Delft University of Technology
Software Engineering Research Group
Technical Report Series

Insights from a structured literature review (SLR) on documented case-studies of Scrum application in globally distributed settings

Peter van Buul and Rini van Solingen

Report TUD-SERG-2016-008

TUD-SERG-2016-008

Published, produced and distributed by:

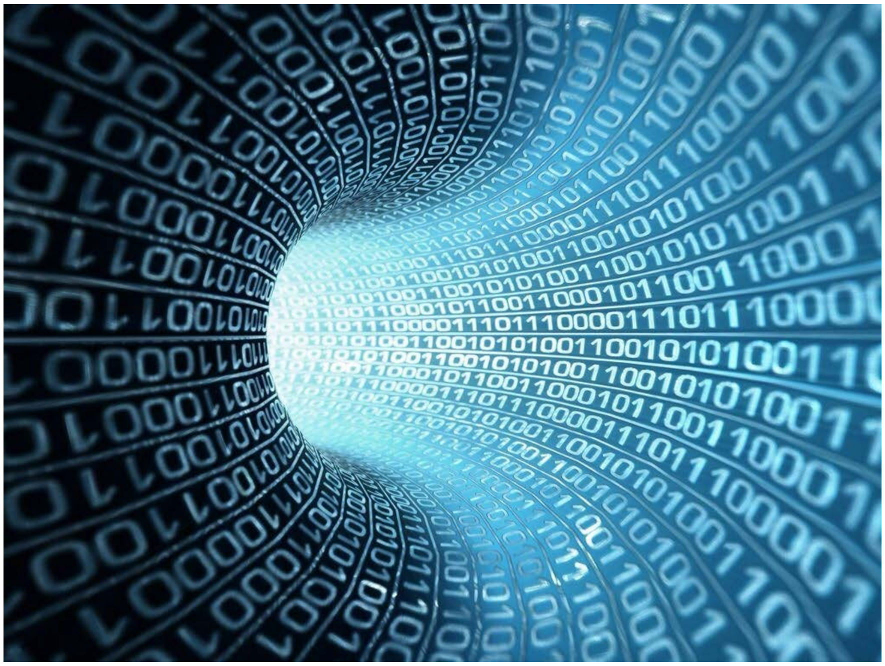
Software Engineering Research Group
Department of Software Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4
2628 CD Delft
The Netherlands

ISSN 1872-5392

Software Engineering Research Group Technical Reports:
http://www.se.ewi.tudelft.nl/techreports/

For more information about the Software Engineering Research Group:
http://www.se.ewi.tudelft.nl/

Insights from a structured literature review (SLR) on documented case-studies of Scrum application in globally distributed settings



Peter van Buul

Rini van Solingen

TU Delft Delft University of Technology

ABSTRACT

Over the past decade the rise of both Globally Distributed Software Engineering and Agile has resulted in a new research area: Distributed Agile Development. One of the most used agile frameworks is Scrum. This systematic literature review (SLR) combines the results of scientifically published case-studies regarding Scrum in a distributed setting. This way, insights are gained into the practical results of research done on Scrum in Distributed Agile Development. Most problems are caused by cultural differences, incorrect application of Scrum, and little or no communication and coordination. To solve these problems solutions have been found of which most come down to: correct application of Scrum and more communication and coordination.

Table of Contents

1. Introduction

This report presents a systematic literature review (SLR) on the topic of Scrum in Distributed Agile Development, also known as Distributed Scrum. Agile Development originates from The Agile Manifesto written by Fowler [1] in which the four core principles of agile have been described. In this systematic literature review the use of Scrum, an implementation of the Agile Manifesto, in a globally distributed setting is researched through reviewing scientifically published case-studies.

First, the need for this review will be identified, the research questions presented and the review protocol discussed. Second, the main study will be presented in which the sub research questions will be answered. Third, the gathered data will be discussed to answer the main research question. Fourth, the summary and conclusions will be presented, followed by recommendations for future research. Finally, there will be a reflection on this systematic literature review.

2. Review protocol

In this chapter the need for a review and the review protocol are discussed. The structure of this chapter is based on the procedures and guidelines by Kitchenham and Charters in [2] and [3].

2.1. The need for a systematic review

In the field of Distributed Agile Development much research has been done by conducting industrial case-studies. The focus of these case-studies differs depending on the study. The study can focus on tools used, communication, cultural differences or any other aspect of Globally Distributed Software Engineering. Therefore, an overview of what these case-studies have concluded and how these conclusions relate is beneficial to the academic community and possibly others. This review combines the different results and provides such an overview.

2.2. Review commissioning

This SLR was done in the research chair on Global Software Engineering, in the Software Engineering research group of the Software Technology department in the Faculty Electrical Engineering, Mathematics and Computer Science of Delft University of Technology and was not commissioned.

2.3. Research questions

This review is written to gain insight in the use of Scrum in a globally distributed setting. The research question that will be answered in this review is:

- What insights can be gained from scientifically published case-studies on using Scrum in globally distributed settings?

This question can be divided into four questions that need to be answered in sequence in order to systematically derive conclusions. These questions are:

- What problems are presented in the different scientifically published case-studies on using Scrum in a globally distributed setting?
- What solutions do these studies propose for the problems?
- Are there correlations between the different case-studies?
- Can new insights be gained based on these correlations?

2.4. Research protocol

Based on the review protocol proposed by [2] and [3] this section describes the protocol that has been used in this systematic literature review.

2.4.1. Search strategy

As the area of research, agile in Globally Distributed Software Engineering, is known as Distributed Agile Development, both terms are used to search the literature. Table 2-1 provides an overview of different search terms that have been used to browse the literature. Different search engines are used: IEEE Xplore, ACM, and ScienceDirect.[1] “IEEE Xplore looks for matches in any of the following fields: document title, publication title, author, Abstract, index terms, and affiliation.” [4] ACM and ScienceDirect search in all fields indexed in the database.

[1] Google Scholar was not used because queries give too many and polluted results to process systematically.

Table 2-1: Search queries

Query
distributed AND Scrum
distributed AND Scrum AND (case study OR case-study OR casestudy)
distributed AND Scrum AND (case study OR case-study OR casestudy) AND development
globally AND distributed AND Scrum AND (case study OR case-study OR casestudy) AND development
globally AND distributed AND Scrum AND (case study OR case-study OR casestudy)
distributed AND agile AND development
distributed AND agile AND development AND Scrum
distributed AND agile AND development AND Scrum AND (case study OR case-study OR casestudy)

2.4.2. Selection criteria

The selection criteria are applied on studies within the field of Globally Distributed Software Engineering. Many different topics exist within this field of research. Any study that discusses agile is considered for this review regardless of what other topics within the field it focuses on.

The following selection criteria were used to determine whether a study should be included in this review:

- The study is published after the year 2000 and before September 1, 2015, because the subject of Globally Distributed Software Engineering is relatively new, therefore papers from 2000 and the review was started in September 2015
- The study is published in the form of a paper, journal article, conference proceeding, report or thesis
- The study is published in English, in order to avoid misinterpretation due to language
- Regarding the content of the studies:
  - The study discusses Scrum in a globally distributed setting
  - The problems that are presented in the study are based on a case-study
  - The conclusions that are presented in the study are based on a case-study
  - The case-study is conducted within a real-life organization

To determine if a study should be excluded the following selection criterion for exclusion have been used:

- The case-study findings are reported coincidental, meaning that the findings or conclusions were outside the scope, goal and focus of the study, and were as such not intend and could be incidents without context.

2.4.3. Selection procedures

Studies are selected based on the criteria mentioned in the previous section.

2.4.4. Quality assessment and checklists

All studies that have been considered are case-studies. Therefore, the quality level of all studies is: “4-2 Evidence obtained from case series, either post-test or pre-test/post-test” according to [2]. To check the quality of the selected studies the checklist for qualitative studies presented in [3] has been used, this checklist can be found in table 2-2. This was done to guide the data analysis, therefore, it has been done during data extraction. Question 17 is omitted as this is not relevant for the subject of this review.

Table 2-2: Checklist for qualitative studies [3]

Number	Question
1	How credible are the findings?
1.1	If credible, are they important?
2	How has knowledge or understanding been extended by the research?
3	How well does the evaluation address its original aims and purpose?
4	How well is the scope for drawing wider inference explained?
5	How clear is the basis of evaluative appraisal?
6	How defensible is the research design?
7	How well defined are the sample design/target selection of cases/documents?
8	How well is the eventual sample composition and coverage described?
9	How well was data collection carried out?
10	How well has the approach to, and formulation of analysis been conveyed?
11	How well are the contexts and data sources retained and portrayed?
12	How well has diversity of perspective and context been explored?
13	How well have detail, depth, and complexity (i.e. richness) of the data been conveyed?
14	How clear are the links between data, interpretation and conclusions - i.e. how well can the route to any conclusions be seen?
15	How clear and coherent is the reporting?
16	How clear are the assumptions/theoretical perspectives/values that have shaped the form and output of the evaluation? What evidence is there of attention to ethical issues?
17	What evidence is there of attention to ethical issues?
18	How adequately has the research process been documented?

When analyzing each case-study, the questions have been considered in order to judge the quality of the study. Studies of too little quality were rejected.

2.4.5. Data extraction strategy

The data extraction form contains the standard information, as described in [3]. Additional data has to be extracted to answer the research questions of this review. For each study the problems and solutions have been extracted. The data extraction forms can be found in table 2-3 and 2-4.

Table 2-3: Data extraction form based on [3]

Data Item	Value	Notes
Date of extraction		
Study title		
Study author(s)		
Study year		
Publication details		

Table 2-4: Data extraction form for problems and solutions

Problem	Solution	Notes

2.4.6. Synthesis of the extracted data

The gathered data regarding problems and solutions are combined to gain additional insights into the data. Problems occurring multiple times have been summarized and grouped together. The same approach is used for the solutions. The data is presented in a table such as table 2-5 and 2-6.

