

*Revista*  
*Española de*  
**Innovación,**  
**Calidad e**  
**Ingeniería del Software**



**Volumen 6, Número 3 (especial XI JICS), noviembre,  
2010**

**Web de la editorial:** [www.ati.es](http://www.ati.es)

**Web de la revista:** [www.ati.es/reicis](http://www.ati.es/reicis)

**E-mail:** [calidadsoft@ati.es](mailto:calidadsoft@ati.es)

**ISSN:** 1885-4486

Copyright © ATI, 2010

Ninguna parte de esta publicación puede ser reproducida, almacenada, o transmitida por ningún medio (incluyendo medios electrónicos, mecánicos, fotocopias, grabaciones o cualquier otra) para su uso o difusión públicos sin permiso previo escrito de la editorial. Uso privado autorizado sin restricciones.

Publicado por la Asociación de Técnicos de Informática (ATI), Via Laietana, 46, 08003 Barcelona.

Secretaría de dirección: ATI Madrid, C/Padilla 66, 3º dcha., 28006 Madrid



## **Editor**

**Dr. D. Luís Fernández Sanz (director)**

Departamento de Ciencias de la Computación, Universidad de Alcalá

## **Miembros del Consejo Científico**

**Dr. Dña. Idoia Alarcón**

Depto. de Informática  
Universidad Autónoma de Madrid

**Dr. D. José Antonio Calvo-Manzano**

Depto. de Leng y Sist. Inf. e Ing. Software  
Universidad Politécnica de Madrid

**Dra. Tanja Vos**

Depto. de Sist. Informáticos y Computación  
Universidad Politécnica de Valencia

**Dña. M<sup>a</sup> del Pilar Romay**

CEU Madrid

**Dr. D. Alvaro Rocha**

Universidade Fernando Pessoa  
Porto

**Dr. D. Oscar Pastor**

Depto. de Sist. Informáticos y Computación  
Universidad Politécnica de Valencia

**Dra. Dña. María Moreno**

Depto. de Informática  
Universidad de Salamanca

**Dra. D. Javier Aroba**

Depto de Ing. El. de Sist. Inf. y Automática  
Universidad de Huelva

**D. Guillermo Montoya**

DEISER S.L.  
Madrid

**Dr. D. Pablo Javier Tuya**

Depto. de Informática  
Universidad de Oviedo

**Dra. Dña. Antonia Mas**

Depto. de Informática  
Universitat de les Illes Balears

**D. Jacques Lecomte**

Meta 4, S.A.  
Francia

**Dra. Raquel Lacuesta**

Depto. de Informática e Ing. de Sistemas  
Universidad de Zaragoza

**Dra. María José Escalona**

Depto. de Lenguajes y Sist. Informáticos  
Universidad de Sevilla

**Dr. D. Ricardo Vargas**

Universidad del Valle de México  
México

---

## Contenidos

---

REICIS

<b>Editorial</b>	<b>4</b>
<i>Luís Fernández-Sanz</i>	
<b>Presentación</b>	<b>5</b>
<i>Luis Fernández-Sanz</i>	
<b>Taxonomía de factores críticos para el despliegue de procesos software</b>	<b>6</b>
<i>Sussy Bayona, Jose Calvo-Manzano, Gonzalo Cuevas, Tomás San Feliu</i>	
<b>Sistema de Gestión Integrado según las normas ISO 9001, ISO/IEC 20000 e ISO/IEC 27001</b>	<b>25</b>
<i>Antoni Lluís Mesquida, Antònia Mas, Esperança Amengual, Ignacio Cabestrero</i>	
<b>Implantación de CMMi nivel de madurez 2 en una PYME</b>	<b>35</b>
<i>Fernando Ramos, Olimpia Torres, Nicolás Sánchez, Manuel Alba</i>	
<b>Pruebas de Aceptación en Sistemas Navegables</b>	<b>47</b>
<i>José Ponce, Francisco José Domínguez-Mayo, M. José Escalona, Manuel Mejías, Diego Pérez, Gustavo Aragón, Isabel Ramos</i>	
<b>Análisis de métricas básicas y herramientas de código libre para medir la mantenibilidad</b>	<b>56</b>
<i>Emanuel Irrazábal, Javier Garzás</i>	
<b>Reduciendo distancia en proyectos de Desarrollo de Software Global Ágiles con técnicas de Ingeniería de Requisitos</b>	<b>66</b>
<i>Mariano Minoli, Valeria de Castro, Javier Garzás</i>	
<b>CMMI después de la certificación</b>	<b>76</b>
<i>Vanesa Cabral y Juanjo Cukier</i>	
<b>Comparando UML y OWL en la representación del conocimiento: correspondencia sintáctica</b>	<b>84</b>
<i>Susana M. Ramírez, Yisel Alonso, Violena Hernández, Arturo Cesar Arias y Dayana La Rosa</i>	

