

Desarrollo de Software Global Ágil: una Caracterización del Conocimiento

Mariano Minoli¹, María Valeria de Castro², Javier Garzas Parra^{2,3}

¹ Assertum Tecnologías S.L.,
C/Albasanz 57, 28037 Madrid, España
mariano.minoli@assertum.es

² Grupo de Investigación Kybele, Universidad Rey Juan Carlos I
C/Tulipán s/n, 28933 Móstoles, Madrid, España
{javier.garzas, valeria.decastro}@urjc.es

³ Kybele Consulting S.L.
Oliva 18, Las Rozas, Madrid, España
javier.garzas@kybeleconsulting.com

Abstract. La inclusión de metodologías ágiles en proyectos de Desarrollo Global de Software (GSD por las siglas en inglés de Global Software Development) se ha convertido en un tema de discusión recurrente en los últimos años tanto en ambientes académicos como empresariales. A pesar de que un enfoque ágil para equipos GSD (A-GSD) podría verse como evolución de los equipos distribuidos hacia las nuevas prácticas de ingeniería de software, existen varios aspectos que deben ser tratados para combinar exitosamente distribución y agilidad. Esta problemática, unida a la cantidad y diversidad de propuestas que han emergido en los últimos años, hace que sea difícil para los profesionales tomar decisiones en entornos A-GSD. Por estos motivos hemos decidido llevar a cabo una revisión de la literatura existente y una caracterización del conocimiento en A-GSD a través de una ontología. La ontología que presentamos permitirá a los profesionales identificar los principales desafíos en este tipo de equipos e identificar las mejores prácticas y herramientas para resolverlos.

Palabras Clave: Desarrollo Global de Software, Métodos Ágiles, Equipos Distribuidos.

1 Introducción

Dos de las disciplinas de ingeniería del software que han evolucionado de manera considerable en los últimos años son el Desarrollo Global de Software (GSD) [8] y los métodos Ágiles [9]. GSD es la evolución del negocio de desarrollo de software hacia la descentralización de tareas a sitios remotos normalmente más rentables. Esta tendencia está conducida por motivos económicos y ha obligado a la adaptación de los métodos y técnicas de desarrollo. Los métodos de desarrollo ágil han ganado protagonismo en los últimos años como la reacción de la comunidad a las metodologías formales que tradicionalmente han dominado el mundo de la ingeniería de software. Aunque estas disciplinas han nacido y evolucionado de manera separada, en años recientes se ha comenzado a plantear su uso en conjunto. Las primeras evidencias de ello muestran discusiones contradictorias sobre si aplicar métodos ágiles en GSD

contribuye o resta valor a los equipos de desarrollo [4][5][7]. Por ejemplo, en [4] el autor David Parnas describe a la utilización de métodos ágiles en GSD como “*la solución equivocada para un problema real y antiguo*”. En su exposición explica que los problemas (o desafíos) relacionados a GSD son los mismos a los cuales se enfrentan profesionales de desarrollo que trabajan en un mismo sitio, con un incremento en los problemas asociados a la comunicación. En este sentido explica que los métodos ágiles lejos de mejorar este aspecto, lo complican aún más. Sin embargo, en [7] los autores concluyen que “*la incorporación cuidadosa de agilidad en entornos distribuidos de desarrollo de software es esencial para hacer frente a varios de los desafíos relacionados a equipos distribuidos, como la comunicación, el control y la confianza*”. Por último en [5] los autores hacen especial referencia a la “*necesidad de disponer de herramientas tecnológicas que permitan a los equipos de desarrollo enfrentarse a los desafíos de A-GSD*”, sin especificar cuáles son estas herramientas. El análisis de las revisiones sistemáticas encontradas nos indica un aumento de la actividad de investigación en los últimos años [2][3]. Estas revisiones realizan un correcto análisis de los desafíos a los que se enfrentan los equipos de A-GSD, sin embargo carecen de una completa revisión sobre las prácticas y herramientas que están siendo propuestas y usadas. Dos ejemplos claros de estas herramientas son [10] y [11] en donde los autores proponen la utilización de técnicas avanzadas de Wikis y redes sociales para mejorar la comunicación en equipos A-GSD. Esta situación es aún más evidente en la comunidad *no académica* de usuarios de métodos ágiles. El estado de la práctica de esta comunidad no se ve reflejado en ninguna de las revisiones encontradas, sin embargo representa una fuente valiosa de herramientas que están siendo utilizadas. Esta dispersión de la información, distribuida en distintas fuentes de información hace muy difícil a los profesionales la utilización de todos los avances que se han producido en el ámbito de A-GSD en los últimos años.