Table 2-5: Table for synthesis of data – Problems - (example data)

Problem	Times mentioned
Problem A	3
Problem B	2

Table 2-6: Table for synthesis of data – Solutions - (example data)

Solution	Times mentioned
Solution A	5
Solution B	3

2.5. Project timetable

The systematic literature review was started in the 1st semester of the academic year of 2015/2016 of the Delft University of Technology. The first meeting regarding this review took place on the September 4, 2015. The review was finished at the beginning of the 2nd semester which is March 21, 2016.

2.6. Research protocol review

The research protocol was reviewed by the second author and adjusted based on feedback. These adjustments are included in the protocol as presented in the previous sections.

3. Main study

In this chapter the results of the systematic literature review are presented. First, the database search is presented. Second, the results of the data extraction are shown. Finally, the synthesis of the extracted data is listed.

3.1. Database search

An overview of the accepted papers as a result of the database search is given in table 3-1.

Table 3-1: Overview accepted papers

Accepted papers	# papers
[5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29]	25

The database search was executed on three databases: IEEE, ACM, and ScienceDirect. Table 3-2 shows an in-depth overview of the results of the database search.

Table 3-2: Results of database search

Query	Database	# hits	# hits on topic	Accepted papers	Rejected papers
distributed AND scrum	IEEE	74	-	Refine search	Refine search
distributed AND scrum AND (case study OR case-study OR casestudy)	IEEE	19	13	[5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16]	[30] [31] [32] [33] [34] [35] [36]
distributed AND scrum AND (case study OR case-study OR casestudy) AND development	IEEE	17	11	[5] [8] [9] [7] [10] [12] [13] [15] [11] [6]	[30] [31] [32] [33] [34] [35] [36]
globally AND distributed AND scrum AND (case study OR case-study OR casestudy) AND development	IEEE	2	1	[11]	[30]
globally AND distributed AND scrum AND (case study OR case-study OR casestudy)	IEEE	2	1	[11]	[30]
distributed AND agile AND development	IEEE	285	-	Refine search	Refine search
distributed AND agile AND development AND scrum	IEEE	59	23	[5] [17] [8] [7] [10] [18] [13] [19] [15] [20] [11] [21] [22] [23] [24] [25] [6] [26] [12] [27]	[37] [38] [39] [40] [41] [30] [35] [42] [43] [44] [45] [46] [31] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] [32] [57] [33] [34] [58] [59] [60] [61] [62] [63] [64] [65] [66]

Query	Database	# hits	# hits on topic	Accepted papers	Rejected papers
					[67] [68] [36]
distributed AND agile AND development AND scrum AND (case study OR case-study OR casestudy)	IEEE	17	10	[5] [8] [7] [10] [13] [15] [11] [6] [12]	[30] [31] [43] [32] [33] [34] [35] [36]
distributed AND scrum	ACM	21	3	[28] [11] [29]	[69] [70] [71] [72] [73] [56] [74] [75] [76] [77] [78] [79] [80] [81] [82] [83] [84] [85]
distributed AND scrum AND (case study OR case-study OR casestudy)	ACM	5	3	[29] [28] [11]	[69] [85]
distributed AND scrum AND (case study OR case-study OR casestudy) AND development	ACM	5	3	[29] [28] [11]	[69] [85]
globally AND distributed AND scrum AND (case study OR case-study OR casestudy) AND development	ACM	2	2	[29] [11]	-
globally AND distributed AND scrum AND (case study OR case-study OR casestudy)	ACM	2	2	[29] [11]	-
distributed AND agile AND development	ACM	121	-	Refine search	Refine search
distributed AND agile AND development AND scrum	ACM	14	2	[28] [11]	[73] [56] [75] [76] [69] [78] [79] [80] [85]
distributed AND agile AND development AND scrum AND (case study OR case-study OR casestudy)	ACM	5	3	[28] [29] [11]	[69] [85]
distributed AND scrum	ScienceDirect	912	-	Refine search	Refine search
distributed AND scrum AND (case study OR case-study OR casestudy)	ScienceDirect	709	-	Refine search	Refine search
distributed AND scrum AND (case study OR case-study OR casestudy) AND development	ScienceDirect	535	-	Refine search	Refine search
globally AND distributed AND scrum AND (case study OR case-study OR casestudy) AND development	ScienceDirect	53	-	Refine search	Refine search

Query	Database	# hits	# hits on topic	Accepted papers	Rejected papers
globally AND distributed AND scrum AND (case study OR case-study OR casestudy)	ScienceDirect	55	-	Refine search	Refine search
distributed AND agile AND development	ScienceDirect	4262	-	Refine search	Refine search
distributed AND agile AND development AND scrum	ScienceDirect	269	-	Refine search	Refine search
distributed AND agile AND development AND scrum AND (case study OR case-study OR casestudy)	ScienceDirect	228	-	Refine search	Refine search

Table 3-3 shows all rejected papers, including their reason for rejection.

Table 3-3: Overview rejected papers

Rejected paper	Reason for rejection
[30]	The study does not report problems on using Distributed Scrum
[31]	The study does not present problems and solutions
[32]	The case-study is not done in a company, but in an academic setting
[33]	The study is not on Scrum
[34]	The company studied in the case-study was not using Scrum
[35]	The study is not on Scrum, but on risk management
[37]	The case-study is not on teams working distributed
[36]	The study does not discuss Distributed Scrum
[38]	The study is on students not on companies
[39]	The case-study is not done in a company, but in an academic setting
[40]	The case-study is not done in a company, but in an academic setting
[41]	The study is on the V-model not Distributed Scrum
[42]	The study is not on Scrum, but on agile
[43]	The study is published after September 1, 2015
[44]	The study discusses a different form from Scrum, namely ScrumUP
[45]	The study is not on Scrum
[46]	The case-study is not done in a company, but in an academic setting
[47]	The study is not on Distributed Scrum, but about subcontracting with Scrum
[48]	The case-study is not done in a company, but in an academic setting
[49]	The study does not report problems on using Scrum in a distributed setting
[50]	The study is not on distributed teams
[51]	No case-study was done for the purpose of the study
[52]	The study is not on Scrum, but on a new framework
[53]	The study is not on Scrum, but is on a tool to support Scrum
[54]	The study is not on Distributed Scrum, but is on human resources with Scrum
[55]	No case-study is presented in the study
[56]	The study is not on Scrum, but on education
[57]	The case-study is not on teams working distributed
[58]	The study is not on Scrum, but on agile
[59]	No case-study was done for the purpose of the study

Rejected paper	Reason for rejection
[60]	The study not on Scrum, but on education
[61]	The case-study is not on teams working distributed
[62]	The study is not on Scrum, but on XP
[63]	The results of the individual case-studies are not discussed, and some of these are not using Scrum
[64]	The case-study was not conducted on a development team
[65]	No case-study is presented in the study
[66]	The study is not on Scrum, but on education
[67]	The study is not on Scrum, but on education
[68]	The study is not on Scrum, but on agile tooling
[69]	The study is published after September 1, 2015
[70]	No case-study is presented in the study
[71]	No case-study is presented in the study
[72]	The study is not on Scrum, but on education
[73]	The study is published after September 1, 2015
[74]	The study is not on Scrum, but on education
[75]	No case-study is presented in the study
[76]	The study is not on Scrum, but on education
[77]	No case-study is presented in the study
[78]	The paper is not a study
[79]	The study is not on Scrum, but discusses agile and .NET
[80]	The study is published after September 1, 2015
[81]	No case-study is presented in the study
[82]	The study is not on Scrum, but on education
[83]	The study is not on Scrum, but on education
[69]	The study is not on Scrum, but on agile group work
[85]	The study is not on Scrum, but on agile
[86]	The study is published after September 1, 2015
[87]	The study is not on Scrum, but on education
[88]	The study is not on Scrum, but on education
[89]	The study is not on Scrum, but on architectural technical debt

3.2. Data extraction

This section answers the research questions: “What problems are presented in the different scientifically published case-studies on using Scrum in a globally distributed setting?” and “What solutions do these studies propose for the problems?”. Table 3-4 contains the basic information which was extracted.

Table 3-4: Data extraction basic info

Ref	Title	Author(s)	Conference	Year
[5]	Using Scrum in distributed agile development: A multiple case-study	Paasivaara, Maria Durasiewicz, Sandra Lassenius, Casper	ICGSE	2009
[6]	Why Scrum works: A case-study from an agile distributed project in Denmark and India	Pries-Heje, Lene Pries-Heje, Jan	AGILE	2011
[7]	Scaling Scrum in a large distributed project	Paasivaara, Maria Lassenius, Casper	ESEM	2011
[8]	Distributed agile development: Using Scrum in a large project	Paasivaara, Maria Durasiewicz, Sandra Lassenius, Casper	ICGSE	2008
[9]	Challenges in Adapting Scrum in Legacy Global Configurator Project	Gupta, Rajeev Kumar Manikreddy, Prabhulinga	ICGSE	2015
[10]	From RUP to Scrum in global software development: A case-study	Noordeloos, Ramon Manteli, Christina van Vliet, Hans	ICGSE	2012
[11]	Inter-team coordination in large-scale globally Distributed Scrum: Do Scrum of Scrums really work?	Paasivaara, Maria Lassenius, Casper Heikkila, Ville T	ESEM	2012
[12]	Adapting to Changes in a Project's DNA: A Descriptive Case-study on the Effects of Transforming Agile Single-Site to Distributed Software Development	Vallon, Raoul Drager, Christopher Zapletal, Alexander Grechenig, Thomas	AGILE	2014
[13]	Doing Scrum Rather Than Being Agile: A Case-study on Actual Nearshoring Practices	Zieris, Franz Salinger, Stephan	ICGSE	2013
[14]	Scaling a running agile fix-bid project with near shoring: Theory vs. reality and (best) practice	Wawryk, Violetta J Krenn, Christian Dietinger, Thomas	ICSTW	2015
[15]	Influences on agile practice tailoring in enterprise software development	Bass, Julian M	AGILE India	2012
[16]	Distributed Agile: Growing a practice together	Vax, Michael Michaud, Stephen	AGILE	2008