# **Reduciendo distancia en proyectos de Desarrollo de Software Global Ágiles con técnicas de ingeniería de requisitos**

Mariano Minoli  
Assertum Tecnologías  
mariano.minoli@assertum.es

Valeria de Castro, Javier Garzás  
Universidad Rey Juan Carlos  
{valeria.decastro, javier.garzas}@urjc.es

## **Resumen**

La tendencia a incluir agilidad en los métodos de desarrollo de software ha sido un tema recurrente durante los últimos años en ámbitos académicos y empresariales. Los equipos de desarrollo que trabajan de forma distribuida (conocidos como equipos *GSD*, *Global Software Development*) no han sido una excepción. Múltiples propuestas abogan por la adopción de técnicas ágiles para el desarrollo y la administración de estos equipos de desarrollo. Aunque a primera vista esta evolución hacia lo ágil parece natural, en realidad surgen múltiples interrogantes relacionados a combinar *agilidad* con el factor más complicado de tratar en equipos distribuidos: la distancia. Desde nuestro punto de vista, técnicas existentes de ingeniería de software pueden ser útiles para abordar esta problemática. En este artículo presentamos el uso de técnicas de ingeniería de requisitos como herramienta para estrechar la brecha que se produce al mezclar agilidad y distancia.

**Palabras clave:** Desarrollo Global de Software, Métodos Ágiles, Ingeniería de Requisitos.

## **Using Requirement Engineering Techniques to reduce Distance on Agile Global Software Development Projects**

### **Abstract**

The trend of including agility on software development methods has been recurrent in recent years both in academia and business. Global Software Development (GSD) teams have not been an exception, many proposals advocate for the adoption of agile techniques for this kind of teams. Although at first glance the trend towards agile seems natural, there actually are many questions related to combining agility with distance. From our point of view, existing software engineering techniques can be useful in addressing this problem. In this paper we will present the use of requirements engineering techniques as a tool to bridge the gap that occurs when mixing these two things: agility and distance.

**Key words:** Global Software Development, Agile Methods, Requirement Engineering.

*Minoli, M., Castro, V. y Garzás, J., "Reduciendo distancia en proyectos de Desarrollo de Software Global Ágiles con técnicas de ingeniería de requisitos", REICIS, vol. 6, no.3, 2010, pp.66-75. Recibido: 8-11-2010; revisado: 14-11-2010; aceptado: 19-11-2010*

## **1. Introducción**

Dos de las tendencias que han marcado el camino en el mundo del desarrollo de software en los últimos años han sido el Desarrollo Global de Software (*GSD, Global Software Development*) y las metodologías Ágiles. GSD es la evolución natural del negocio de desarrollo de software a la descentralización de tareas (como la programación o las pruebas) hacia sitios remotos, normalmente más rentables [1]. Estos sitios remotos pueden ser algo tan cercano como otra empresa dentro de la misma ciudad o país, o algo tan lejano como empresas en otro continente, incluso con diferente zona horaria, cultura, etc. Esta tendencia ha sido motivada por cuestiones de negocio, más que por cuestiones relacionadas a la ingeniería de software. Es por ello que muchas de las metodologías de desarrollo de software están siendo adaptadas a estos entornos [2] [3]. Los métodos de desarrollo ágil [4] han ganado protagonismo en los últimos años como la reacción de la comunidad a las metodologías formales, sobre-documentadas, que tradicionalmente han dominado el mundo de la ingeniería de software. La necesidad de combinar estas tendencias parece ineludible, sin embargo genera cuestiones que deben ser respondidas [5] [6]. La investigación existente en este área se ha focalizado en analizar el impacto de la distancia en proyectos de GSD (ágiles y no ágiles) [7] [8] [9], sin embargo existen pocos trabajos que propongan prácticas para resolver los problemas (normalmente de comunicación) generados por la distancia.