Con el objetivo de valorar el estado del arte y la práctica de manera conjunta y organizar toda esta información de manera que pueda ser utilizada por los profesionales de A-GSD, hemos decidido realizar una revisión de la literatura existente y organizar el resultado de la misma en una caracterización del conocimiento. Para la revisión hemos tenido en cuenta fuentes de datos formales (como congresos de referencia o bases de datos científicas) y también las fuentes de datos informales (cómo repositorios de código abierto). Para organizar la información de manera sistemática hemos decidido utilizar una ontología semi-formal basada en UML, lo cual facilitará su utilización por parte de la comunidad de profesionales de A-GSD.

El resto del artículo está estructurado de la siguiente manera, en la sección 2 se presentan los resultados de la revisión de la literatura realizada. En la sección 3 se presenta la caracterización de conocimiento.

2 Resultados de la Revisión de la literatura sobre A-GSD

Debido a la extensión del trabajo, en este artículo nos limitaremos a presentar los principales resultados de la revisión realizada.

De los 89 trabajos identificados, el 47,20% trata sobre desafíos, el 47,20% trata sobre prácticas, un 19,10% propone metodologías y un 35,95% propone herramientas.

Haciendo un análisis cronológico de la información encontrada puede verse que hay poca actividad relacionada a A-GSD antes de 2006. En segunda instancia que entre 2006 y 2008 ha habido más actividad relacionada con los desafíos y prácticas, y por último que entre 2008 y 2010 la cantidad de propuestas relacionadas a herramientas ha crecido por sobre las demás. En la figura 1 se muestra esta relación.

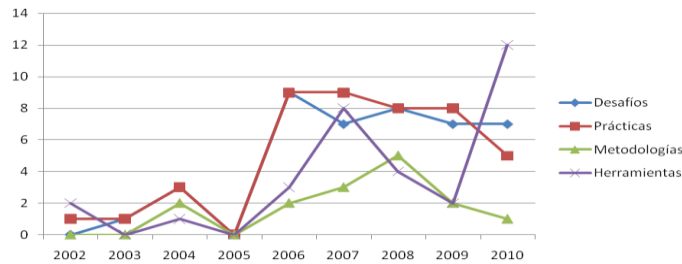


Fig. 1. Distribución cronológica de los trabajos encontrados en la revisión.

A continuación presentaremos las respuestas encontradas a las preguntas planteadas en la revisión de literatura.

- *RQ1: ¿Qué tipos de desafíos enfrentan los equipos en entornos A-GSD?*

El análisis de los trabajos encontrados que tratan sobre desafíos ratifica la idea de que el principal desafío presente en proyectos A-GSD es la comunicación. De 82 trabajos que se han analizado, 42 de ellos tratan sobre desafíos en A-GSD. Todos los trabajos que tratan sobre desafíos consideran a la comunicación como el principal reto en este tipo de entornos. Además 17 trabajos tratan exclusivamente sobre el aspecto de la comunicación en A-GSD. Aunque esta conclusión no aporta novedad alguna a lo expresado en trabajos relacionados, la categorización encontrada nos permitirá construir una ontología más rica y en consecuencia poder seleccionar mejor las prácticas que solucionan cada tipo de desafíos.

- *RQ2: ¿Cuáles prácticas y metodologías han sido propuestas para responder a estos desafíos?*

En la tabla 1 se muestran 12 prácticas que se han encontrado para A-GSD. A estas prácticas deberían agregarse aquellas que han sido propuestas para equipos ágiles (pero no distribuidos) o equipos distribuidos (pero no ágiles). Las prácticas que se han encontrado ponen especial énfasis en la comunicación. Aunque estas prácticas proponen ideas interesantes, es necesario un mayor desarrollo de las mismas. También es necesaria su prueba en entornos reales a través de técnicas de validación de casos de estudios u otros métodos científicos.

Tabla 1. Lista de prácticas A-GSD encontradas en la revisión.

