

El modelado de requerimientos en las metodologías ágiles

Silvia Rivadeneira¹, Gabriela Vilanova², María Miranda², Diana Cruz¹
Departamento Ciencias Exactas y Naturales

¹Unidad Académica Río Turbio, ²Unidad Académica Caleta Olivia,
Universidad Nacional de la Patagonia Austral
grivadeneira@uart.unpa.edu.ar, vilanova@uolsinectis.com.ar, dianalrcruz@gmail.com

Resumen

Las metodologías ágiles surgidas a fines de los '90, no han sido muy tenidas en cuenta por el mundo académico, quizás recién a mediados de la década anterior se comienzan a encontrar trabajos de investigación al respecto.

Nos enfocamos en uno de los trabajos realizados en el marco de la investigación en el que se estudiaron seis metodologías ágiles para analizar que prácticas y artefactos implementaban en el proceso de desarrollo de proyectos de software, específicamente en las actividades de ingeniería de requerimientos. Aquí se rescatan las metodologías XP, SCRUM y APM como las únicas que abarcan estas etapas, solas o en combinación con otras tradicionales o actuales.

Palabras clave: modelado de requerimientos, gestión de requerimientos, metodologías ágiles

Contexto

La Universidad Nacional de la Patagonia Austral (UNPA) está ubicada en la provincia de Santa Cruz y, está compuesta por cuatro unidades académicas dispuestas en las localidades de: Caleta Olivia, San Julián, Río Gallegos y Río Turbio.

Este artículo pertenece al Proyecto de Investigación 29/B134 denominado "Modelado de Requerimientos y Diseño de Sistemas Complejos" radicado en la Unidad Académica Caleta Olivia (UACO). El proyecto es del tipo 1 de Ciencia y Tecnología y responde a temáticas relacionadas con carrera de pre-grado, grado y postgrado dictadas en UNPA y comprende actividades de investigación básica y aplicada.

El proyecto de investigación es financiado parcialmente por UNPA.

El grupo de investigación conformado es nuevo en la UNPA y sus integrantes residen en varias localidades patagónicas, tales como Comodoro Rivadavia, Caleta Olivia, Río Turbio y Punta Arenas (Chile).

Introducción

Las actividades de elicitación, documentación y verificación de requerimientos forman parte de la ingeniería de requerimientos y permiten conocer la realidad de una organización, pero es la inestabilidad o cambio de estos

requerimientos lo que complejiza el desarrollo de un proyecto, por lo que, aún hoy, buscando una gestión más adecuada de esos cambios, se sigue investigando sobre los métodos, técnicas y estrategias que permitan alcanzar resultados positivos en el desarrollo de software.

En este artículo se analizan las metodologías ágiles más utilizadas enfocándose en la etapa de gestión de requerimientos dentro del ciclo de vida del software, donde se explicitan las prácticas, técnicas y artefactos que se utilizan en la mencionada etapa. Para ello se indagó en la literatura existente para conocer cada una de las metodologías involucradas, logrando una primera selección y comparación de las mismas.

1. Acercándonos a las metodologías ágiles

Las denominadas metodologías ágiles surgen en la década de los '90 y a comienzos del 2001 se reúnen sus representantes para discutir nuevos métodos, crear *The Agile Alliance* [2] y producir el Manifiesto por el Desarrollo Ágil del Software [1]. Si bien, el manifiesto ágil define una serie de principios y valores que guían al equipo del proyecto de software, comúnmente no se cuenta con especificaciones concretas relacionadas con aspectos metodológicos clave cubiertos [13]. En [11] se expresa que existen cuatro metodologías (XP, Scrum, DSDM¹ y FDD²) que se destacan por proveer una guía concreta, en su especificación, respecto de los procesos, prácticas, roles y productos. Pero en lo que respecta a las etapas de inicio del proyecto y especificación de requerimientos [9] XP y Scrum son las que proporcionan procesos, prácticas, actividades y artefactos específicos, así como la gestión del proyecto, y en este caso también podríamos agregar APM³.