Nuestra investigación nos ha llevado a plantear la hipótesis de que se podrían utilizar herramientas existentes de ingeniería de software para apoyar la implantación metodologías ágiles en entornos de desarrollo global de software (A-GSD). En particular hemos escogido la utilización de técnicas de ingeniería de requisitos (IR). Históricamente la mala gestión de requisitos ha sido motivo de frecuentes fracasos de proyectos de desarrollo [11]. En este trabajo se propone la adaptación y el uso de técnicas conocidas de IR con el objetivo de disminuir la problemática relacionada a la comunicación en equipos de A-GSD. El objetivo de utilizar estas técnicas (con cierto grado de adaptación) es el de cubrir la brecha existente entre los miembros de equipos que se encuentran distribuidos geográfica, temporal y culturalmente sin afectar el carácter ágil de los procesos empleados por equipo de trabajo.

El presente artículo se estructura de la siguiente manera. En la sección 2 se presentan los trabajos relacionados con esta propuesta. En la sección 3 se presenta la aportación de este trabajo. La sección 4 resume las conclusiones y futuros trabajos.

## **2. Trabajos relacionados**

En esta sección se presentarán las principales áreas que fundamentan nuestra propuesta. En la sección 2.1 expondremos la problemática del desarrollo en entornos A-GSD. En la sección 2.2 presentaremos trabajos realizados en torno a gestión de requisitos en entornos GSD (no ágiles) que serán utilizados como base para nuestra propuesta.

### **2.1. Desarrollo Global de Software Ágil (A-GSD)**

El mundo de los negocios TIC ha evidenciado en los últimos años una irreversible tendencia hacia la descentralización de las operaciones. La aparición de Internet a mediados de los años noventa y distintas circunstancias económicas han favorecido la práctica de brindar servicios desde centros de producción remotos. El desarrollo de software ha sido una de las disciplinas donde se ha hecho más frecuente esta relocalización de la fuerza de trabajo. Hoy en día es ampliamente aceptado el hecho de que muchas de las principales compañías producen total o parcialmente su software en ubicaciones diferentes a donde se encuentra el cliente final. Esta tendencia [1] se conoce con el nombre de Desarrollo Global de Software (GSD).

En paralelo, en los últimos años han ganado popularidad un conjunto de nuevas técnicas y métodos de desarrollo con el nombre de metodologías ágiles [12]. Existen estudios [13] [14] que demuestran que en ciertos entornos (normalmente equipos pequeños con un alto grado de implicación del usuario) estas técnicas pueden resultar más productivas que los métodos de desarrollo tradicionales. Parece lógico que los equipos de GSD estén siendo conducidos a seguir esta tendencia para poder aprovechar todo su potencial [2]. Sin embargo, la combinación de GSD y métodos ágiles, no es trivial. Aunque la primera reacción a mezclar Métodos Ágiles y GSD ha sido de escepticismo [15], trabajos recientes [5] [6] han demostrado su factibilidad, siempre que ciertos aspectos sean tenidos en cuenta. El denominador común detrás de los aspectos que deben ser considerados es la *comunicación* entre los integrantes del equipo [8] [16] [17]. Si tenemos en cuenta que la mayor parte de las prácticas propuestas por las metodologías ágiles se basan en la interacción entre los participantes, la brecha resulta evidente. Existen varios estudios que presentan diferentes enfoques para disminuir esta brecha [7] [9] [18], sin embargo ninguno de ellos lo hace desde el punto de vista de la ingeniería del software. Además ninguno de ellos tiene en cuenta el aspecto ágil en los equipos de proyecto GSD.

