

Docencia en Desarrollo Global de Software: Una Revisión Sistemática

Miguel J. Monasor, Aurora Vizcaíno, Mario Piattini

Grupo de Investigación Alarcos

Instituto de Tecnologías y Sistemas de Información

Escuela Superior de Informática, Universidad de Castilla-La Mancha

13071, Ciudad Real, España

MiguelJ.Monasor@gmail.com, {Aurora.Vizcaino, Mario.Piattini}@uclm.es

Resumen

En este artículo presentamos los resultados de una Revisión Sistemática de la Literatura en el campo de la educación y el entrenamiento de habilidades convenientes para el Desarrollo Global de Software (DGS). Nuestro objetivo consiste en recopilar y estudiar las diferentes propuestas y estrategias empleadas en este campo que sean de utilidad tanto para profesionales como para investigadores y que permitan identificar las mejores prácticas que se deben cubrir para afrontar los problemas que conlleva el DGS.

Summary

In this paper we present the results of a Systematic Literature Review in the field of Global Software Development (GSD) training and education. Our aim is to collect and study the various proposals and strategies employed in this field that may be useful for both professionals and researchers and to identify best practices that should be covered to address the problems that GSD entails.

Palabras clave

Desarrollo Global de Software, Desarrollo Distribuido de Software, Enseñanza, Educación, Revisión Sistemática de la Literatura.

1. Introducción

El Desarrollo Global de Software (DGS) es un paradigma emergente que consiste en que los miembros involucrados en el desarrollo de software permanecen geográficamente distribuidos mas allá de las fronteras de un país [16]. La principal razón de su aplicación radica en que de este modo se optimizan los recursos y se

reducen los costes a través de la expansión hacia zonas más viables económicamente y donde existe una mayor disponibilidad de profesionales cualificados.

Sin embargo, este tipo de desarrollo también conlleva ciertas desventajas, principalmente debidas a la distancia que separa a los equipos, así como las diferencias temporales, culturales y de lenguaje. Estos inconvenientes frecuentemente dificultan el entendimiento entre los participantes del proyecto, especialmente cuando éstos deben usar un lenguaje común (no nativo), pudiendo surgir malentendidos que afectan a la comunicación y la coordinación del trabajo y que podrían suponer un riesgo para el proyecto [25].

Frecuentemente, los responsables de las empresas de desarrollo de software indican que los recién licenciados carecen de las habilidades necesarias para abordar los nuevos problemas que implica el DGS. Argumentan que su experiencia se limita estrictamente a proyectos relativamente cortos, y que los programas educativos no se ocupan de estos temas a un nivel apropiado [10].

Con el objetivo de proporcionar a los estudiantes experiencias reales que les permitan desarrollar las habilidades técnicas, destrezas y competencias requeridas en DGS, son necesarios nuevos contenidos teóricos y herramientas, así como coordinación entre universidades distantes [32]. Sin embargo estos requisitos rara vez son tenidos en cuenta hasta el momento debido a las dificultades que conllevan [14].

Nuestra línea de investigación se centra en el diseño de métodos y técnicas de enseñanza adecuadas para el entrenamiento de los miembros involucrados en DGS. En este trabajo presentamos los resultados de una Revisión Sistemática de la Literatura (RSL) en el campo de la educación y el entrenamiento del DGS con el

objetivo de ofrecer a profesionales, investigadores y educadores una visión rigurosa de los principales retos, estrategias y propuestas disponibles hasta la fecha.

2. Procedimiento de la Revisión Sistemática

Una RSL permite identificar, evaluar e interpretar todos los estudios relevantes disponibles en relación con una pregunta de investigación, tema o fenómeno, de acuerdo con una estrategia predefinida. En este trabajo hemos aplicado el método para realizar RSLs proporcionado por Kitchenham y Charters [18] con la intención de responder a la siguiente pregunta de investigación:

¿Cuáles son las iniciativas llevadas a cabo en relación con la educación y el entrenamiento del Desarrollo Global de Software?