Prácticas	
Agile Risk Management Practices	Task Coordination in Agile Distributed Project
Electronic Paper Prototyping	Distributed Agile Planning
Distributed Pair Programming	Architecture as a means of overcoming Communication Challenges
Distributed Requirement Engineering	Coordinating mechanisms in Agile GSD

Prácticas	
Test-Driven Development in Large Projects	Social Software into Distributed Agile Development Environments
Architecture-Based Development	Requirements Elicitation Using A Spatial HypertextWiki

- *RQ3: ¿Existen herramientas que soporten estas prácticas y metodologías?*

El 92% de las herramientas han sido encontradas en repositorios de código y no en publicaciones científicas. Así mismo, el 78% de las herramientas encontradas no presentaba ningún tipo de relación con prácticas existentes. Esto denota que, como ha sucedido en otras áreas de los métodos ágiles, la comunidad de profesionales ha evolucionado de forma paralela (pero separada) a la comunidad científica. Esta diferencia se hace evidente en el hecho de que la mayoría de las herramientas encontradas no se basan en prácticas o desafíos presentados en estudios científicos.

- *RQ3.1: ¿Cuáles de estas herramientas han sido específicamente concebidas para entornos A-GSD?*

De los 32 trabajos que proponen herramientas, sólo 14 de ellos proponen herramientas para A-GSD (no sólo Ágiles o sólo GSD). Además, todas las herramientas que se han encontrado en realidad son adaptaciones de herramientas ágiles o GSD a A-GSD. Por último ninguna de las herramientas que se encuentran en los repositorios de código se ha basado en trabajos previos de investigación y no han sido evaluadas formalmente.

3 Caracterización del conocimiento en A-GSD

La cantidad y diversidad de trabajos encontrados en la revisión nos ha llevado a plantear la necesidad de unificar el conocimiento como paso previo a la utilización del mismo. La organización de conocimiento existente a través de formulaciones ontológicas se ha consolidado como práctica habitual en los últimos años en diversas ramas de la ingeniería de software [6]. La ontología construida está organizada en cuatro sub-ontologías. En la figura 2 se presentan las cuatro sub-ontologías identificadas. Con el fin de facilitar su utilización práctica, hemos decidido establecer los tipos de relación existentes entre las sub-ontologías. Estas relaciones serán analizadas en profundidad en trabajos futuros con el fin de identificar las buenas prácticas aplicables a cada situación.

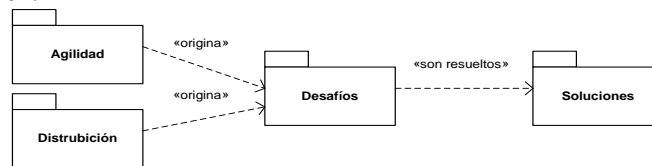


Fig. 2. Ontología de A-GSD y sus sub-ontologías.

Para cada una de las sub-ontologías se ha construido un modelo UML y una tabla de descripción de conceptos. Por cuestiones de espacio sólo hemos incluido el modelo correspondiente a la sub-ontología Distribución y una descripción de las demás sub-

ontologías. Los modelos completos y sus tablas de descripciones se encuentran disponibles en www.kybele.es/ontologiaAGSD.htm.

Sub-Ontología Distribución. En la figura 3 se presenta la ontología relacionada a la distribución. Esta sub-ontología engloba los principales conceptos relacionados a la distribución. En ella se presentan las características particulares de este tipo de entornos (tipo de distancia y del tipo de proyecto) así como también los elementos que pueden ser distribuidos (personas o entornos de trabajo).

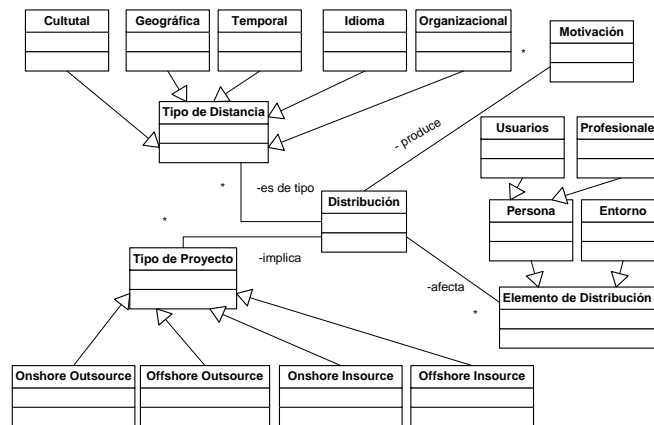


Fig. 3. Modelo UML de la sub-ontología Distribución.

Sub-Ontología Agilidad. En esta sub-ontología se encuentran los conceptos relacionados a la utilización de métodos ágiles en el ámbito de A-GSD. En esta sub-ontología se han identificado los conceptos que en general aplican a todas las metodologías, prácticas ágiles y a los principios del manifiesto ágil [1].