Entre las características del enfoque ágil podemos expresar que impulsan generalmente una gestión de proyectos que promueve el trabajo en equipo, la organización y responsabilidad propia, un grupo de buenas prácticas de ingeniería de software que brindan una entrega rápida de software de alta calidad, y un enfoque de negocios que alinea el desarrollo con las necesidades del cliente y los objetivos de la empresa [10]. El ciclo de desarrollo que aplican las metodologías ágiles es iterativo e

¹ Dynamic System Development Methods

² Feature-Driven Development

³ Agile Project Management

incremental, lo que permite entregar el software en pequeñas partes denominadas incrementos. Cada iteración se puede considerar como un sub-proyecto en el que las actividades de análisis de requerimientos, diseño, implementación y testing son llevadas a cabo con el fin de producir un subconjunto del sistema final. El proceso se repite varias veces produciendo un nuevo incremento en cada ciclo hasta que se elabora el producto completo. Aunque todas las metodologías ágiles adoptan este ciclo, cada una presenta sus propias características [16].

Entre las metodologías ágiles que cubren las primeras etapas de un proyecto de software podemos describir:

- **XP** [17] el proceso está constituido por seis fases (exploración, planificación, iteraciones, producción, mantenimiento, cierre de proyecto). Los roles son reducidos, un cliente, programadores, verificadores, consultores técnicos, un consejero, un rastreador y el Gran Jefe. Algunas de sus prácticas son: entregas pequeñas y frecuentes, diseño simple, prueba continua, refactorización continua, programación de a pares, integración continua, todo el equipo radica en el mismo lugar.
- **Scrum** [18] está indicado para proyectos en entornos complejos, donde se necesitan rápidos resultados y los requerimientos son altamente cambiantes o poco definidos. Es adaptativo, ágil, auto-organizado y con pocos tiempos muertos [12]. Fue concebido para utilizar en combinación con otras metodologías. Su proceso se caracteriza por sprints o iteraciones de un mes. El resultado del sprint es un incremento ejecutable que se muestra al cliente. Otra característica son las reuniones diarias que no llevan más de 15 minutos y su objetivo es coordinar e integrar el producto a entregar. El proceso se compone de 4 fases (pre-juego – planeamiento, pre-juego – montaje, juego y post-juego), 6 roles (scrum master, propietario, equipo, cliente, gestor y usuario), dispone de prácticas y herramientas para la gestión de sus fases tales como: product backlog, sprint backlog, estimación de esfuerzos, gráfico burn-down, gráfico burn-up, planning poker [9].
- **APM** [19] el proceso se compone de 5 fases (previsión, especulación, exploración, adaptación y cierre). Los roles que podemos encontrar son: patrocinador ejecutivo, gestor de proyectos, gestor de producto, ingeniero jefe, gestores, equipo de cliente, equipo de proyecto, proveedores y gobierno. Proporciona prácticas para cada

fase, tales como: caja de producto, hoja de datos de proyecto, listado de características de producto, gestión de carga de trabajo, reuniones diarias, decisiones consensuadas y participativas, revisión de adaptación del producto, proyecto y equipo. Al igual que Scrum es un marco de trabajo, y permite la combinación con otros enfoques de gestión de proyectos más formales.

2. El proceso de requerimientos y las prácticas ágiles

En la búsqueda por mejorar el proceso de gestión de requerimientos nuestro trabajo se centró en las actividades que se realizan en esa etapa y para ello analizamos que prácticas ágiles y artefactos suelen aplicarse en las metodologías antes mencionadas.

Historia de Usuario	
Número: 1	Nombre: Ingreso o supresión de Roles
Usuario: Administrador	
Modificación de Historia Número:	Iteración Asignada:
Prioridad en Negocio: Baja (Alta / Medio / Baja)	Puntos Estimados:
Riesgo en Desarrollo: (Alto / Medio / Bajo)	Puntos Reales:
Descripción: En la administración del sistema tendrá la opción de administrar usuarios, al ingresar a esta opción se desplegará un listado de los usuarios, los usuarios van a tener la opción de asignar roles, el administrador hace clic sobre esta opción relacionada con el usuario y el sistema le despliega el listado de roles disponibles para que el administrador seleccione los adecuados para ese usuario. Una vez el usuario administrador del sistema de la opción de guardar, el sistema pide confirmación y luego procederá a almacenar los cambios.	
Observaciones:	

Figura 1. Historia de usuario.