El objetivo consiste en identificar los mejores procedimientos, modelos y estrategias utilizados en la formación y el entrenamiento de los ingenieros de software. Para ello, se estableció la siguiente cadena de búsqueda:

("distributed software development" OR "global software development" OR "global software engineering" OR "distributed software engineering") AND ("learning" OR "teaching" OR "education" OR "training" OR "simulation" OR "simulator").

La estrategia de búsqueda se basó en las siguientes decisiones:

- **Fuentes de búsqueda:** La cadena de búsqueda se adaptó a los siguientes motores de búsqueda:
 - o Science@Direct (www.sciencedirect.com)
 - o SpringerLink (www.springerlink.com)
 - o IEEE Digital Library (www.computer.org)
 - o ACM Digital Library (<http://portal.acm.org>)
 - o Wiley Interscience (www.interscience.wiley.com)
- **Tipo de estudios considerados:** artículos de conferencias, artículos de revistas y talleres.
- **Periodo de publicación:** Desde el año 2000. Puesto que el DGS es una tendencia reciente como se muestra en la revisión sistemática realizada en [38], donde el 74 % de los artículos encontrados eran posteriores a 2006.

2.1. Selección de estudios primarios

Los criterios de inclusión para la selección de los estudios primarios son los siguientes:

- Estudios que describen cursos de DGS en entornos universitarios o empresas.
- Estudios que describen experiencias reales, problemas o factores de éxito.
- Estudios que proponen herramientas de formación o entornos para llevar a cabo el proceso de formación.

Se excluyeron los estudios que cumplían los siguientes criterios:

- Estudios que no responden de manera rigurosa a la pregunta de investigación. Estudios que no aportan ninguna propuesta o información relevante relativas a la formación del DGS o que no enmarcan rigurosamente su propuesta en el contexto del DGS.

Tras la selección inicial de estudios primarios, se llevó a cabo la evaluación de su calidad en dos etapas. Primero se revisó su idoneidad, teniendo en cuenta los criterios de inclusión y exclusión, y después se llevó a cabo una revisión más precisa en paralelo con el proceso de extracción de información. Durante este proceso, se verificó la relevancia y la calidad de los estudios, teniendo en cuenta la claridad de sus métodos y propuestas.

A través del procedimiento de búsqueda (detallado en la Figura 1) se encontraron 38 estudios primarios, listados en la Tabla 1.

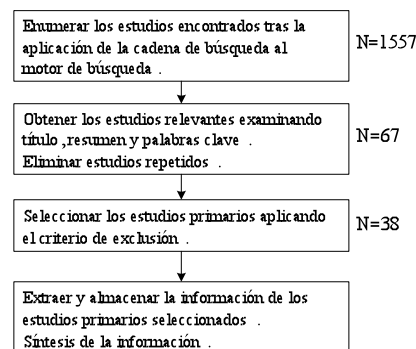


Figura 1. Proceso de selección de estudios primarios

Para cada estudio primario se aplicó un proceso de extracción de información a través de un formulario de datos predefinido con la siguiente información: título, autores, referencia, año, país

de los investigadores, origen, número de páginas, alcance, propuesta, tipo de la organización (universidad, empresa) y tamaño de la empresa, procesos cubiertos, población objetivo, fecha de la evaluación y metodología seguida.