Sub-Ontología Desafíos. En esta sub-ontología se modelan los conceptos relacionados a los problemas o desafíos concernientes al ámbito de A-GSD. La correcta identificación de los mismos permitirá proponer las mejores soluciones: prácticas y herramientas. Estos conceptos permiten identificar los problemas a los cuales se enfrentan los equipos A-GSD y a categorizarlos (tipo de desafío) en desafíos culturales, de comunicación, relacionados a las distintas zonas horarias, de colaboración o técnicos.

Sub-Ontología Soluciones. Las soluciones que pueden ser prácticas (como programación de a pares distribuida o desarrollo dirigido por pruebas distribuido); o metodologías (agrupación de prácticas por ejemplo Scrum distribuido); o herramientas (representan las herramientas tecnológicas).

4 Conclusiones y Trabajos Futuros

En este trabajo hemos presentado los resultados de la revisión de la literatura realizada sobre A-GSD y la caracterización del conocimiento que hemos realizado a partir de

ella. La dispersión de la información existente en este campo de la ingeniería del software, unida al incremento de la actividad académica y empresarial de los últimos años dificulta a los profesionales tener una visión completa sobre la problemática. En consecuencia es difícil para éstos tomar decisiones sobre las mejores prácticas, metodologías o herramientas a utilizar en proyectos A-GSD. El principal aporte de este trabajo es brindar a los profesionales una versión unificada de todo el conocimiento existente en este campo y permitirá a los profesionales el análisis de las variables relevantes que afectan a este tipo de proyectos. Por otro lado, el haber construido la caracterización del conocimiento como una ontología semi-formal con modelos UML facilitará su adopción por parte de la comunidad de profesionales. Además, servirá como base para su implantación en un lenguaje de ontologías formal como OWL. En futuras publicaciones se presentará la ontología completa desarrollada con el lenguaje OWL y un análisis de las relaciones existentes entre desafíos y soluciones. Estas relaciones permitirán la construcción de reglas de aplicación de mejores prácticas y herramientas para distintos escenarios A-GSD. Estas reglas serán validadas a través de casos de estudio con experiencias en equipos de desarrollo reales.

Agradecimientos. Trabajo en el marco de los proyectos MODEL CAOS (Ref. TIN2008-03582) y Agreement Technologies (CONSOLIDER CSD2007-0022).

Referencias

1. Manifiesto for Agile Software Development, En www.agilemanifesto.org, Utah, 2001
2. Fabio Q.B. da Silva, Catarina Costa, A. César C. França, Rafael Prikladinicki. Challenges and Solutions in Distributed Software Development Project Management: a Systematic Literature Review. 5th IEEE International Conference on Global Software Engineering, 2010.
3. Samireh Jalali, Claes Wohlin. Agile Practices in Global Software Engineering – A Systematic Map. 5th IEEE International Conference on Global Software Engineering, 2010.
4. J. Ågerfalk , Brian Fitzgerald: Flexible and distributed software processes: old petunias in new bowls, Communications of the ACM, 2006
5. K. Dullemond, B. van Gameren, R. van Solingen, How Technological Support Can Enable Advantages of Agile Software Development in a GSE setting”, Fourth IEEE International Conference on Global Software Engineering, 2009, pp. 143-152.
6. Francisco Ruiz, José R. Hilera: Using Ontologies in Software Engineering and Technology, Ontologies for Software Engineering and Software Technology, Pages 49-102, 2006
7. B Ramesh, L. C.: Can distributed software development be agile? Communications of the ACM, Vol. 49, No. 10 (2006)
8. Lanubile, F., Ebert, C., Prikladnicki, R., Vizcaino, A.: Collaboration Tools for Global Software Engineering, IEEE Software, Vol. 27, Issue: 2, pp. 52 -- 55, 2010
9. Manifiesto for Agile Software Development, www.agilemanifesto.org, Utah, 2001
10. Abbattista, F.; Calefato, F.; Gendarmi, D.; Lanubile, F.: Incorporating social software into distributed agile development environments, Automated Software Engineering - Workshops, 2008. ASE Workshops 2008. 23rd IEEE/ACM, Page(s): 46 - 51, 2008
11. Solis, C.; Ali, N.: Distributed Requirements Elicitation Using A Spatial HypertextWiki, IEEE ICGSE 2010, Page(s): 237 - 246, 2010