Ref	Title	Author(s)	Conference	Year
[17]	Distributed Scrum: Agile Project Management with Outsourced Development Teams	Sutherland, Jeff Viktorov, Anton Blount, Jack Puntikov, Nikolai	HICSS	2007
[18]	Scrum practice mitigation of global software development coordination challenges: A distinctive advantage?	Bannerman, Paul L Hossain, Emam Jeffery, Ross	HICSS	2012
[19]	Yahoo! Distributed Agile: Notes from the World Over	Drummond, Brian Unson, John Francis	AGILE	2008
[20]	Usage of SCRUM Practices within a Global Company	Cristal, Mauricio Wildt, Daniel Prikladnicki, Rafael	ICGSE	2008
[21]	Skiing and Boxing: Coaching Product and Enterprise Teams	Prokhorenko, Sergey	AGILE	2012
[22]	Fully Distributed Scrum: The secret sauce for hyperproductive offshored development teams	Sutherland, Jeff Schoonheim, Guido Rustenburg, Eelco Rijk, Maurits	AGILE	2008
[23]	Fully Distributed Scrum: Linear Scalability of Production between San Francisco and India	Sutherland, Jeff Schoonheim, Guido Kumar, Narinder Pandey, V Vishal, S	AGILE	2009
[24]	From Anarchy to Sustainable Development: Scrum in Less Than Ideal Conditions	Therrien, Isabelle LeBel, Erik	AGILE	2009
[25]	Adapting Agile Methodology to Overcome Social Differences in Project Members	Ozawa, Hitoshi Zhang, Lan	AGILE	2013
[26]	Implementing Scrum in a Distributed Software Development Organization	Anitha, PC Savio, Deepti Mani, VS	AGILE	2013
[27]	From CMMI and Isolation to Scrum, Agile, Lean and Collaboration	Cannizzo, Fabrizio Marcionetti, Gabriela Mose, Paul	AGILE	2009

Ref	Title	Author(s)	Conference	Year
[28]	Software evolution in agile development: a case-study	Sindhgatta, Renuka Narendra, Nanjangud C. Sengupta, Bikram	OOPSLA	2010
[29]	Towards an understanding of tailoring Scrum in global software development: a multi-case-study	Hossain, Emam Bannerman, Paul L. Jeffery, Ross	ICSSP	2011

The tables 3.5 to 3.29 show per paper what problems are presented and solutions are offered.

Table 3-5: Using Scrum in distributed agile development: A multiple case-study

[5] Using Scrum in distributed agile development: A multiple case-study	
Problem	**Solution**
Cultural differences for example about reporting impediments	No solution presented in paper
It takes time to learn to report information that is useful for others	No solution presented in paper
No problem presented in paper	Frequent communication with Scrum practices bring transparency to a distributed project
No problem presented in paper	Frequent communication with Scrum practices reveals problems early on
No problem presented in paper	Daily Scrum creates contacts and encourages informal communication, especially between sites
No problem presented in paper	Weekly Scrum of Scrums distributes information between the teams
No problem presented in paper	Weekly Scrum of Scrums opens discussion channels and encourages informal communication between teams
Different religious or other holidays in different countries cause synchronization challenges	No solution presented in paper
No problem presented in paper	Sprints provide frequent monitoring opportunities between the sites
Time zone differences make it challenging to arrange meetings, especially long ones	No solution presented in paper
Cultural and language differences may cause some participants to be more silent	No solution presented in paper
Long teleconferences are exhausting	No solution presented in paper
Sound quality not always good enough	No solution presented in paper
Difficult to recognize speakers when their faces are not visible	No solution presented in paper
No problem presented in paper	Sprint planning meetings give a possibility for team members from all sites to participate, to ask for clarification, to understand tasks and to commit to shared goals
No problem presented in paper	Sprint demo prevents problems by providing a frequent monitoring opportunity between sites
No problem presented in paper	Sprint demo ensures the understanding of the requirements, especially regarding the offsite
It is unclear who is responsible for updating backlogs	No solution presented in paper
Wiki editor is awkward to use	No solution presented in paper
No problem presented in paper	All team members can access, pick items and follow progress with shared backlogs
No problem presented in paper	Frequent face-to-face visits
No problem presented in paper	Have many different possibilities to communicate between sites
All sites need to understand Scrum	Give all sites a proper Scrum training

Table 3-6: Why Scrum works: A case-study from an agile distributed project in Denmark and India

[6] Why Scrum works: A case-study from an agile distributed project in Denmark and India	
Problem	**Solution**
No problem presented in paper	Daily Scrum of Scrums builds relations and trust
No problem presented in paper	Burn down chart builds relations and trust
No problem presented in paper	Deliverables at end of sprint build relations and trust
No problem presented in paper	The product backlog provides coordination
No problem presented in paper	The sprint backlog provides coordination
No problem presented in paper	The Scrum board provides coordination
No problem presented in paper	The daily Scrum provides coordination
No problem presented in paper	User stories provide help for communication
No problem presented in paper	The product backlog provides help for communication
No problem presented in paper	Visible Scrum Boards provide help for communication
No problem presented in paper	The Product Owner role provides help for communication
No problem presented in paper	The Scrum Master role provides help for communication

Table 3-7: Scaling Scrum in a large distributed project

[7] Scaling Scrum in a large distributed project	
Problem	**Solution**
No problem presented in paper	Area Product Owners
Collaboration and communication between Area Product Owners is not good	Requirement and design workshop before sprint planning
No problem presented in paper	Common sprint planning with representatives
No problem presented in paper	Common sprint demo
No problem presented in paper	Common retrospective

Table 3-8: Distributed agile development: Using Scrum in a large project

[8] Distributed agile development: Using Scrum in a large project	
Problem	**Solution**
No problem presented in paper	Weekly Scrum of Scrums
No problem presented in paper	Synchronized 4-week sprints
No problem presented in paper	Distributed sprint planning meetings
No problem presented in paper	Distributed sprint demo's
No problem presented in paper	Distributed retrospective meetings
No problem presented in paper	Separate backlogs for each team
No problem presented in paper	Unofficial distributed meetings
No problem presented in paper	Visiting engineer during first iteration
No problem presented in paper	On-site system expert
No problem presented in paper	Frequent visits
No problem presented in paper	Annual gathering
No possibility for videoconference - The network	No solution presented in paper

[8] Distributed agile development: Using Scrum in a large project	
Problem	**Solution**
connection between offices is not fast enough for videoconferencing	
Misunderstanding requirements - This is undetected until the demo	Product Owner asks follow up questions
Silence caused by distance - Norwegian team asked many questions, Malaysian team preferred to listen	No solution presented in paper
Silence caused by distance - Norwegian team felt unsure whether the off-site team had really understood everything	No solution presented in paper

Table 3-9: Challenges in Adapting Scrum in Legacy Global Configurator Project

[9] Challenges in Adapting Scrum in Legacy Global Configurator Project	
Problem	**Solution**
Team collaboration when working in different time zones - not everyone involved in status meetings	Daily 20 min Scrum of Scrums
	Weekly 30 min Scrum of Scrums
	Bi-Weekly 60 min Scrum of Scrums
	Dedicated daily collaboration of 45 min
	Monthly peer feedback from each peer in the Scrum team
	Travel to strengthen communication and trust
Working in silos - rarely any knowledge sharing	10-15% of sprint time is reserved for sharing knowledge and learning new skills

Table 3-10: From RUP to Scrum in global software development: A case-study

[10] From RUP to Scrum in global software development: A case-study	
Problem	**Solution**
Legal barrier between the onshore and offshore location, a person from outside the EU is not allowed to access financial systems of a company within the EU	No solution provided, one can only "work with it"
Requirements engineering when "throwing the requirements over the fence" does not allow developers and tester to participate	Using Scrum all team members participate in requirements engineering
	The unified Scrum team ensured only one party was communicating to the team in India
No problem presented in paper	Using Scrum, the daily interactions enable the team to find misunderstandings early in the process
With RUP only certain people communicated with each other using video-conference	Using Scrum all team members have face-to-face meetings which builds trust

Table 3-11: Inter-team coordination in large-scale globally Distributed Scrum: Do Scrum of Scrums really work?

[11] Interteam coordination in large-scale globally Distr. Scrum: Do Scrum of Scrums really work?	
Problem	**Solution**
In the Scrum of Scrums problems are not reported because the reporter assumes it is not relevant	No solution presented in paper
In the Scrum of Scrums problems are not reported because the reporter assumes they will receive no help anyway	No solution presented in paper
Scrum of Scrum meetings are not always useful because information is not relevant for everyone	Feature Scrum of Scrum meetings for the teams working on the same feature

Table 3-12: Adapting to Changes in a Project's DNA: A Descriptive Case-study on the Effects of Transforming Agile Single-Site to Distributed Software Development

[12] Adapting to Changes in a Project's DNA: A Descriptive Case-study on the Effects of Transforming Agile Single-Site to Distributed Software Development	
Problem	**Solution**
Test-ready features at the end of the sprint instead of deployment ready ones	Give test site the right to reject stories
	Increase contact visits
	Review of test cases for critical user stories by a developer
	Constant availability of team members in instant messaging
Transparency for both sites not accomplished	All relevant informal communication needs to be appended to the feature ticket in the issue tracking tool
	Contact visits to improve trust and team spirit
Low software quality and lack of focus in sprint	Incremental inclusion of test site
	Bug fix iteration (should be avoided)
	Sunshine cases should work when a story is passed to the tester so the tester can focus on corner cases
	Build all aspects of a story including non-functional requirements (e.g. stability and performance)
	Cont. deployment to customer every sprint
Volatile specification	No informal story updates
	Meetings with the customer before sprint
	Involve customer more in prioritization
	Small manageable stories
	Increased up-front planning and specification to identify problems, corner cases and impact on existing software early and improve estimation of team
	Staffing: new Business Analyst
No up-to-date test cases	No story updates during sprint
No up-to-date test cases	On-demand specification meetings with members on both sites
Process adaption in DSD is slower and more difficult than in regular collocated Scrum	Use retrospective as a driver for continuous process improvement