XP demanda comunicación oral tanto para los requerimientos como para el diseño [12]. Como puede observarse en la Figura 1, las historias de cliente, tarjetas de historia (story cards), o historias de usuario son tarjetas simples en papel donde se describen breves requerimientos similares a las tarjetas CRC, con una granularidad de 10 a 20 días. Se utilizan para estimar prioridades, alcances y tiempo de realización. Si existen discrepancias se utiliza la más optimista. Otros artefactos son lista de tareas en papel o pizarra y gráficos visibles en pared [9]. Una práctica que los equipos están utilizando es SDD⁴ [15] como metodología para gestionar los requerimientos en proyectos ágiles.

Product Backlog	
	Fecha:
	Estimado:
Tipo: Nuevo ___ Mejora ___ Arreglo ___	Fuente:
Descripción:	
Notas:	

Tabla 1. Product Backlog.

Scrum no se ocupa de los detalles relacionados con el modelado de requerimientos, pero claramente el artefacto Product Backlog recoge los requerimientos del sistema y sus estimaciones. Los desarrolladores no pueden implementar los requerimientos si no están en el Product Backlog (ver Tabla 1). En cada iteración, el equipo de desarrollo comienza con la lista priorizada de

⁴ Storytest-Driven Development

requerimientos almacenada en el Sprint Backlog. Luego, cuando los requerimientos surgen y evolucionan debido al entorno dinámico del negocio o una solicitud de cambio de las partes interesadas, la modificación de las prioridades de los requerimientos en el product backlog se produce sobre la base actual del valor de los requerimientos de negocio [24].

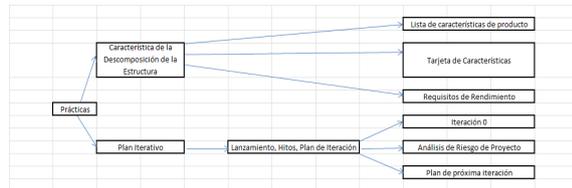


Figura 2. Prácticas de la Fase de Previsión.

En APM [19] la fase de previsión define la visión del producto, los objetivos del proyecto, la comunidad del proyecto y el equipo que trabajará. En la fase de especulación el equipo se ocupa de generar el plan de entregas basado en funcionalidades utilizando información de la especificación del producto, plataforma de arquitectura, recursos, análisis de riesgo, niveles de defecto, restricciones de negocio y fechas objetivo. Las prácticas ágiles que se utilizan se muestran en las Figuras 2 y 3.



Figura 3. Prácticas de la Fase de Especulación.

Las metodologías analizadas se pueden combinar con otras que sean tradicionales o más actuales. Estas combinaciones pueden darse en una o en todas las fases para lograr cubrir todo el ciclo de vida. Las metodologías que cubren las etapas de requerimientos, suelen hacerlo en una o dos fases, en la práctica para obtener una visión general del sistema se trabajan con los requerimientos de más alto nivel, tanto funcionales como no funcionales, a través del consenso entre el equipo de desarrollo y los clientes; y en la especificación logran la especificación de requerimientos generales, refinándolos concurrentemente hasta llevarlos a especificaciones más detalladas [9].

Para los autores [12], [14] y [26] la documentación de los procesos ágiles es informal, prefiriéndose la comunicación cara a cara tratando de reducir los costos de mantenimiento.

En [14] y [25] se sugieren prácticas para aplicar a la gestión de requerimientos:

- Asegurar la participación del cliente. Negociar los requerimientos explícitos.
- Definir la trazabilidad de los requerimientos.

- Gestionar los cambios de requerimientos.
- Iterar-Inspeccionar-Adaptar.
- Construir prototipos y demostraciones.
- Realizar talleres de colaboración.
- Incorporar conceptos orientados a aspectos coincidiendo con [27].