## **2.2. Gestión de Requisitos en entornos GSD**

Desde la aparición de GSD como tendencia mundial, varios autores han escrito sobre el impacto de la distribución geográfica de equipos de trabajo en la ingeniería de requisitos [19] [20] [21]. Estos estudios han concluido en la necesidad de adaptar las técnicas de IR a los entornos GSD [19]. Según [22], los mayores desafíos en la gestión de requisitos en entornos distribuidos de trabajo son: Lograr una correcta comunicación, facilitar herramientas de gestión del conocimiento y mitigar diferencias culturales y de zonas horarias. El problema de estos enfoques es que ninguno de ellos ha tenido en cuenta el aspecto ágil, han sido pensados para GSD, no para A-GSD. Es por ello que nuestra propuesta selecciona y adapta un conjunto de técnicas de IR tradicionales [23] y distribuidas [19] adaptadas con el objetivo concreto de disminuir la brecha de comunicación causada por la combinación de distribución y agilidad en equipos A-GSD. En la próxima sección se proponen adaptaciones de estas técnicas.

## **3. Requisitos en A-GSD**

Una buena comunicación es la piedra angular de los proyectos A-GSD. Todas las disciplinas asociadas a la IR implican comunicación. La IR tradicional propone una serie de técnicas para estas disciplinas. Pueden utilizarse estas técnicas para mantener principios del manifiesto ágil como promover a los individuos y las interacciones por sobre los procesos y las herramientas, el software en funcionamiento sobre la documentación, la colaboración del cliente sobre la negociación de contratos y la respuesta al cambio sobre los planificaciones rigurosas [24]. En la siguiente sección se propone la utilización de técnicas de IR como complemento a las técnicas ágiles con el objetivo de ser usadas en A-GSD.

### **3.1. Uso de técnicas de IR en A-GSD**

La mayor parte de los autores [23] [25] [26] [27] [28] están de acuerdo en que las principales disciplinas relacionadas a la IR son: elicitación, análisis y negociación, documentación, validación y gestión de los requisitos. Para los propósitos de este trabajo, se usarán estas disciplinas para agrupar las herramientas propuestas. Según [28], algunos de los principales desafíos que aparecen al incluir la componente ágil a proyectos distribuidos son: falta de visibilidad del estado del proyecto, pérdida del contexto técnico y de negocio,

falta de una infraestructura común, disminución del caudal de comunicación y disminución de la confianza.