Ref.	Fuente	Ref.	Fuente	Ref.	Fuente
[5]	IEEE	[20]	IEEE	[24]	ACM
[11]	IEEE	[17]	ACM	[23]	IEEE
[8]	IEEE	[1]	ACM	[34]	ACM
[9]	IEEE	[7]	IEEE	[39]	ACM
[21]	IEEE	[41]	IEEE	[27]	ACM
[19]	IEEE	[30]	IEEE	[12]	IEEE
[35]	ACM	[14]	IEEE	[28]	IEEE
[3]	IEEE	[26]	ACM	[40]	IEEE
[31]	IEEE	[29]	ACM	[37]	Wiley
[10]	IEEE	[22]	IEEE	[6]	ACM
[2]	IEEE	[32]	IEEE	[15]	IEEE
[13]	Springer Link	[42]	Springer Link	[4]	Springer Link
[33]	Springer Link	[36]	Springer Link		

Tabla 1. Lista de estudios primarios

3. Tendencias en la Investigación de la educación del DGS

En esta sección se analiza el contenido y las características de los estudios primarios encontrados. La metodología de los estudios primarios se clasificó atendiendo a las siguientes categorías: casos de estudio, revisiones de la literatura, experimentos, simulaciones y encuestas. También se aplicó el modelo no experimental para clasificar los estudios que presentan propuestas sin realizar pruebas experimentales.

La Figura 2 muestra que la mayoría son casos de estudio que básicamente describen experiencias en cursos universitarios. No se encontraron revisiones de la literatura en la materia.



Figura 2. Tipo de estudios analizados

La mayoría de los estudios primarios se contextualizan en un entorno universitario, y básicamente describen cómo grupos de estudiantes han llevado a cabo desarrollos conjuntos desde localizaciones distantes. A pesar de esto, es interesante destacar que también se han encontrado algunos enfoques llevados a cabo en empresas así como estudios desarrollados en colaboración entre universidades y empresas.

3.1. Tendencia de las publicaciones

Atendiendo a los principales procesos de desarrollo hacia los que se han enfocado los estudios primarios, detallados en la Tabla 2, se desprende que los mayores esfuerzos se centran principalmente en la construcción de software, diseño de software, ingeniería de requisitos y pruebas de software.

Procesos	Estudios primarios
Construcción del software	[34], [24], [5], [10], [29], [28], [31], [32], [40], [9], [1], [23],
Diseño del software	[34], [5], [10], [28], [31], [32], [40], [9], [1], [14], [20], [19]
Análisis de requisitos	[34], [5], [10], [28], [31], [32], [9], [14]
Pruebas del software	[34], [6], [5], [28], [40], [9], [15]
Gestión del proyecto	[5], [26], [28], [31], [9], [1], [42]
Elicitación de requisitos	[34], [5], [10], [28], [33], [14]
Gestión organizacional	[8], [31], [4], [36], [37]
Gestión de la configuración	[24], [29], [32], [20]
Aseguramiento de la calidad	[26], [29], [28], [13]
Documentación	[34], [29], [1]

Tabla 2. Principales procesos tratados

4. Resultados de la RSL

En esta sección, se sintetizan los desafíos, métodos y propuestas identificadas a través de la RSL, y se discuten los estudios más relevantes.

4.1. Entornos de aprendizaje

Algunos de los estudios primarios seleccionados tratan sobre entornos de aprendizaje que

proporcionan funcionalidades para la formación y el entrenamiento en actividades típicas del DGS.

iBistro [8] es un entorno que, básicamente permite a los miembros distribuidos colaborar en proyectos y aprender conceptos de gestión de proyectos, desarrollo de software y habilidades sociales. iBistro trata los problemas de comunicación entrenando a los estudiantes a través de reuniones informales en las que pueden capturar las estructuras y los conocimientos surgidos de dichas reuniones. Para ello, se apoya en una herramienta que almacena la información contextual de las reuniones y permite representarlas y navegar por su contenido.

En [35] se presenta un entorno colaborativo utilizado en un curso de desarrollo distribuido de software. Dicho entorno utiliza espacios virtuales donde se lleva a cabo la colaboración del equipo. Estos espacios virtuales pueden contener páginas con contenidos y dentro de ellas, los participantes pueden interactuar a través de diferentes canales de comunicación (como chat o correo electrónico).