Table 3-13: Doing Scrum Rather Than Being Agile: A Case-study on Actual Nearshoring Practices

[13] Doing Scrum Rather Than Being Agile: A Case-study on Actual Nearshoring Practices	
Problem	**Solution**
No problem presented in paper	Code Buddy as a bi-directional proxy to facilitate inter-team communication (1 different team member stay's in the German office for one week)
Differences in interests during retrospective	No solution presented in paper
Different approaches with sprint planning	No solution presented in paper
No problem presented in paper	Daily sync meeting next to daily Scrum for technical details
No problem presented in paper	The War Room – before a story is given to the Product Owner the development team tries to "break" the feature, and thus discover failures early
Multiple Product Owners – Product Owners not really in sync gives a messy road map	No solution presented in paper
Multiple Product Owners – Teams miss intrinsic motivation because of a messy road map	No solution presented in paper
Multiple Product Owners – Lack of responsibility with team because of messy road map	No solution presented in paper

Table 3-14: Scaling a running agile fix-bid project with near shoring: Theory vs. reality and (best) practice

[14] Scaling a running agile fix-bid project with near shoring: Theory vs. reality and (best) practice	
Problem	**Solution**
Product Owner was project manager, therefore stayed with the client	Proxy Product Owner was used to have a Product Owner on site
Due to distance between PO and the team missing information was delaying sprint goals	No solution presented in paper
User stories where in German and had to be explained in English	Explain user stories face-to-face in English
	Explain the German user stories to the German speaking site manager via screen sharing in telephone conferences
	Translate user stories in English
Holidays where not overlapping	No solution presented in paper
Proxy Product Owner is not always present leading to delays and missing information	Proxy Product Owner on site with the client
No problem presented in paper	Team building - Visit the other location
No problem presented in paper	User stories need to be in more detail

Table 3-15: Influences on agile practice tailoring in enterprise software development

[15] Influences on agile practice tailoring in enterprise software development	
Problem	**Solution**
No problem presented in paper	Use sprints to empower and motivate the development team
No problem presented in paper	Daily standup meeting with video conference technologies

Table 3-16: Distributed Agile: Growing a practice together

[16] Distributed Agile: Growing a practice together	
Problem	**Solution**
No problem presented in paper	Use nearshore resources – have people in the same time zone that can meet face-to-face regularly
No problem presented in paper	Have a strict communication plan
No problem presented in paper	Shared electronic workspaces (e.g. Sharepoint)
Having people in three time zones is difficult bordering on insane	Have teams in two time zones and split work accordingly
No problem presented in paper	Hold retrospectives asynchronous and post result in the shared electronic workspace
No problem presented in paper	Have multiple lines of communication
No problem presented in paper	Get the right people and get them talking

Table 3-17: Distributed Scrum: Agile Project Management with Outsourced Development Teams

[17] Distributed Scrum: Agile Project Management with Outsourced Development Teams	
Problem	**Solution**
Difficult leveraging available resources, best practices are often deemed proprietary, time consuming and difficult to maintain	Chief Product Owner and Chief Scrum Master to manage Scrum of Scrums and backlog management centrally
Synchronize work between sites	Write the answers to the daily Scrum meeting down and email this before the daily Scrum
	Scrum masters of all sites meet for a Scrum of Scrums each Monday
	Architects are allocated to the Scrum teams and also meet each Monday
	The chief Product Owner meets regularly with all Product Owners
No problem presented in paper	Resolve questions regarding stories by mail before the daily Scrum

Table 3-18: Scrum practice mitigation of global software development coordination challenges: A distinctive advantage?

[18] Scrum practice mitigation of global software development coordination challenges: A distinctive advantage?	
Problem	**Solution**
Increased coordination costs (due to temporal distance)	Increase overlapping work hours between sites to enable synchronous communication
	Use of tools that enable synchronous communication between teams (e.g. conference calls and video conference)
	Use of tools that enable asynchronous communication between teams (e.g. email , IM)
Reduced informal contact can lead to lack of critical task awareness	Increase overlapping work hours between sites to enable synchronous communication
	Use of tools that enable synchronous communication between teams (e.g. conference calls and video conference)
	Use of tools that enable asynchronous communication between teams (e.g. email and instant messaging)
	Enable frequent communication
	Iterate to monitor progress and resolve issues
	Enable reflections to review the process
	Enable planning activities to scope work and schedule it
Inconsistent work practices can impinge on effective coordination	Increase overlapping work hours between sites to enable synchronous communication
	Use of tools that enable synchronous communication between teams (e.g. conference calls and video conference)
	Use of tools that enable asynchronous communication between teams (e.g. email and instant messaging)
	Enable frequent communication
	Enable planning activities to scope work and schedule it
Reduced cooperation arising from misunderstandings	Increase overlapping work hours between sites to enable synchronous communication
	Use of tools that enable synchronous communication between teams (e.g. conference calls and video conference)
	Use of tools that enable asynchronous communication between teams (e.g. email and instant messaging)
	Enable frequent communication
	Iterate to monitor progress and resolve issues
	Enable reflections to review the process
	Enable planning activities to scope work and schedule it
Reduced cooperation arising from misunderstandings	Visit the other site to enable face-to-face meetings

Table 3-19: Yahoo! Distributed Agile: Notes from the World Over

[19] Yahoo! Distributed Agile: Notes from the World Over	
Problem	**Solution**
Difference in dinnertime caused one location to plan meetings for the other location during dinner time without knowing so	Quarterly meetings with everybody present
	Talk about cultural differences
Problems are not shared; one location only shares the positive things	Have retrospectives with local teams
Meetings with everyone present with different time zones	Do the meetings locally
Having meetings outside office hours	Time box those (local) meetings
Information sharing in the hallway is lost	Be aware of the loss of informal information sharing in the hallway and compensate with other practices
Keeping track of progress	Share pictures of the physical Scrum boards on a wiki
Having to work late at night makes one location feel as being second hand	Alternate the schedule so everyone has to work late at night at some point
	Have a 3rd party present at retrospectives
The team that gets to pick from the backlog first gets all the interesting stories	Put stories on the backlog for specific teams based on local business value
Waiting long time for the Product Owner to reply for a simple question	Have someone who regularly syncs with the Product Owner who can answer those questions for that location (a proxy PO)
Succeeding in sprints is difficult	Do not follow Scrum practices religiously
Losing communication bandwidth (tone and emotion) due to the usage of video conferencing and phone calls	No solution presented in paper
Having to be at the office at late hours for meetings because the hardware is not available at home	Have the senior members update one another with a call from time to time

Table 3-20: Usage of SCRUM Practices within a Global Company

[20] Usage of SCRUM Practices within a Global Company	
Problem	**Solution**
Lack of formal document for requirements	No solution linked in paper
Inadequate team structure	No solution linked in paper
Communication barriers resulting in a lack of team commitment	No solution linked in paper
Management not used to agile practices	No solution linked in paper
No problem linked in paper	Use cases should be integrated with user stories
No problem linked in paper	While piloting agile practices the team must document Scrum meeting minutes
No problem linked in paper	Scrum master needs to be a strong negotiator
No problem linked in paper	Assigning testers and developers to work together
No problem linked in paper	Use a “global” task board

Table 3-21: Skiing and Boxing: Coaching Product and Enterprise Teams

[21] Skiing and Boxing: Coaching Product and Enterprise Teams	
Problem	**Solution**
No problem presented in paper	Specialized coaching at each location
Resistance to communicating long term plans to development teams	Communicate long term vision to maintain team commitment
No problem presented in paper	Use a Proxy Product Owner
No problem presented in paper	Successful Product Owners communicate bilateral

Table 3-22: Fully Distributed Scrum: The secret sauce for hyperproductive offshored development teams

[22] Fully Distributed Scrum: The secret sauce for hyperproductive offshored development teams	
Problem	**Solution**
No problem presented in paper	Sprint planning done with whole team present
No problem presented in paper	Standup done with whole team present at start of the Dutch work day
No problem presented in paper	Retrospective done with whole team present
No problem presented in paper	Demo done with customer and Dutch members, Indian members briefed after
No problem presented in paper	Daily Scrum of Scrums with Scrum Masters after standup
Cultural differences	Travel to develop personal relationships
	Team culture aimed at openness and direct communication
	Company culture of openness with an equal value
Difficult to fully communicate client nuances, context and priorities	Regular traveling
	Always-open Skype connections
	A project news gazette after every iteration
	Informal updates by the Product Owner
Managing customers new to agile	Add a project manager to manage the client and deliver a clear backlog
No problem presented in paper	Have a local collocated team to clear roadblocks that can only be done by a local team
Having the right tools for communication and collaboration	Electronic shared backlogs and a wiki to share information and documentation

Table 3-23: Fully Distributed Scrum: Linear Scalability of Production between San Francisco and India

[23] Fully Distributed Scrum: Linear Scalability of Production between San Francisco and India	
Problem	**Solution**
No problem presented in paper	Start with a collocated period
No problem presented in paper	Sprint planning prepared separately
No problem presented in paper	Sprint planning done with whole team
No problem presented in paper	Standup once a week done with whole team
No problem presented in paper	Daily standup of Indian team with proxy member of US team

[23] Fully Distributed Scrum: Linear Scalability of Production between San Francisco and India	
Problem	**Solution**
No problem presented in paper	Retrospective done with whole team
No problem presented in paper	Demo done with whole team
No problem presented in paper	Shared wiki updated at end of day
Hard to plan demos with all stakeholders present	No solution presented in paper
Shared designing hard without overlapping hours	No solution presented in paper
No budget for traveling	No solution presented in paper

Table 3-24: From Anarchy to Sustainable Development: Scrum in Less Than Ideal Condition

[24] From Anarchy to Sustainable Development: Scrum in Less Than Ideal Condition	
Problem	**Solution**
No problem presented in paper	Organize face-to-face quarterly meetings
No problem presented in paper	Favor small teams of 4 to 6 over larger teams
No problem presented in paper	Use video conferencing over email and other forms of communication
No problem presented in paper	Do not be afraid to make mistakes, but be ready to implement changes when you see something is not working