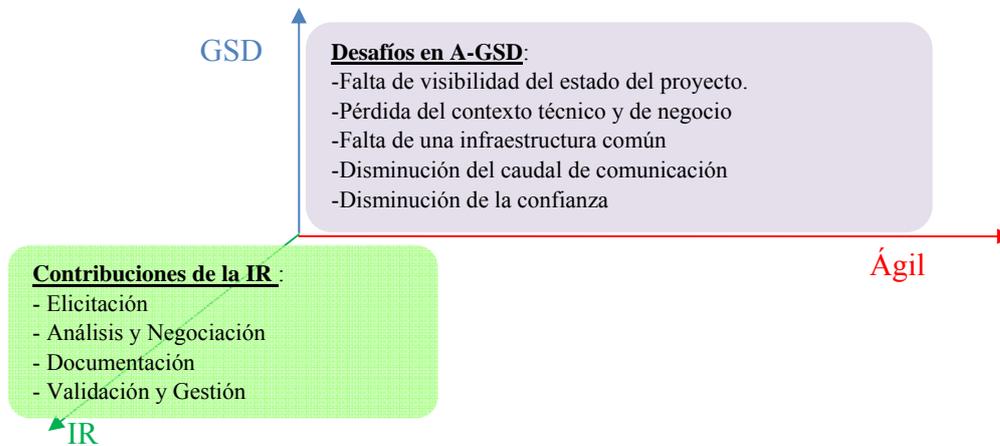


Figura 1. Contribuciones de la IR en entornos A-GSD

Problemática en A-GSD	Disciplinas IR	Técnicas
Falta de visibilidad del estado del proyecto	Análisis y Negociación	Priorización de requisitos
	Validación y Gestión	Trazabilidad de requisitos Repositorio único de requisitos
Pérdida del contexto técnico y de negocio	Elicitación	Técnicas tradicionales de elicitación de requisitos (cuestionarios y encuestas, entrevistas, análisis de documentación existente) Técnicas de elicitación grupales ( <i>brainstorming</i> y grupos de enfoque)
	Análisis y Negociación	Joint Application Development (JAD)
Falta de una infraestructura común	Documentación	Prototipos Técnicas de ingeniería conducida por modelos
	Validación y Gestión	Repositorio único de requisitos
Disminución del caudal de comunicación	Análisis y Negociación	Joint Application Development (JAD) Registro de la interacciones con el usuario
	Documentación	Casos de Uso Sitios Wiki
Disminución de la confianza	Elicitación	Técnicas tradicionales de elicitación de requisitos (cuestionarios y encuestas, entrevistas, análisis de documentación existente). Técnicas de elicitación grupales ( <i>brainstorming</i> y grupos de enfoque).
	Análisis y Negociación	Registro de la interacciones con el usuario

Tabla 1. Relación de técnicas de IR propuestas

Nuestra hipótesis plantea que todos estos problemas pueden ser resueltos aplicando técnicas de IR. La Figura 1 muestra estos desafíos (surgidos a incluir el aspecto Ágil en GSD) y como podrían aportar la inclusión de técnicas de IR. En la Tabla 1 se resumen las

técnicas propuestas, agrupadas por cada disciplina. En las próximas secciones se describirán cada una de estas técnicas.

No es el propósito de este trabajo describir estas conocidas técnicas, en las próximas secciones nos limitaremos a explicar aspectos relacionados a cómo utilizarlas en entornos A-GSD.

### **3.2.1 Uso de técnicas de elicitación de requisitos en A-GSD**

La información recolectada durante la elicitación de requisitos siempre debe ser interpretada, analizada, modelada y validada antes de que los ingenieros de requisitos puedan documentarla [23]. En entornos A-GSD, las técnicas tradicionales de elicitación de requisitos pueden aplicarse sin cambios significativos. Técnicas de elicitación de requisitos tradicionales como cuestionarios y encuestas, entrevistas y análisis de documentación existente pueden ser usadas sin necesidad de adaptarlas a entornos A-GSD. Aquí es importante tener en cuenta la utilización de medios digitales que faciliten el intercambio de la información a través de Internet. Por otro lado, gracias a herramientas digitales de comunicación grupal, es posible también el empleo de elicitación de requisitos grupales como *brainstorming* y grupos de enfoque. En [30] puede encontrarse una recopilación de productos de comunicación grupal.

### **3.2.2. Uso de técnicas de análisis y negociación de requisitos en A-GSD**

Los procesos de análisis y negociación están íntimamente relacionados a la elicitación de requisitos. La técnica JAD (*Joint Application Development*) es una sesión de grupo (o taller), con un enfoque de análisis estructurado. Durante las sesiones JAD los desarrolladores y los clientes pueden discutir las características del producto [31]. Este tipo de discusión puede llevarse a cabo en una forma distribuida utilizando las tecnologías actuales de comunicación.

### **3.2.3. Uso de técnicas de documentación de requisitos en A-GSD**

Las técnicas de requisitos propuestas para equipos ágiles [32] [33] tienden a minimizar la documentación generada durante el proyecto. Sin embargo, para el caso de A-GSD, consideramos que este comportamiento debe ser modificado. Según estudios analizados [34] existe una relación de complementación entre las técnicas de documentación de IR tradicional (como los casos de uso) y las técnicas de requisitos ágiles (como los Story

Cards). Nuestra hipótesis postula que, en A-GSD, la utilización de Casos de Uso mejora la comunicación debido a que permite la formalización de requisitos. Esta formalización permite facilitar la comunicación cuando los integrantes del equipo no se encuentran en la misma ubicación física. Otras técnicas de documentación como los prototipos o las técnicas de ingeniería dirigida por modelos pueden ser usadas para complementar los casos de uso. Una decisión importante que deberá tomar el equipo es el límite con respecto a cuánto documentar. En [35], Scott Ambler defiende: “El objetivo es implementar requisitos, no documentarlos”. Nuestra consideración al respecto en entornos distribuidos es que, aunque este espíritu debe mantenerse, no menos importante es considerar que cierto grado de formalización contribuirá a disminuir la brecha de comunicación en equipos A-GSD.