En [27] se presenta una plataforma colaborativa basada en Web que facilita la comunicación y gestión de contenidos, proporcionando foros de discusión, un repositorio de archivos compartido y un calendario de proyecto. Los instructores pueden añadir módulos formativos, y los estudiantes pueden acceder a las instrucciones, hitos y resultados finales trabajando con sus compañeros de equipo para cumplir los objetivos marcados.

En la misma línea, [40] presenta una propuesta orientada a la enseñanza mediante la colaboración de equipos de estudiantes distribuidos que interactúan haciendo uso de herramientas colaborativas (incluyendo chat, pizarra compartida, herramienta de compartición de aplicaciones, herramientas de diseño de documentos UML, etc.). Los autores también presentan un software de gestión de cursos para ayudar a los profesores en tareas relacionadas con la administración de grupos y la recogida de información de las acciones del estudiante, proporcionando así un medio para evaluarlos.

Con el fin de estandarizar el trabajo diario del proyecto y evitar la ausencia de miembros en reuniones y otros problemas causados por la informalidad típica en estos entornos algunos estudios aplican la idea del **contrato de equipo**

[11], en el que se establecen unas normas relativas a la comunicación, los tiempos de respuesta y responsabilidades.

Finalmente, también hemos encontrado la aplicación de enfoques de *e-learning* como OAS!S [39]; un entorno de enseñanza virtual creado a través de la personalización de WebCT Vista, cuyo uso está ampliamente extendido en muchas universidades. Permite foros de discusión, sistemas de correo, chat y gestión de contenidos. En este sentido, también debemos mencionar la plataforma de aprendizaje OLAT (*Online Learning And Training*) [39], que proporciona características similares al anterior ofreciendo soporte a varios estándares de e-learning como IMS o SCORM.

4.2. Enseñanza del DGS en el aula

Los estudios primarios también describen clases teóricas tradicionales adaptadas a las nuevas necesidades de la Ingeniería de Software [20]. Muchos de ellos coinciden en destacar la necesidad de que los cursos se realicen de manera conjunta entre diferentes universidades de forma que los estudiantes puedan interactuar con alumnos de diferentes culturas [29]. Sin embargo, un problema común de este enfoque radica en la dificultad de lograr un nivel adecuado de coordinación y colaboración con las diferentes universidades [3].

En el Máster europeo en DGS descrito en [20] participaron varias universidades de diferentes países y básicamente se centra en las dimensiones técnicas y culturales del DGS. En este tipo de prácticas se debe tener en cuenta que los estudiantes de diferentes universidades tienen diferentes tipos de formación, aptitudes y experiencia, por lo que es importante analizar su educación previa y caracterizarlos con el fin de preparar los contenidos del curso [19].

Algunos estudios combinan la enseñanza en las aulas con prácticas y desarrollos reales. El desarrollo de las aptitudes necesarias en DGS requiere práctica en los problemas típicos que se pueden presentar en entornos reales, por lo que el enfoque más propuesto consiste en aprender a través de la práctica.

Algunas universidades tienden a organizar actividades prácticas en colaboración con universidades de diferentes países en las que los estudiantes se suelen comunicar mediante correo

electrónico, teléfono y mensajería instantánea [17] para afrontar procesos similares a los aplicados en la industria.

En la experiencia presentada en [31], participaron tres universidades de diferentes países. El estudio destaca la importancia del desarrollo de las habilidades comunicativas informales, incidiendo en que es necesario aprender a trabajar eficazmente en equipo y reaccionar rápidamente a los cambios en los requisitos, la arquitectura y la organización.

En [39] se presenta el desarrollo de dos proyectos en los que participan estudiantes de cuatro países. Durante esta experiencia se estudió el rendimiento del equipo a través de la medición de ciertos factores que afectan al trabajo colaborativo. Igualmente, en el caso presentado en [28] los estudiantes se dividieron en pequeños grupos (en algunos casos distribuidos en diferentes países), que colaboraron para desarrollar una aplicación completa.