Table 3-25: Adapting Agile Methodology to Overcome Social Differences in Project Members

[25] Adapting Agile Methodology to Overcome Social Differences in Project Members	
Problem	**Solution**
Difference in Openness of society	No solution presented in paper
Difference in willingness to adopt new techniques and technology	No solution presented in paper
Difference in communication resulting in miscommunication	No solution presented in paper
Misunderstanding between the Product Owner and development team	Assign a Proxy Product Owner with in debt understanding of both cultures

Table 3-26: Implementing Scrum in a Distributed Software Development Organization

[26] Implementing Scrum in a Distributed Software Development Organization	
Problem	**Solution**
Coordinating between four time zones	Have a daily Scrum of Scrums meeting with another higher level Scrum of Scrums meeting above it
Planning with really big teams	Have one big backlog
No problem presented in paper	Multiple Product Owners
Hard to ensure that the product satisfies the market needs and has the required documentation and has the best quality	Create a validation team that defines customer configurations
	Create a documentation team that handles documentation
	Create an integration and automation team that ensures a stable environment for all teams

Table 3-27: From CMMI and Isolation to Scrum, Agile, Lean and Collaboration

[27] From CMMI and Isolation to Scrum, Agile, Lean and Collaboration	
Problem	**Solution**
Requirements written for solution domain which is not understood	Move the Product Owner to the same room as the developers and make part of the team
Documented requirements only for the "tip of the ice berg"	Move the Product Owner to the same room as the developers and make part of the team
Increasing time needed for integrating code	Move the Product Owner to the same room as the developers and make part of the team
No problem presented in paper	Global structure, working with clearly distributed daily Scrum and planning to manage the work focus and boundaries
No problem presented in paper	Establishing technical and business feedback rhythms
No problem presented in paper	One global technical infrastructure
No problem presented in paper	Continuously improving development practices
No problem presented in paper	Requirements management and agile planning
No problem presented in paper	Training the teams in knowing the problem domain
No problem presented in paper	Cross culture understanding with more collaboration and onsite visits
No problem presented in paper	Communication protocols

Table 3-28: Software evolution in agile development: a case-study

[28] Software evolution in agile development: a case-study	
Problem	**Solution**
Coordinating the continuous changes and refinements of the project	Use well defined mechanisms for collaboration

Table 3-29: Towards an understanding of tailoring Scrum in global software development: a multi-case-study

[29] Towards an understanding of tailoring Scrum in global software development: a multi-case-study	
Problem	**Solution**
No problem presented in paper	Sprint planning modified with pre- and post-planning meetings with onshore team
No problem presented in paper	Sprint planning modified with additional pre-planning meeting
No problem presented in paper	Sprint planning modified, held weekly with project manager and sub-team coordinators
No problem presented in paper	Sprint planning modified with additional pre-planning meeting
No problem presented in paper	Daily Scrum distributed via Skype
No problem presented in paper	Daily Scrum done locally
No problem presented in paper	Two daily Scrums, one locally one distributed
No problem presented in paper	Daily Scrum done locally
No problem presented in paper	Scrum of Scrums distributed done weekly
No problem presented in paper	Scrum of Scrums not used

[29] Towards an understanding of tailoring Scrum in global software development: a multi-case-study	
Problem	**Solution**
No problem presented in paper	Scrum of Scrums done weekly combined with sprint planning and review
No problem presented in paper	Scrum of Scrums distributed done weekly
No problem presented in paper	Sprint review done with onshore team
No problem presented in paper	Sprint review done distributed
No problem presented in paper	Sprint review not done
No problem presented in paper	Sprint review done with only Scrum Masters, project master and customer present
No problem presented in paper	Retrospective not done
No problem presented in paper	Retrospective not done
No problem presented in paper	Retrospective done locally every two weeks
No problem presented in paper	Retrospective done locally every four weeks
No problem presented in paper	Backlog shared between locations
No problem presented in paper	Backlog shared between locations
No problem presented in paper	Backlog shared between locations
No problem presented in paper	Backlog shared between locations

3.3. Synthesis of the extracted data

This section will answer the research questions: “What problems are presented in the different scientifically published case-studies on using Scrum in a globally distributed setting?” and “What solutions do these studies propose for the problems?”.

Table 3-30 shows the metadata of this systematic literature review. Some studies have presented problems without proposing a solution while other papers offer solutions without describing a problem. This visualization provides an overview of the data found in this review.

Table 3-30: Metadata

Metadata	Amount
Number of problems and solutions	245
Number of problems	70
Number of solutions	212
Number of problems with solution	37
Number of problems without solution	34
Number of solutions with problem	99
Number of solutions without problem	113

3.3.1. Problem listing

This section answers the research question: “Are there correlations between the different case-studies?” for the problems found. To answer this question, the problems are grouped and summarized. When a problem occurred more than once, it is summarized and listed in table 3-31. Single occurrences are omitted. How these problems are grouped can be found in Appendix A. The problems that are not grouped can be found in Appendix B.

Table 3-31: Problem groups

Problem group	Times mentioned
Incorrect execution of Scrum When Scrum is not executed correctly this results into problems. It ranges from being unclear who is responsible for the backlog, sites that do not understand Scrum to test cases that are not up to date.	8
Not communicating all information to team Due to the distance between the team and the Product Owner all information has to be conveyed using mail or video conferencing. This makes it difficult to communicate all client nuances, context, and priorities to the development team. This results in volatile requirement documents.	7
Misunderstanding due to cultural differences Misunderstanding due to cultural differences happens often. This is caused by a difference in the way of communicating. Requirements can be misunderstood and cooperation might be less as a result.	6
No syncing between sites Sites working in silos are not synced well, work is not synced and changes during the project are not coordinated.	5
Planning a meeting with everyone present is difficult Time zone differences make it challenging to arrange a meeting with everybody present. The longer the meeting the harder it is to arrange. Results of this is that not everybody is present or meetings are outside office hours.	5
Hardware and tools not sufficient The available hardware and tools are not always sufficient. It is not possible to use video conferencing due to network issues or the sound quality is not good, or the tools are awkward in use.	5
Difference in reporting impediments Depending on cultural background different ways of reporting impediments have been presented. Some cultures tend to not report failures while others do report this. This can result in different interests during a retrospective.	4
Product Owner not present Due to the distance between the team and the Product Owner the Product Owner is not always present. This results into delays because the team has to wait for answer of the Product Owner for small questions.	4
Integration difficulties Integration is difficult due to difference in legislation and a distributed way of working. Integrating code takes more time and it can be problematic to ensure that the product satisfies customer needs.	4
Lack of focus With distributed teams it is hard to keep focus during the sprint resulting in low software quality.	3
Scrum of Scrums not effectively used Scrum of Scrums is considered not useful. Because the reporter assumes that information is not relevant and he does not get help anyway, and because information that is shared is not relevant for everyone.	3
Coordinating in multiple time zones is difficult Coordinating when working in multiple time zones is difficult. This coordination is even called bordering on the insane.	3
Different holidays Different cultures have different holidays. If these do not overlap it results in	2

Problem group	Times mentioned
problems with planning.	
Silence / passivism Due to cultural or language differences some participants can become silent resulting in no communication from those participants.	2
Different work practices Differences in work practices make it hard to coordinate. A result of this is that the outcome of the same meeting is different per location.	2
Multiple Product Owners not in sync When working with Proxy Product Owners it is difficult to keep these Proxy Product Owners in sync. A result of this is that the team has to wait for the Product Owner or might even get misinformation.	2
Informal contact is lost When losing informal contact, information sharing in the hallway is lost. This leads to lack of critical task awareness.	2
Meetings at the office outside office hours Meetings outside office hours are planned due to time zone differences. These meetings sometimes have to be held at the office because the needed hardware is only available at the office.	2
No transparency between sites Transparency between sites is not always sufficient, result of this is that one site can feel unsure whether everything is clear for the other site.	2
Features not being deployment ready at end of sprint Succeeding in sprints is considered difficult when features are test ready at the end of the sprint instead of deployment ready.	2
Managing customers new to agile Customers and managements that are new to agile are not always easily convinced of the benefits of working agile making transition to Distributed Scrum hard.	2
Time differences Because of time differences teams have to work late at night or during dinner time.	2

3.3.2. Solution listing

This section answers the research question “Are there correlations between the different case-studies?” for the solutions. This question is answered in the same way as for the problems; the solution is summarized and only listed if it mentioned more than once. A solution can be mentioned multiple times within a paper as being a solution for different problems, these are counted as different solutions. In table 3-32 the solutions are listed. How these solutions are grouped can be found in Appendix C. The solutions that are not grouped can be found in Appendix D.