#### **3.2.4. Uso de técnicas de validación y gestión de requisitos en A-GSD**

Las técnicas de documentación descritas en la sección anterior deben servir como base para la validación por parte del usuario y su posterior gestión. Es muy importante disponer de un repositorio único que aglutine todos los documentos generados durante la etapa de documentación. Algunos autores recomiendan la utilización de Wikis [36] como herramienta que centralice todos los esfuerzos de documentación y a la vez admitan información no estructurada.

### **4. Conclusiones y trabajo futuro**

En este trabajo se ha propuesto la introducción de técnicas de IR en A-GSD para reducir la *distancia* y obtener una mejor comunicación entre los integrantes de equipos distribuidos. Se ha seleccionado un conjunto de técnicas conocidas que pueden aportar valor a procesos de desarrollo distribuido ágil, y se ha incluido una descripción de cómo estas técnicas pueden ser utilizadas.

Aunque a primera vista la introducción de estas técnicas tradicionales podría ser vista como un contraste a los métodos ágiles, nuestra opinión es que complementan muy bien a los métodos ágiles en equipos distribuidos. Así, una mayor documentación es necesaria en equipos A-GSD, en contraste a que ocurre en equipos ágiles que se encuentran ubicados en el mismo sitio.

Son necesarios más estudios para determinar cómo deberían ser aplicadas estas técnicas en metodologías concretas como eXtreme Programming, SCRUM o Crystal

Family entre otras. Por otro lado, todas las herramientas propuestas necesitan ser probadas en entornos de desarrollo reales. Ambas, propuestas y experiencias serán presentadas en trabajos futuros.

## **Agradecimientos**

Este trabajo está parcialmente financiado por el proyecto MODEL CAOS financiado por el Ministerio de Ciencia y Tecnología de España (Ref. TIN2008-03582) y Agreement Technologies (CONSOLIDER CSD2007-0022).

## **Referencias**

- [1] Herbsleb, JD, Moitra, D., “Global Software Development” *IEEE Software*, 2001.
- [2] Holmström, H, Fitzgerald, B, Ågerfalk, PJ, “Agile practices reduce distance in global software development”, *Information Systems Summer 2006 ABI/INFORM Global*, pp. 7.
- [3] Vax, M., Michaud, S., “Distributed Agile: Growing a Practice Together” *Agile Conference*, 2008
- [4] Cockburn, A. *Agile Software Development* Addison Wesley, 2001.
- [5] Paasivaara, M., Lassenius, C., “Could Global Software Development Benefit from Agile Methods?” *International Conference on Global Software Engineering, 2006.*, pp. 109 - 113.
- [6] Ramesh, B, Cao, L, Mohan, K y Xu., P, “Can distributed software development be agile?” *Communications of the ACM*, vol. 49, nº 10, 2006.
- [7] Prikladnicki, Rafael, Audy, Jorge Luis Nicolas, Evaristo, Roberto, “Global Software Development in Practice Lessons Learned”, *Software Process Improvement 2003*, pp. 267–281.
- [8] Herbsleb, JD “Global software engineering: The future of socio-technical coordination”. *Int. Conference on Software Eng, Future of Software Engineering*, pp. 188-198, 2007
- [9] Carmel, E. Agarwal, R. “Tactical approaches for alleviating distance in global software development.”, *IEEE Software*, vol 18, nº 2, pp. 22 – 29, 2001.
- [10] Prior, P., “Requirements Management in a Distributed Agile Environment.”, 2º *WEC*, 2005.
- [11] McConnell, S., “Avoiding S. E. classic mistakes”, *Software IEEE*, vol. 13, nº 5, 1996.