En [13], los autores presentan un curso sobre aseguramiento de la calidad aplicado a tres universidades de diferentes países que colaboraron en el desarrollo de un software. Esta práctica permitió a los estudiantes interactuar con compañeros de diferentes culturas, y desempeñar diferentes roles en el proyecto. Los autores también sugieren que los instructores deberían desempeñar el papel de jefes de proyecto; lo que les permitiría guiar a los estudiantes a lo largo del proceso y aprovechar su experiencia para evitar ciertos problemas de coordinación.

4.3. Enseñanza del DGS en la empresa

Aunque la docencia de la enseñanza de DGS en entornos empresariales no es común, los estudios primarios, detallan algunas experiencias. El ejemplo más destacable es el presentado en [30], consistente en una iniciativa de una multinacional que aplica DGS e impartió un curso de formación práctico relacionado con prácticas comunicativas, diferencias culturales, coordinación y confianza entre los participantes. Después de la finalización de cada proyecto, los estudiantes generaban un documento sobre las lecciones aprendidas.

El mayor grado de disponibilidad de personal con experiencia en estos entornos hace posible la aplicación del concepto de **redes de aprendizaje** [13]. Dado que los instructores no pueden ser expertos en todas las áreas del DGS y que en

consecuencia no pueden cubrir todas las áreas, estas redes proporcionan una nueva forma de realizar el proceso de aprendizaje basándose en el conocimiento de un conjunto multidisciplinar de formadores con experiencia.

En [22] se describe un ejemplo de una red de aprendizaje en la que un equipo de expertos en desarrollo de software de una compañía combinaron su trabajo con actividades de formación aprovechando sus experiencias como ingenieros, jefes de proyecto, responsables de calidad, etc., en dicha compañía.

5. Competencias necesarias en DGS

Una vez analizados los enfoques utilizados en la actualidad para la enseñanza del DGS se consideró conveniente estudiar cuales eran las competencias que se pretendían desarrollar en dichos enfoques. Se obtuvieron las siguientes:

- Conocimiento de protocolos y costumbres de las diferentes culturas implicadas en la comunicación [32], [14].
- Capacidad para comunicarse eficientemente usando una terminología y lengua común [42].
- Habilidad para ganarse la confianza del equipo [42] y para resolver conflictos [27].
- Conocimiento de técnicas de negociación y redacción de contratos [10].
- Gestión de la ambigüedad e incertidumbre que derivan de los problemas comunicativos [32].
- Uso de herramientas de gestión del conocimiento, gestión de documentos y de control de versiones [8], [1].
- Capacidad para liderar las reuniones y gestionar el tiempo [22], [28].
- Habilidad para trabajar en equipo y pensar desde la perspectiva del interlocutor [11].
- Capacidad de improvisación y habilidad para la comunicación informal [31].
- Habilidad para tratar con un equipo multidisciplinar [9].

6. Conclusiones y Trabajo Futuro

En este trabajo se ha realizado una RSL con el objetivo de analizar rigurosamente los estudios relacionados con la formación y el entrenamiento del DGS que nos ha llevado a obtener las siguientes conclusiones:

Conclusión 1. En los últimos años existe un creciente interés en la docencia del DGS.

Conclusión 2. La docencia del DGS debe apoyarse en experiencias prácticas a través de las que los estudiantes puedan aprender mientras trabajan en tareas reales.

Conclusión 3. Para las universidades es difícil simular la complejidad de los entornos reales. Coordinar a estudiantes distantes y con diferentes horarios es uno de los principales desafíos.

Conclusión 4. Los instructores no pueden conocer a fondo todos los problemas que se presentan en las diferentes etapas del DGS [22], por lo que se deben especializar en un campo específico.

Conclusión 5. Los estudiantes que participan en programas de formación de DGS suelen experimentar una falta de motivación, problemas para ajustarse a los horarios y dificultades de comunicación [35], y esto se ve acentuado cuando aparecen diferencias culturales y de lenguaje [11].