Table 3-32: Solution groups

Solution group	Times mentioned
Apply Scrum Applying Scrum in a distributed setting can be challenging but this is also the most named solution. Some Scrum practices have been extracted into different solution groups. However, most are placed in this group. Applying Scrum ranges from the use of user stories to using a retrospective.	24
Travel and meet face to face Traveling between sites with face-to-face visits enables growth of personal relationships. These relationships increase team spirit, communication and trust.	17
Different communication channels The use of different communication channels ensures that the sites can contact the other site in the way appropriate for their question. This enables quick answering of small questions with synchronous communication while bigger questions can be discussed more broadly using asynchronous communication.	14
Non-Scrum practices In some case-studies extra practices that are not in Scrum have been added to smoothen the process and overcome problems. These practices range from mailing the answers of the daily Scrum questions before the daily Scrum starts to reserving sprint time for sharing knowledge.	14
Proxy Product Owners By using Proxy Product Owners, the Product Owners can become part of the team. This way the team does not have to wait for the Product Owner to answer their questions as they can ask their local Proxy Product Owner. These Proxy Product Owners should have an in-depth understanding of the cultures.	12
Use the daily Scrum Using the daily Scrum ensures that the sites have at least one formal contact point. This moment of synchronization between sites enables the teams to find misunderstandings early in the process and keep the sites in sync. The daily Scrum is done both locally and distributed in the case-studies.	11
Shared backlog and resources The use of shared backlogs and other shared resources such as Sharepoint enables the sites to have insight into what the other sites are doing and what has to be done.	11
Weekly Scrum of Scrums Having a weekly Scrum of Scrums opens communication channels between teams. Participants in the Scrum of Scrums can be the Scrum Masters or architects.	10
Frequent communication Communicating frequently between sites brings transparency to a distributed project and can reveal problems early on. Sprints provide frequent communication opportunities between sites.	10
Sprint planning with all sites present The sprint planning is often done with all sites present. This enables team members from all sites to participate, ask for clarification, to understand tasks and commit to common goals.	10
Non-Scrum meetings Many studies add extra meetings such as pre- and post-sprint planning	9

Solution group	Times mentioned
meetings. These meetings are used to overcome problems due to working distributed.	
Meetings with local team Doing meetings with the local team makes it easier to plan and coordinate the meeting. Results of the local meetings can then be shared in electronic workspaces.	7
Iterate / use sprints Using an iterative way of working, in which at the end of each iteration working software is delivered, and continuous feedback and improvement is enabled. This empowers the team to make its own decisions which motivates the team to work hard.	7
Successful team composition Creating a successful team ensures that the project is working good. Different practices that help to create a successful team are for example: communicate long term vision, favoring small teams of 4 to 6 people and having a Scrum Master who is a good negotiator.	7
Sprint demo with all sites present Doing a sprint demo is a Scrum practice, doing the demo with all sites present provides a frequent opportunity for monitoring between sites. This monitoring enables the sites to check whether the requirements were understood.	6
Increase overlapping hours By increasing the number of overlapping hours there is more time when synchronous communication is possible. This can be done by using an alternating schedule in which all sites have to work late at night at some point.	5
Retrospective with all sites present Same as for the demo, the retrospective is a Scrum practice, doing this with all sites present enables the sites to synchronize by detecting and solving problems.	4
Daily Scrum of Scrums Doing a daily Scrum of Scrums ensures that there is a daily sync moment between sites. This can feel as a lot of overhead but it does help keeping the sites in sync.	4
Sprint planning prepared in separate meeting By preparing the sprint planning with the local teams in a separate meeting many questions can be answered beforehand so the focus during the sprint planning can be on planning the user stories.	4
Give all sites Scrum training All sites need to have the same understanding of Scrum. To achieve this sites with not enough knowledge can get Scrum training before the project is started.	3
Scrum Board and burn down charts Scrum boards and burn down charts visualize progress and results. This visualization helps for communication and can help build relations and trust.	3
Use universal language to increase clarity Using a universal language (e.g. English) for documented communication (e.g. user stories) between the local team members enables another site to join a	3

Solution group	Times mentioned
project which is already in progress regardless of their native language.	
Communication strategy Having a communication strategy with protocols ensures that all sites know what communication channel to use when they want to contact another site. This also ensures that the site that is contacted knows what kind of questions can be expected on what channel.	3
Separate backlogs for each team Having a backlog per team enables the teams to function separately without having to consult the other teams when picking up new stories.	2
Retrospective not done Planning a retrospective with all sites can be challenging, and if all sites are present there can be too many people present to handle the retrospective properly. To avoid this, it is possible to not do retrospectives. However, this does remove the feedback mechanism for teams. Teams are no longer provided with an opportunity to discover problems, solve them and improve the team.	2

3.3.3. Problem and solution classification

To provide more insight in the answer of the research question: “Are there correlations between the different case-studies?”, the problem and solution groups are classified in more abstract terms. These problem and solution classes are used to gain insight into what general problems and solutions are found.

For each problem group the root cause defines how it is classified. The following six problem classes are defined as in table 3-33.

Table 3-33: Problem classes

Class	Number of related groups
Coordination	6
Culture	5
Scrum	3
Communication	3
Time zone	3
Technical	1

The communication and coordination classes may seem similar as most coordination problems also have issues with communication. Problems for which the cause is both communication and coordination are put in the coordination class as the root cause for the problem is then coordination. Table 3-33 shows that most problems are due to coordination and cultural issues.

Table 3-34: Problem group classification

Problem group	Class
No syncing between sites	Coordination
Planning a meeting with everyone present is difficult	Coordination
Integration difficulties	Coordination
Lack of focus	Coordination

Problem group	Class
Coordinating in multiple time zones is difficult	Coordination
Multiple Product Owners not in sync	Coordination
Misunderstanding	Culture
Difference in reporting impediments	Culture
Different holidays	Culture
Silence / passivism	Culture
Different work practices	Culture
Incorrect execution of Scrum	Scrum
Scrum of Scrums not effectively used	Scrum
Features not being deployment ready at end of sprint	Scrum
Managing customers new to agile	Scrum
Not communicating all information to team	Communication
Product Owner not present	Communication
Informal contact is lost	Communication
Meetings at the office outside office hours	Time zone
No transparency between sites	Time zone
Time differences	Time zone
Hardware and tools not sufficient	Technical

Solution classes are defined by the general manner in which their members solve a problem. For example, using sprints is a Scrum solution to a problem while Proxy Product Owners are a non-Scrum solution. The following four solution classes are defined as in table 3-35.

Table 3-35: Solution classes

Class	Number of related groups
Scrum	10
Coordination	7
Non-Scrum	4
Communication	4

Looking at table 3-35 most solutions presented are using Scrum and coordination, but also Non-Scrum and communication are used often as a solution.

Table 3-36: Solution group classification

Solution group	Class
Apply Scrum	Scrum
Proxy Product Owners	Scrum
Use the daily Scrum	Scrum
Weekly Scrum of Scrums	Scrum
Sprint planning with all sites present	Scrum
Iterate / use sprints	Scrum
Sprint demo with all sites present	Scrum
Retrospective with all sites present	Scrum
Daily Scrum of Scrums	Scrum
Scrum Board and burn down charts	Scrum

Solution group	Class
Shared backlog and resources	Coordination
Meetings with local team	Coordination
Successful team composition	Coordination
Increase overlapping hours	Coordination
Give all sites Scrum training	Coordination
Communication strategy	Coordination
Separate backlogs for each team	Coordination
Non-Scrum practices	Non-Scrum
Non-Scrum meetings	Non-Scrum
Sprint planning prepared in separate meeting	Non-Scrum
Retrospective not done	Non-Scrum
Travel and meet face to face	Communication
Different communication channels	Communication
Frequent communication	Communication
Use universal language to increase clarity	Communication

Based on the problem and solution groups, problem classes have been defined. These are different from the problem classes presented by Kajko-Mattson, Azizyan and Magarian in [90]. They found the following seven problem classes:

- Culture
- Time Zone
- Communication
- Trust
- Customer Collaboration
- Training
- Technical

The culture, time zone, technical and communication classes are found in both studies. The problem class coordination found in this study would be communication according to [90]. However, coordination issues are not necessarily due to communication. Therefore, it was split from communication. The classes customer collaboration, training and trust from [90] where all based on a single problem.

The customer collaboration problem corresponds to the Product Owner not present group which is classified as a communication problem. The training problem would be due to lack of Scrum experience which is classified as a Scrum problem because it is incorrect execution of Scrum. The trust problem described in [90] had as root cause the lack of face to face communication, this would therefore be classified as a communication problem. Thus, all problems found can be described by using the six problem classes presented in this systematic literature review.

4. Solution and problem linking

This chapter answers the research question: “Can new insights be gained based on these correlations?”. This is done by linking problem groups to solution groups. If a problem in a problem group is linked to a solution in a solution group these groups are linked.

Using these links new insight is gained into what solutions can be applied when different problems are encountered.

4.1. Solutions per problem

In this chapter each problem and the corresponding solutions are presented.

4.1.1. Incorrect execution of Scrum

The most common problem with Distributed Scrum is the incorrect execution of Scrum. Being a common problem there are also many different solutions for this problem. Giving all sites Scrum training and applying Scrum clearly solves the problem. Using separate backlogs for each team enables teams to function independently as intended in Scrum. Using a non-Scrum meeting, an on demand specification meeting with members on both sites can solve the problem of missing up-to-date test cases. This is similar to doing the sprint planning with all sites present. Using a shared backlog and shared resources can solve the problem, as can communicating frequently.

4.1.2. Not communicating all information to team

The main solution for the problem of not communicating all information to the team is communicating. Specifically, traveling to the other sites so teams can meet face to face, by using multiple communication channels and communicating frequently. Application of Scrum helps communicating all information and adding Proxy Product Owners ensures the Product Owner is nearby for questions. Non-Scrum meetings that are applied to solve this are meeting with the customer before the sprint and increased up-front planning and specification.

4.1.3. Misunderstanding due to cultural differences

Solving misunderstandings due to cultural differences is done by focusing on communication. Increasing overlapping work hours to increase contact is a possible solution. Using multiple communication channels and communicate frequently to ensure more communication. Traveling and meeting face to face so that relationships are developed and communication is easier. Applying Scrum with its iterative way of working, increases the number of contact moments which improves communication.

4.1.4. No syncing between sites

The problem that different sites are not in sync can be solved by more communication. Having a communication strategy between sites ensures it is evident how to communicate between different sites. Creating teams where the long term vision is clear makes sure the teams are committed to the same goal and helps keeping them in sync. Using Proxy Product Owners ensures that keeping them in sync covers for most of the syncing between sites. Using a weekly Scrum of Scrums ensures that the sites are synced weekly. Two non-Scrum practices that help syncing are reserving sprint time for knowledge sharing and learning, and mailing the answers for the daily Scrum before the daily Scrum is started.

4.1.5. Planning a meeting with everyone present is difficult

When planning a meeting with everyone present is difficult, no real solution is offered. All solutions aim to minimize the consequences rather than solving the problem. A solution can be doing the meetings locally which is done in some case-studies. Other case-studies suggest applying Scrum practices such as the daily Scrum and Scrum of Scrums regularly to keep in sync and handle the consequences. Another solution is sharing resources for improved visibility of the tasks of every team, which also tackles the consequences rather than the problem. Traveling and meeting face to face is also proposed to handle the consequences.

