte esencial del proceso) entre ellos. Esta práctica de los procesos ágiles, es lo que permite mejorar el software y el tiempo de producción del mismo, pues en sus reuniones diarias(stand-up meeting) en las que hablan sobre el estado del proyecto, problemas que se presentaron y hasta donde se abarcara durante la jornada, les permite solucionar el o los problemas presentados basados en la experiencia de los integrantes; lo que facilita el desarrollo de las actividades.

Al estar influenciado por procesos ágiles de cierta forma muestra que estos procesos están marcando una tendencia, al ver la gráfica de todas las empresas que utilizan o han utilizado procesos de desarrollo ágil, muestra que de cierta

forma esta desplazando a el proceso unificado, y de solo pensar que uno de los autores de este proceso, tiende a utilizar parte de la esencia o teoría de los procesos ágiles, para tener una propuesta nueva basada en el proceso unificado es porque estos están funcionando de una manera adecuada.

Ademas de dichos motivos por los que el Proceso Unificado debe evolucionar, es debido a que fue influenciado por otras áreas, no propias de la programación, lo que hace que sea tan riguroso en cuanto a toda la documentación que es necesario hacer previo empezar a programar,

The Rational Unified Process incorporates material in the areas of data engineering, business modeling, project management, and configuration management. and a detailed test process inherited from SQA..[5]

Software quality assurance (SQA) consists of a means of monitoring the software engineering processes and methods used to ensure quality. The methods by which this is accomplished are many and varied, and may include ensuring conformance to one or more standards, such as ISO 9000 [3]

ISO 9000 is a series of standards, developed and published by the International Organization for Standardization (ISO), that define, establish, and maintain a quality assurance system for manufacturing and service industries [4]

Lo que se trata de demostrar con las tres citas previas es ver como el Proceso Unificado esta influenciado por áreas que mucha relación con el desarrollo de software no posee; con esto queda claro que al existir influencia de otras áreas, pues no es totalmente este proceso, por lo cual, suele ser bastante incomodo para muchos programadores. Ademas se comprende porque es necesario tanto papeleo para poder comenzar a programar, puesto que en muchas otras profesiones se requiere hacer todo ese papeleo y justificativo; pasar por muchos puntos de control para ir avalando poco a poco el proyecto, y suele tomar bastante tiempo contar con el visto bueno del encargado del punto de control.

En el caso de los procesos ágiles, si bien parecería que no existiese todo ese control, existe, solo que es mas rápido(calvez a eso se debe el nombre ágil) pues lo mas importante para los programadores y el cliente es contar con el programa funcionando; estos vistos buenos se van dando en las reuniones diarias, cuando se va a desarrollar otra parte del proyecto, cuando se realiza un bosquejo del como será, en ese momento será validado o no, e inmediatamente corregido, de esa manera es como debería de funcionar un proyecto que se realice con un proceso ágil.

### 3 Conclusión

Claro uno no debe sacar una conclusión tan rápidamente y decir "los procesos ágiles son mejores", pues esto es algo subjetivo, y también tiene su costo, puesto que al menos una empresa en Cochabamba de desarrollo de software aplica practicas ágiles, es decir partes de lo que es el desarrollo ágil, como ser los "stand-up meetings", y en ciertas ocasiones aplican todo el proceso en un proyecto, y con esto me refiero a todo lo que implica la aplicación de un proceso ágil. El porque solo se utilizan practicas ágiles, pues debe ser complicado ser el dueño de la empresa y confiar al 100 % en tus empleados, por ello necesitas de algo o alguien que de cierta forma ponga presión en ellos para que cumplan con sus tareas, y esta es sin duda uno de los motivos por los que es tan difícil ver una empresa de desarrollo de software en la que cada proyecto se emplee un proceso de desarrollo ágil "puro"

Finalmente, al observar características de diferentes procesos de desarrollo de software, mostrar como esta cambiando el desarrollo del mismo y las preferencias de las empresas a la hora de optar por el tipo de desarrollo, sin duda alguna los procesos ágiles son los preferidos, por ser mas propios del desarrollador, puesto que no posee influencia de otros profesionales o áreas, sin tanta rigurosidad de documentación o hacer grandes ceremonias para avalar un esquema y aunque no se los aplique al 100% están predominando. Es debido a esto que el proceso unificado debe adaptarse(ESSENTIAL UP) o desaparecer conforme pase el tiempo.

Finalizo con lo siguiente sacado de un libro muy interesante,

No uses la frase "I don't have enough time.", use la frase "I HAVE TOO MUCH TO DO" [6]

pues al utilizar la segunda, priorizas tus tareas, y eliges cuales no hacer, ademas de hacer todo en el mejor tiempo posible, con la primera solo te excusas para no hacer ninguna.

### 4 Bibliografia

#### References

- [1] BEDARRA RESEARCH LABS (2003) *The Essential Unified Process(EssUP)-New Life for the Unified Process*
- [2] este punto fue concedido (Lic. Flores Soliz Juan Marcelo, entrevista personal)
- [3] [en.wikipedia.org/wiki/Software\\_quality\\_assurance](http://en.wikipedia.org/wiki/Software_quality_assurance)
- [4] [en.wikipedia.org/wiki/ISO\\_9000](http://en.wikipedia.org/wiki/ISO_9000)
- [5] [www.informit.com/articles/article.aspx?p=169549&seqNum=5](http://www.informit.com/articles/article.aspx?p=169549&seqNum=5)

- [6] KENT BECK, MARTIN FOWLER (2000) *Planning Extreme Programming*  
Addison Wesley
- [7] JAMES RUMBAUGH, IVAR JACOBSON, GRADY BOOCH *El Proceso Unificado de Desarrollo de Software*, Addison Weley
- [8] [www.slideshare.net/Sofylutqm/el-proceso-unificado-3943047](http://www.slideshare.net/Sofylutqm/el-proceso-unificado-3943047)
- [9] [es.wikipedia.org](http://es.wikipedia.org)
- [10] [www.proyectosagiles.org/historia-de-scrum](http://www.proyectosagiles.org/historia-de-scrum)