Conclusión 6. Son necesarias herramientas específicas para la comunicación, colaboración y gestión de documentos. Una adecuada selección de herramientas es por tanto un aspecto clave [10].

Por otra parte, creemos que puesto que el DGS es un área extensa, este estudio podría ampliarse buscando enfoques relacionados que, a pesar de no abordar este tema directamente, se enfoquen hacia ámbitos tales como el aprendizaje de idiomas y las diferencias culturales, por lo que su estudio sería importante en un trabajo futuro.

Este trabajo se enmarca dentro del inicio de nuestra investigación relacionada con la docencia del DGS. El trabajo inmediato consistirá en analizar las deficiencias de estos enfoques para tratar de dar solución a algunas de ellas a través de la propuesta de un marco metodológico para la docencia del DGS.

Como parte de esta metodología, se pretende además desarrollar una herramienta que ofrezca soporte para que los estudiantes puedan desempeñar lo distintos roles que pueden surgir en un proyecto de DGS (por ejemplo, jefe de proyecto, analista, diseñador, implementador, etc.) de esta forma podrán enfrentarse a casi todas las posibles situaciones que pueden aparecer en este tipo de desarrollo.

La información recogida en este trabajo será también útil para diseñar escenarios de entrenamiento apropiados.

Agradecimientos

Este trabajo está financiado por el proyecto PEGASO/MAGO (Ministerio de Ciencia e Innovación MICINN y Fondos FEDER, TIN2009-13718-C02-01). También por los proyectos MEVALHE (HITO-09-126) y ENGLOBAS (PII2109-0147-8235), financiados por la Consejería de Educación y Ciencia (Junta de Comunidades de Castilla-La Mancha), y Fondos FEDER, así como por GLOBALIA (PEII11-0291-5274) (Consejería de Educación y Ciencia, Junta de Comunidades de Castilla-La Mancha) y ORIGIN (IDI-2010043 (1-5)) financiado por CDTI y FEDER.

Referencias

- [1] Adya, M., D. Nath, A. Malik, and V. Sridhar, *Bringing global sourcing into the classroom: experiential learning via software development project*, *Conference on Computer personnel research: The global information technology workforce*. 2007, ACM: St. Louis, Missouri, USA. p. 20-27.
- [2] Ahamed, S.I. *Model for Global Software Engineering Project Life Cycle and How to Use it in Classroom for Preparing Our Students for the Globalization*. in *COMPSAC '06*. 2006. Chicago, Illinois, USA.
- [3] Bellur, U., *An Academic Perspective on Globalization in the Software Industry*, in the *30th Annual International Computer Software and Applications Conference*. 2006, IEEE CS. p. 53-54.
- [4] Berkling, K., M. Geisser, T. Hildenbrand, and F. Rothlauf, *Offshore Software Development: Transferring Research Findings into the Classroom*, in *Software Engineering Approaches for Offshore and Outsourced Development*, S. Berlin, Editor. 2007: Heidelberg. p. 1-18.
- [5] Bmegge, B., A.H. Dutoit, R. Kobylinski, and G. Teubner, *Transatlantic project courses in a university environment*, in the *Seventh Asia-Pacific Software Engineering Conference*. 2000, IEEE CS. p. 30-37.
- [6] Bondi, A.B. and J.P. Ros, *Experience with Training a Remotely Located Performance Test Team in a Quasi-agile Global Environment*, in *ICGSE '09 - 2009*, IEEE CS. p. 254-261.