4.1.6. Hardware and tools not sufficient

New developments in hardware, for example faster network connections should make the hardware sufficient. However, this is not noted in the case-studies. Finding the right tools can be difficult. The case-studies suggest using a tool with a shared backlog and resources is the best solution.

4.1.7. Difference in reporting impediments

The only solution provided for handling the difference in reporting requirements is to do the retrospectives with the local team.

4.1.8. Product Owner not present

In order to make sure that the Product Owner is present more often a Proxy Product Owner can be used.

4.1.9. Integration difficulties

Integration difficulties can be tackled by using Proxy Product Owners. The Proxy Product Owners can discover integration problems early on and ensure these are solved. Applying the retrospective as a driver for continuous process improvement, enabling the team to discover and solve integration problems themselves rather than waiting for the Proxy Product Owner to discover the problem.

4.1.10. Lack of focus

Creating focus by using Scrum is done by iterations of 2-4 weeks, called sprints. Two non-Scrum practices are also presented as ways to create focus. The first one is making sure that the tester can focus on corner cases by letting the developer test the “sunshine cases”. The second one is including the test site during development. Both these practices are not in accordance with the Agile Manifesto [1].

4.1.11. Scrum of Scrums not effectively used

By doing a feature Scrum of Scrums meeting with only the stakeholders present that are relevant for the feature, ensures that everyone has an interest in the Scrum of Scrums.

4.1.12. Coordinating in multiple time zones is difficult

Coordinating in multiple time zones is difficult. To make this coordination easier, more communication is key. Using multiple communication channels enables more communication both when work hours are overlapping as well as not overlapping. Having more overlapping work hours enables direct communication which makes coordination easier. By dividing the work accordingly over the teams, less coordination is needed. This division has to be done properly so that no dependencies between the sites are created. And lastly, by doing a daily Scrum of Scrums a daily sync moment between sites is present, which improves coordination.

4.1.13. Different holidays

No solution was found to solve the problem of different holidays.

4.1.14. Silence / passivism

No solution was found to solve the problem of silence.

4.1.15. Different work practices

Solving problems which arise due to different work practices can be done by applying Scrum which offers the teams a unified way of working. Similar to the coordination of multiple time zones, more communication is key. Frequent communication through different communication channels with more overlapping work hours helps to minimize the differences in work practices.

4.1.16. Multiple Product Owners not in sync

Syncing multiple Product Owners can be done by doing a requirements and design workshop before the sprint planning.

4.1.17. Informal contact is lost

Be aware of the loss of informal contact and compensate this with Scrum practices. This can be done with the application of Scrum practices, using sprints, retrospectives and planning. Increasing overlapping hours makes room for more informal contact and frequent communication over different channels also stimulates informal contact.

4.1.18. Meetings at the office outside office hours

Communicating frequently can eliminate the necessity of meetings outside office hours. However, if these meetings do occur, applying the Scrum practice of time boxing to those meetings avoids unnecessary overtime.

4.1.19. No transparency between sites

Increasing transparency between sites can be done by traveling and meeting face to face in order to get to know the other sites. Appending all relevant informal communication to the feature ticket provides a better insight for the other site in the informal contact of the site, this results in more transparency.

4.1.20. Features not being deployment ready at end of sprint

In order to get features deployment ready communication is key, travel and meet face to face to develop personal relationships and communicate using different communication channels. Non-Scrum practices like giving a test site the right to reject stories and review of test cases for critical user stories by a developer can be used. These are, however, not in line with The Agile Manifesto [1].

4.1.21. Managing customers new to agile

The only solution provided for managing customers new to agile is the non-Scrum practice of assigning a project manager to that particular client.

4.1.22. Time differences

Solving problems due to working with time differences can be done by increasing the overlapping work hours. Traveling and meeting face to face can also ensure that problems due to time difference are understood by the all sites. To find the problems a 3rd party can be brought in.

4.2. Solutions without problem

Most solutions are used to solve problems, however not for every solution a problem is found. Table 4-1 lists the solutions that are not linked to a problem. All of these solutions, except for using a universal language (e.g. English), are directly related to the execution of Scrum. Therefore, these could also have been grouped under the solution group: “Apply Scrum”. However, because these solutions occurred multiple times they were put in their own groups. Since working with a universal language is expected from an international company regardless of the use of Scrum, no problem regarding Distributed Scrum can be linked.

Table 4-1: Solution groups without problem

Solution group
Sprint demo with all sites present
Retrospective with all sites present
Sprint planning prepared in separate meeting
Scrum Board and burn down charts
Use universal language (e.g. English) to increase clarity
Retrospective not done

5. Conclusion

This chapter contains the summary and conclusions of this systematic literature review as well as recommendations for future research.

5.1. Summary

Applying Scrum in a distributed setting can be challenging. This systematic literature review provides an answer to the following research question: “What insights can be gained from scientifically published case-studies on using Scrum in globally distributed settings?”.

Where some of the reviewed case-studies describe the problems that where encountered, others describe the solutions that they used. In some cases, both the problem and the solution are mentioned. For these last studies the problem and solution groups are linked. This provides a validated link between a problem and a solution. This validated link provides insight into how the problem can be solved.

These insights are summarized in table 5-1.

Table 5-1: Solution Problem linking

Problem group	Solution groups	Solution Class
Incorrect execution of Scrum	Separate backlogs for each team Give all sites Scrum training Shared backlog and resources Apply Scrum Sprint planning with all sites present Non-Scrum meetings Frequent communication	Coordination Coordination Coordination Scrum Scrum Non-Scrum Communication
Not communicating all information to team	Travel and meet face to face Different communication channels Frequent communication Apply Scrum Proxy Product Owners Non-Scrum meetings	Communication Communication Communication Scrum Scrum Non-Scrum
Misunderstanding	Increase overlapping hours Successful team composition Apply Scrum Iterate / use sprints Different communication channels Frequent communication Travel and meet face to face	Coordination Coordination Scrum Scrum Communication Communication Communication
No syncing between sites	Non-Scrum practices Weekly Scrum of Scrums Proxy Product Owners Communication strategy Successful team composition Shared backlog and resources	Non-Scrum Scrum Scrum Coordination Coordination Coordination
Planning a meeting with everyone present is difficult	Apply Scrum Use the daily Scrum Daily Scrum of Scrums Weekly Scrum of Scrums	Scrum Scrum Scrum Scrum

	Travel and meet face to face Meetings with local team Shared backlog and resources	Communication Coordination Coordination
Hardware and tools not sufficient	Shared backlog and resources	Coordination
Difference in reporting impediments	Meetings with local team	Coordination
Product Owner not present	Proxy Product Owners	Scrum
Integration difficulties	Apply Scrum Proxy Product Owners	Scrum Scrum
Lack of focus	Non-Scrum practices Iterate / use sprints	Non- Scrum Scrum
Scrum of Scrums not effectively used	Non-Scrum meetings	Non- Scrum
Coordinating in multiple time zones is difficult	Increase overlapping hours Successful team composition Different communication channels Daily Scrum of Scrums	Coordination Coordination Communication Scrum
Different holidays	No solution found	
Silence / passivism	No solution found	
Different work practices	Increase overlapping hours Different communication channels Frequent communication Apply Scrum	Coordination Communication Communication Scrum
Multiple Product Owners not in sync	Non-Scrum meetings	Non-Scrum
Informal contact is lost	Increase overlapping hours Different communication channels Frequent communication Iterate / use sprints Apply Scrum Non-Scrum practices	Coordination Communication Communication Scrum Scrum Non-Scrum
Meetings at the office outside office hours	Frequent communication Apply Scrum	Communication Scrum
No transparency between sites	Non-Scrum practices Travel and meet face to face	Non-Scrum Communication
Features not being deployment ready at end of sprint	Non-Scrum practices Travel and meet face to face Different communication channels	Non-Scrum Communication Communication
Managing customers new to agile	Non-Scrum practices	Non-Scrum
Time differences	Travel and meet face to face Increase overlapping hours Apply Scrum	Communication Coordination Scrum

5.2. Answer to the research questions

This systematic literature review intended to answer the following question: “What insights can be gained from scientifically published case-studies on using Scrum in globally distributed settings?”. To determine to what extent this question is answered the four subquestions are first evaluated.

The first sub question: “What problems are presented in the different scientifically published case-studies on using Scrum in a globally distributed setting?” has been answered by listing the different problems of the literature in section 3.3.1 and providing an overview of the problems and the amount they are mentioned in the included publications.

The second sub question: “What solutions do these studies propose for the problems?” has also been answered. The different solutions from the literature have been listed in section 3.3.2 and linked to problems in chapter 4 when applicable.

The third sub question: “Are there correlations between the different case-studies?” has been answered by grouping similar problems and solutions in section 3.3.3. These grouped problems and solutions show the correlations between the different case-studies.

The fourth sub question: “Can new insights be gained based on these correlations?” has not been answered explicitly. Though the correlations do give insight into problems that have not been solved in one paper but are solved in another. This does not lead to new insights explicitly, however if the link was not made before, it does provide a new deducted insight. Also some ideas are provided in section 5.4 for future research and table 4-1 contains solutions without a clear problem domain.

5.3. Conclusions

The five most mentioned problems are: incorrect execution of Scrum, not communicating all information to the team, misunderstanding, lack of synchronization between sites, and difficulties when it comes to planning a meeting with everyone present. These are classified as problems with Scrum, communication, culture and coordination, which represent the most common problem classes.

To solve these problems different solutions are provided. Correct application of Scrum is the most used solution. Also, specific Scrum aspects are mentioned, for example Proxy Product Owners and using the daily Scrum. Except for Scrum also frequent communication over different communication channels and traveling between sites are often mentioned as a solution. These solutions emphasize the importance of communication. Increasing coordination can also solve problems by, for example sharing backlogs or increasing overlapping work hours.