Líneas de investigación y desarrollo

Las líneas de investigación del grupo están centradas en las prácticas y artefactos para el análisis de requerimientos y el diseño de soluciones software, y en particular se están trabajando las siguientes:

- Modelado de requerimientos combinando BPMN y UML
- Modelado de requerimientos aplicando metodologías ágiles
- Integración de BPMN y SOA
- Arquitectura de Software y diseño de sistemas complejos

Por otro lado, no por ello menos importante, nos interesamos en la construcción del conocimiento en red, las siguientes líneas son las que involucra:

- La interacción que se produce en entornos colaborativos
- La construcción colaborativa del conocimiento

Resultados y objetivos

Del análisis de literatura existente, encuestas realizadas por investigadores, experiencias exitosas, asociación a comunidades de desarrolladores ágiles locales y regionales, hemos obtenido información de las metodologías ágiles vigentes. En [9] hicimos una primera selección para analizar en detalle, teniendo en cuenta criterios como: (a) la metodología más presente en Internet, (b) la metodología más documentada, (c) la metodología con certificaciones y entrenamiento, (d) la metodología con comunidad propia, (e) la metodología más utilizada por organizaciones y (f) la metodología más utilizada en proyectos software; esto dio como resultado la selección de:

- Agile Project Management
- Crystal Methods
- Dynamic System Development Methods
- Extreme Programming
- Scrum
- Test Driven Development

De este listado surgieron aquellas que abarcaban las etapas iniciales de un proyecto, con prácticas y artefactos definidos y que, por supuesto, podríamos aplicar eran APM, XP y Scrum.

Existen trabajos que evalúan metodologías cualitativamente tales como [20], [21], [22], [23] y [24], otros lo hacen cuantitativamente [16] y en este caso evalúan si las metodologías ágiles cumplen o no con los principios del Manifiesto. Nosotros

reconocemos que seleccionar una metodología ágil a utilizar en un proyecto [3] requiere considerar cinco dimensiones de riesgo: tamaño, criticidad, dinamismo, equipo y cultura, y esto sin tener en cuenta el tipo de desarrollo.

Hemos generado una encuesta utilizando tecnología Google Docs que colocamos en línea durante cuatro meses y que difundimos en redes sociales, comunidades de desarrolladores ágiles y a través de correo electrónico pero no logramos obtener una muestra representativa (20 respuestas), es así que decidimos analizar otras encuestas, algunos de los resultados observados [4][5][6][7][8] muestran que:

- Los equipos ágiles están utilizando XP, Scrum, sus variantes y FDD.
- Las prácticas ágiles más utilizadas son las reuniones diarias.
- Las principales razones para implantar la agilidad son: incrementar el time-to-market, gestionar los cambios de prioridades y alinear el negocio e IT.
- Las prácticas menos utilizadas son: BDD⁵, la entrega continua y la programación por pares.
- Los procesos ágiles son apropiados para una gran proporción de proyectos.
- Las prácticas que más afectaron la calidad de las entregas fueron: guías de código comunes, refactorización, test de regresión y la integración continua.

Este estudio ha logrado que integrantes del proyecto incluyan esta temática en la planificación de los espacios curriculares a los que pertenecen y que por supuesto están relacionados con ingeniería de software y programación. Asimismo, los grupos de alumnos involucrados en la realización del trabajo final de carrera están utilizando algunas de sus prácticas.

El grupo espera durante este nuevo ciclo:

- Realizar eventos en el área de extensión de la Universidad relacionados con prácticas ágiles.
- Evaluar la aplicación de las prácticas ágiles durante el desarrollo del proyecto.
- Estudiar la integración entre las otras líneas de investigación y metodologías ágiles.
- Analizar las metodologías ágiles más adecuadas para trabajar en las organizaciones de la patagonia austral.

Formación de Recursos Humanos

Entre los integrantes del proyecto existen dos alumnos de la Maestría de Informática y Sistemas – uno avanzado y otro iniciando -, dos

alumnos de la Maestría en Educación en Entornos Virtuales – uno de ellos en trabajo de tesis -, un alumno en proceso de trabajo final en la Especialización en Management Tecnológico y dos alumnos de pre-grado de Analista de Sistemas en proceso de realización de trabajo final de carrera, en estos casos uno de ellos está trabajando sobre esta temática en particular.