- [12] Dyba, T., Dingsoyr, T., “What Do We Know about Agile S. D.?” , *Soft. IEEE*, pp. 6-9, 2009
- [13] Cohn, M., *User Stories Applied: For Agile Software Development*. Addison-Wesley, 2004.
- [14] Sutherland, J., Schwaber, K, *The Scrum Papers*. Scrum, 2007.
- [15] Daniel B. Markham, Roanoke, Virginia, “Agile Won't Work Implementing Agility in Non-standard Teams”, *IEEE DOI 10.1109/AGILE. 32*, 2009.
- [16] Paasivaara, M., Lassenius, C., “Collaboration practices in global inter-organizational software development projects”, *GSD: Growing Opportunities, Ong. Challenges*, pp. 183–199.
- [17] Herbsleb, JD, Mockus, A, Finholt, TA, Grinter, RE, “An empirical study of global software development: distance and speed”, *23rd Int. Conf. on Software Eng. Toronto*, pp. 81-90, 2001.
- [18] Ye , Y., Nakakoji, K., Yamamoto, Y., “Measuring and Monitoring Task Couplings of Developers and Development Sites in Global Software Development”, *Software Engineering Approaches for Offshore and Outsourced Development*, pp. 181-195, 2009.
- [19] Damian, D, “Stakeholders in Global Requirements Engineering: Lessons Learned from Practice”, *Software IEEE*, vol 24 , nº 2, pp. 21-27, 2007.
- [20] Lopez, A., Nicolas, J. y Toval, A, “Risks and Safeguards for the Requirements Engineering Process in Global Software Development”, *ICGSE*, pp. 394 – 399, 2009.
- [21] Sabahat, Nosheen “An iterative approach for global requirements elicitation: A case study analysis”, *Electronics and Information Engineering (ICEIE)*, pp. V1-361 - V1-366, 2010.
- [22] Damian, D., Zowghi, Didar, “Requirements Engineering challenges in multi-site software development”, *Requirements Engineering Journal*, pp. 149-160, 2003.
- [23] Nuseibeh, B., Easterbrook, S., “Requirements engineering: a roadmap”, *Proceedings of the Conference on The Future of Software Engineering table of contents*, pp. 35 – 46, 2000.
- [24] Beck, K. *Manifesto for Agile Software Development*. <http://agilemanifesto.org/>. 05/11/10.

- [25] Cheng, B., “Research directions in requirements”, *Future of Soft. Eng.* pp. 285-303, 2007.
- [26] Jiang, L., Eberlein, A., & Far, B., “Combining requirements engineering techniques - theory and case study”, *12th Conf. on the Computer-Based Systems, ECBS '05*, pp. 105-112, 2005.
- [27] Kotonya, G., Sommerville, I., *Requirements Engineering*. John Wiley & Sons. 1997
- [28] Damian, D., Zowghi, D., “Requirements Engineering challenges in multi-site software development”, *Requirements Engineering Journal*, pp. 149-160, 2003.
- [29] Vax, M., Michaud, S., “Distributed Agile: Growing a Practice Together”, *Agile Conf.* 2008.
- [30] Lanubile, F., Ebert, C., Prikladnicki, R., Vizcaino, A., "Collaboration Tools for Global Software Engineering" *IEEE SOFTWARE*, pp. 52-55, 2010.
- [31] Paetsch, F., Eberlein, A., “Requirements Engineering and Agile Software Development”, *12º Workshop on Enabling Technologies: Infrastructure for Collaborative Ent.*, pp. 308, 2003.
- [32] Racheva, Z., Daneva, M., "How Do Real Options Concepts Fit in Agile Requirements Engineering?" *Eighth ACIS International Conference on Software Engineering Research, Management and Applications*. pp. 231-238, 2010.
- [33] Cao, Lan, Ramesh, B., “Agile Requirements Engineering Practices: An Empirical Study” *Software IEEE*, vol 25, nº 1, pp. 60 – 67, 2008.
- [34] Gallardo-Valencia, R, Olivera, V.; Sim, S.E., “Are Use Cases Beneficial for Developers Using Agile Requirements?” *Workshop on Comp. Evaluation in Req Eng.* pp. 11-22, 2007.
- [35] Ambler, S., “Agile Requirements Best Practices”  
<http://www.agilemodeling.com/essays/agileRequirementsBestPractices.htm> 05/11/10
- [36] Fowler, M., “Agile Requirements Best Practices”, [www.martinfowler.com/articles/agileOffshore.html](http://www.martinfowler.com/articles/agileOffshore.html) 05/11/10