- [7] Bouillon, P., J. Krinke, and S. Lukosch, *Software Engineering Projects in Distant Teaching*, in *Proceedings of the 18th Conference on Software Engineering Education & Training*. 2005, IEEE CS. p. 147-154.
- [8] Braun, A., A.H. Dutoit, A.G. Harrer, and B. Brüge, *iBistro: A Learning Environment for Knowledge Construction in Distributed Software Engineering Courses*, in *Proceedings of the Ninth Asia-Pacific Software Engineering Conference*. 2002, IEEE CS. p. 197-203.
- [9] Burnell, L.J., J.W. Priest, and J.R. Durrett, *Teaching Distributed Multidisciplinary Software Development*. IEEE Softw., 2002. **19**(5): p. 86-93.
- [10] Damian, D., A. Hadwin, and B. Al-Ani, *Instructional design and assessment strategies for teaching global software development: a framework*, in *Proceedings of the 28th international conference on Software engineering*. 2006, ACM: Shanghai, China. p. 685-690.
- [11] Favela, J. and F. Peña-Mora, *An Experience in Collaborative Software Engineering Education*. IEEE Softw., 2001. **18**(2): p. 47-53.
- [12] Gotel, O., V. Kulkarni, C. Scharff, and L. Neak, *Integration Starts on Day One in Global Software Development Projects*, in *ICGSE'08*. 2008, IEEE CS. Bangalore, India. p. 244-248.
- [13] Gotel, O., V. Kulkarni, C. Scharff, and L. Neak, *Students as Partners and Students as Mentors: An Educational Model for Quality Assurance in Global Software Development*, S. Berlin, Editor. 2009: Heidelberg. p. 90-106.
- [14] Gotel, O., V. Kulkarni, C. Scharff, and L. Neak, *Working Across Borders: Overcoming Culturally-Based Technology Challenges in Student Global Software Development*, in *Proceedings of the 2008 21st Conference on Software Engineering Education and Training*. 2008, IEEE CS. p. 33-40.
- [15] Hackett, M., *Building Effective Global Software Test Teams through Training*, in *ICGSE'07*. 2007, IEEE CS. Munich, Germany. p. 293-294.
- [16] Herbsleb, J.D. and D. Moitra, *Global software development*. IEEE Software, 2001. **18**(2): p. 16-20.
- [17] Honig, W.L. and T. Prasad, *A classroom outsourcing experience for software engineering learning*, in *SIGCSE'07*. 2007, ACM: Dundee, Scotland. p. 181-185.
- [18] Kitchenham, B. and S. Charters, *Guidelines for performing Systematic Literature Reviews in Software Engineering*, in *Technical Report EBSE-2007-001*. 2007, Keele University and Durham University Joint Report.
- [19] Lago, P., H. Muccini, and M.A. Babar, *Developing a Course on Designing Software in Globally Distributed Teams*, in *ICGSE'08*. 2008, IEEE CS. Bangalore, India. p. 249-253.
- [20] Lago, P., et al., *Towards a European Master Programme on Global Software Engineering*, in *Proceedings of the 20th Conference on Software Engineering Education & Training*. 2007, IEEE CS. p. 184-194.
- [21] Liu, X. *Collaborative global software development and education*. in *COMPSAC'05*. 2005. Edinburgh, Scotland.
- [22] Lutz, B., *Training for Global Software Development in an International "Learning Network"*, in *ICGSE'07*. 2007, IEEE CS. p. 140-150.
- [23] Mead, N.R., A. Drommi, D. Shoemaker, and J. Ingalsbe. *A Study of the Impact on Students Understanding Cross Cultural Differences in Software Engineering Work*. in *33rd Annual IEEE International Computer Software and Applications Conference*. 2009. Seattle, Washington, USA.
- [24] Meneely, A. and L. Williams, *On preparing students for distributed software development with a synchronous, collaborative development platform*, in *the 40th ACM technical symposium on Computer science education*. 2009, ACM: Chattanooga, TN, USA. p. 529-533.
- [25] Monasor, M.J., M. Piattini, and A. Vizcaíno, *Challenges and Improvements in Distributed Software Development: A Systematic Review*. *Advances in Software Engineering*, 2009: p. 14.