El grupo se complementa con un académico perteneciente a la Universidad de Magallanes doctorado en el área de computación y automatización industrial por la Universidad Estatal de Campiñas (Brasil) quien forma docentes y alumnos en la Unidad Académica Río Turbio (UART).

Referencias

- [1] Beck, K., et al.: The Agile Manifesto. Manifesto for Agile Software Development. En: <http://www.agilemanifesto.org>
- [2] Agile Alliance, En: <http://www.agilealliance.org/>
- [3] Boehm, B. y Turner, R. “Balancing Agility and Discipline. A Guide for the Perplexed”. Addison-Wesley. 2004.
- [4] Ambler, S. Survey Says: Agile Works In Practice. En: <http://www.drdoobs.com/architecture-and-design/survey-says-agile-works-in-practice/191800169>. 2006.
- [5] Shine Technologies. Agile Technologies Survey Results. 2003.
- [6] VersionOne. State of agile survey: The State of Agile Development. 2011.
- [7] Alende, A. La utilización de los métodos ágiles en las empresas de desarrollo de software de Argentina. 2010.
- [8] Ali, M. Survey on the State of Agile Practices implementation in Pakistan. International Journal of Information and Communication Technology Research. Vol. 2. Nro. 5. 2012.
- [9] Rivadeneira, S. Metodologías ágiles enfocadas al modelado de requerimientos. ICT UNPA. 2013.
- [10] LNCS. Ingeniería del Software: Metodologías y Ciclos de Vida. Laboratorio Nacional de Calidad de Software. INTECO. España. 2009.
- [11] Meda, R. y Ierache, J. Una propuesta de conjunción de elementos metodológicos en común dentro de los enfoques ágiles para el desarrollo de software.
- [12] Reynoso, C. Métodos Heterodoxos en Desarrollo de Software. UBA. Argentina. 2004.
- [13] Larman, C. Agile and Iterative Development: A manager’s guide. Addison-Wesley. 2004.

⁵ Behavior-Driven Development

- [14]Cao, L. y Ramesh, B. Agile Requirements Engineering Practices: An Empirical Study. IEEE Software. 2008.
- [15]Mugridge, R. Managing Agile Project Requirements with Storytest-Driven Development. IEEE Software. 2008.
- [16]Mendes, K., Estevez, E., Fillotrani, P. Un framework para evaluación de metodologías ágiles.
- [17]Beck, K. Extreme Programming Explained. Embrace Change, Pearson Education, 1999.
- [18]Scrum Alliance. En: <http://www.scrumalliance.org/>
- [19]Highsmith, J. Agile Project Management: creating innovative products. Pearson Education. USA. 2004.
- [20]Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J., Agile Software Development Methods, Review and Analysis, VTT Publications 478. 2002.
- [21]Iacovelli, A., Souveyet, C., Framework for Agile Methods Classification, Workshop on Model Driven Informat. Systems Eng.: Enterprise, User and System Models. 2008.
- [22]Strode, D.E., The Agile Methods: An Analytical Comparison of Five Agile Methods and an Investigation of Their Target Environment, MSc Thesis in Information Systems, Massey University, Palmerstin North, Nueva Zelanda. 2005.
- [23]Visconti, M., Cook, C., An Ideal Process Model for Agile Methods, LNCS, ISBN 978-3-540-21421-2, Vol 3009, pp.439-441. 2004.
- [24]Oyeyipo E., Mueller C. Requirements Management in Agile-Scrum. Texas State University. San Marcos. 2011.
- [25]Gottesdiener E. "Requirements Practices on Agile Projects", Success with Requirements, Vol. 1, No. 8. 2007.
- [26]Qasaimeh, M., Mehrfard, H., Hamou-Lhadj, A. Comparing Agile Software Process Based on the Software Development Project Requirements. IEEE Computer Society. CIMCA 2008
- [27]Araújo, J. y Ribeiro, J. Towards an Aspected-Oriented Agile Requirements Approach. WPSE'05.