- [26] Murphy, C., D. Phung, and G. Kaiser, *A distance learning approach to teaching eXtreme programming*. SIGCSE Bull., 2008. **40**(3): p. 199-203.
- [27] Ocker, R., M.B. Rosson, D. Kracaw, and S.R. Hiltz, *Training Students to Work Effectively in Partially Distributed Teams*. Trans. Comput. Educ., 2009. **9**(1): p. 1-24.
- [28] Petkovic, D., G. Thompson, and R. Todtenhoefer, *Teaching practical software engineering and global software engineering: evaluation and comparison*. SIGCSE Bull., 2006. **38**(3): p. 294-298.
- [29] Petkovic, D., G.D. Thompson, and R. Todtenhoefer, *Assessment and comparison of local and global SW engineering practices in a classroom setting*, in *Proceedings of the 13th annual conference on Innovation and technology in computer science education*. 2008, ACM: Madrid, Spain. p. 78-82.
- [30] Prikladnicki, R. and L. Pilatti, *Improving Contextual Skills in Global Software Engineering: A Corporate Training Experience*, in *ICGSE'08*. 2008, IEEE C S. Bangalore, India. p. 239-243.
- [31] Richardson, I., A.E. Milewski, N. Mullick, and P. Keil, *Distributed development: an education perspective on the global studio project*, in *the 28th international conference on Software engineering*. 2006, ACM: Shanghai, China. p. 679-684.
- [32] Richardson, I., et al., *Globalizing Software Development in the Local Classroom*, in *Proceedings of the 20th Conference on Software Engineering Education & Training*. 2007, IEEE CS. p. 64-71.
- [33] Romero, M., A. Vizcaíno, and M. Piattini, *Using Virtual Agents for the Teaching of Requirements Elicitation in GSD*, in *the international conference on Intelligent Virtual Agents*. 2008, Springer-Verlag: Tokyo, Japan. p. 539-540.
- [34] Rusu, A., et al., *Academia-academia-industry collaborations on software engineering projects using local-remote teams*, in *the 40th ACM technical symposium on Computer science education*. 2009, ACM: Chattanooga, TN, USA. p. 301-305.
- [35] Schümmer, T., S. Lukosch, and J.M. Haake, *Teaching distributed software development with the project method*, in *Conference on computer support for collaborative learning: learning 2005: the next 10 years!* 2005, International Society of the Learning Sciences: Taipei, Taiwan. p. 577-586.
- [36] Setamanit, S.-o. and D. Raffo, *Identifying Key Success Factors for Globally Distributed Software Development Project Using Simulation: A Case Study*, in *Making Globally Distributed Software Development a Success Story*, S. Berlin, Editor. 2008: Heidelberg. p. 320-332.
- [37] Setamanit, S.-o., W. Wakeland, and D. Raffo, *Using simulation to evaluate global software development task allocation strategies*. Software Process: Improvement and Practice, 2007. **12**(5): p. 491-503.
- [38] Silva, F.Q.B.d., C. Costa, A.C. C., and R. Prikladnicki. *Challenges and Solutions in Distributed Software Development Project Management: a Systematic Literature Review*. in *ICGSE '10*. 2010. Princeton, NJ.
- [39] Swigger, K., et al., *Structural factors that affect global software development learning team performance*, in *47th annual conference on Computer personnel research*. 2009, ACM: Limerick, Ireland. p. 187-196.
- [40] Swigger, K., et al. *Teaching Students How to Work in Global Software Development Environments*. in *International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2006. CollaborateCom 2006*. 2006. Atlanta, GA, USA.
- [41] Swigger, K., et al., *A Comparison of Team Performance Measures for Global Software Development Student Teams*, in *ICGSE'09*. 2009, IEEE CS. Limerick, Ireland. p. 267-274.
- [42] Toyoda, S., M. Miura, and S. Kunifuji, *A Case Study on Project-Management Training-Support Tools for Japanese/Chinese/Indian Offshore Development Engineers*, in *Knowledge-Based Intelligent Information and Engineering Systems, S. Berlin, Editor*. 2009: Heidelberg. p. 1222-1229.