Non-Scrum practices are also mentioned to solve problems. However, these tend to avoid problems rather than solving them. Not doing a sprint demo avoids the problem of planning a meeting with everybody present, though this results in less communication and feedback causing even more problems. Not doing a Scrum of Scrums avoids the feeling of useless meetings but also leads to less coordination. A better solution for this is doing a feature Scrum of Scrums, where only the teams working on a specific feature are present.

The problems that are the hardest to solve are cultural problems, since the only two problems without solutions, are classified as such. No solution is found for the problems of different holidays and silence caused by different backgrounds. Solving the other cultural problems can be done by communicating and traveling regularly to meet face to face and develop personal relationships. Though this does not entirely prevent problems due to culture, it does create an understanding between sites which tempers the consequences of the problems.

Altogether, there are multiple ways to solve problems. Which solution should be applied depends on the situation. However, almost any problem is solved by correct application of Scrum and increasing communication and coordination.

5.4. Recommendations for future research

No solution was found for the problems of differences in holidays and passive behavior. Therefore, it is recommended to do additional research regarding these problems to see if the problems can be solved.

The focus of this systematic literature review is on finding problems and their corresponding solutions for Scrum in a globally distributed setting. Result of this focus is that no insight is gained in the results of applying the solutions. Knowing what solution to apply is valuable. However, it is even more valuable if it is also known what results can be expected when the solution is applied. Additional research should therefore be done to map what the results are of applying the different solutions.

The most interesting solutions for this additional research are: regular traveling and meeting face to face, using different communication channels with frequent communication, and the usage of Proxy Product Owners. These are the most mentioned solutions in the case-studies which not differ from Scrum practices.

5.5. Limitations

Doing a systematic literature review limits the research to what is known. In a young field such as Agile and Global Software Engineering, one could wonder if SLR research is able to find much niew insights as the amount of literature will be limited by definition due to the maturity of the research domain itself. Furthermore, answers for the research questions are based on what is found in literature and there is no room to do additional research for unanswered questions. The result of this is that during this review, two problems were found without solution. These problems can be investigated further.

In this systematic literature review, the database search was done with database searches rather than using “backward snowballing” [91]. As stated in [91] the papers that where accepted for the search could be different if a backward snowballing approach had been used. However, the conclusion should remain the same, only differences could be found in the specific problems and solutions. Using both database search techniques would have provided a better overview.

Another limitation to this study relates to the research protocol. Its application differs from person, based on their frame of reference. Small differences in application of the acceptance criteria could lead to different outcomes. Another example is, when extracting the data, different readers might find different problems and solutions in the papers.

6. Acknowledgements

Special thanks to Arend van Buul, Teun van Buul, Liselotte Rambonnet and, Alexander Simes for reviewing initial versions of this Systematic Literature Review report.

Bibliography

[1] M. Fowler and J. Highsmith, "The Agile Manifesto," *Software Development,* vol. 9, no. 8, pp. 28-35, 2001.

[2] B. Kitchenham, "Procedures for performing systematic reviews," *Keele, UK, Keele University,* vol. 33, pp. 1-26, 2004.

[3] B. Kitchenham and S. Charters, Guidelines for performing systematic literature reviews in software engineering, Technical report, Ver. 2.3 EBSE Technical Report (EBSE), 2007.

[4] "IEEE Xplore basic search," IEEE, [Online]. Available: http://ieeexplore.ieee.org/xplorehelp/Help_Using_Basic_Search.html. [Accessed 16 12 2015].

[5] M. Paasivaara, S. Durasiewicz and C. Lassenius, "Using scrum in distributed agile development: A multiple case study," in *Fourth IEEE International Conference on Global Software Engineering (ICGSE)*, 2009.

[6] L. Pries-Heje and J. Pries-Heje, "Why Scrum works: A case study from an agile distributed project in Denmark and India," in *Agile Conference (AGILE'11)*, 2011.

[7] M. Paasivaara and C. Lassenius, "Scaling scrum in a large distributed project," in *International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2011.

[8] M. Paasivaara, S. Durasiewicz and C. Lassenius, "Distributed agile development: Using Scrum in a large project," in *IEEE International Conference on Global Software Engineering (ICGSE)*, 2008.

[9] R. K. Gupta and P. Manikreddy, "Challenges in Adapting Scrum in Legacy Global Configurator Project," in *IEEE 10th International Conference on Global Software Engineering (ICGSE)*, 2015.

[10] R. Noordeloos, C. Manteli and H. Van Vliet, "From RUP to Scrum in global software development: A case study," in *IEEE Seventh International Conference on Global Software Engineering (ICGSE)*, 2012.

[11] M. Paasivaara, C. Lassenius and V. T. Heikkila, "Inter-team coordination in large-scale globally distributed scrum: Do Scrum-of-Scrums really work?," in *ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2012.

[12] R. Vallon, C. Drager, A. Zapletal and T. Grechenig, "Adapting to Changes in a Project's DNA: A Descriptive Case Study on the Effects of Transforming Agile Single-Site to Distributed Software Development," in *Agile Conference (AGILE'14)*, 2014.

[13] F. Zieris and S. Salinger, "Doing Scrum Rather Than Being Agile: A Case Study on Actual Nearshoring Practices," in *IEEE 8th International Conference on Global Software Engineering (ICGSE)*, 2013.

[14] V. J. Wawryk, C. Krenn and T. Dietinger, "Scaling a running agile fix-bid project with near shoring: Theory vs. reality and (best) practice," in *IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, 2015.

[15] J. M. Bass, "Influences on agile practice tailoring in enterprise software development," in *AGILE India (AGILE INDIA)*, 2012.

[16] M. Vax and S. Michaud, "Distributed Agile: Growing a practice together," in *Conference Agile. AGILE'08*, 2008.

[17] J. Sutherland, A. Viktorov, J. Blount and N. Puntikov, "Distributed Scrum: Agile Project Management with Outsourced Development Teams," in *40th Annual Hawaii International Conference on System Sciences, HICSS*, 2007.

[18] P. L. Bannerman, E. Hossain and R. Jeffery, "Scrum practice mitigation of global software development coordination challenges: A distinctive advantage?," in *45th Hawaii International Conference on System Science (HICSS)*, 2012.

[19] B. Drummond and J. F. Unson, "Yahoo! Distributed Agile: Notes from the world over," in *AGILE'08. Conference Agile*, 2008.

[20] M. Cristal, D. Wildt and R. Prikladnicki, "Usage of SCRUM Practices within a Global Company," in *IEEE International Conference on Global Software Engineering (ICGSE)*, 2008.

[21] S. Prokhorenko, "Skiing and Boxing: Coaching Product and Enterprise Teams," in *Agile Conference (AGILE'12)*, 2012.

[22] J. Sutherland, G. Schoonheim, E. Rustenburg and M. Rijk, "Fully distributed scrum: The secret sauce for hyperproductive offshored development teams," in *Conference Agile (AGILE'08)*, 2008.

[23] J. Sutherland, G. Schoonheim, N. Kumar, V. Pandey and S. Vishal, "Fully Distributed Scrum: Linear Scalability of Production between San Francisco and India," in *Agile Conference (AGILE'09)*, 2009.

[24] I. Therrien and E. LeBel, "From Anarchy to Sustainable Development: Scrum in Less Than Ideal Conditions," in *Agile Conference (AGILE'09)*, 2009.

[25] H. Ozawa and L. Zhang, "Adapting Agile Methodology to Overcome Social Differences in Project Members," in *Agile Conference (AGILE'13)*, 2013.

[26] H. Smits and G. Pshigoda, "Implementing Scrum in a Distributed Software Development Organization," in *Agile Conference (AGILE'07)*, 2007.

[27] F. Cannizzo, G. Marcionetti and P. Mose, "From CMMI and Isolation to Scrum, Agile, Lean and Collaboration," in *Agile Conference (AGILE '09)*, 2009.

[28] R. Sindhgatta, N. C. Narendra and B. Sengupta, "Software evolution in agile development: a case study," in *Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion (OOPSLA '10)*, 2010.

[29] E. Hossain, P. L. Bannerman and R. Jeffery, "Towards an understanding of tailoring scrum in global software development: a multi-case study," in *Proceedings of the 2011 International Conference on Software and Systems Process (ICSSP '11)*, 2011.

[30] H.-C. Estler, M. Nordio, C. A. Furia, B. Meyer and J. Schneider, "Agile vs. structured distributed

software development: A case study," *Empirical Software Engineering,* vol. 19, no. 5, pp. 1197-1224, 2014.

[31] T. Niinimäki, "Face-to-face, email and instant messaging in distributed agile software development project," in *Sixth IEEE International Conference on Global Software Engineering Workshop (ICGSEW)*, 2011.

[32] G. Ravanhani Matuck, R. Luiz de Oliveira, T. A. Pivetta, L. A. Vieira Dias and A. M. da Cunha, "Applying Agile Method on Academic Access and Fraud Control System," in *11th International Conference on Information Technology: New Generations (ITNG)*, 2014.

[33] S. Anwar, Y. H. Motlat, Y. Siddiq, S. Asghar, M. S. Hassan and Z. I. Khan, "User-Centered Design Practices in Serum Development Process: A Distinctive Advantage?," in *IEEE 17th International Multi-Topic Conference (INMIC)*, 2014.

[34] S. Beecham, J. Noll and I. Richardson, "Using Agile Practices to Solve Global Software Development Problems -- A Case Study," in *IEEE International Conference on Global Software Engineeering Workshops (ICGSEW)*, 2014.

[35] S. Sundararajan, M. Bhasi and P. K. Vijayaraghavan, "Case study on risk management practice in large offshore-outsourced Agile software projects," *IET Software,* vol. 8, no. 6, pp. 245-257, 2014.

[36] B. Fitzgerald, K.-J. Stol, R. O'Sullivan and D. O'Brien, "Scaling agile methods to regulated environments: an industry case study," in *Proceedings of the 2013 International Conference on Software Engineering*, 2013.

[37] E. V. Woodward, R. Bowers, V. S. Thio, K. Johnson, M. Srihari and C. J. Bracht, "Agile methods for software practice transformation," *IBM Journal of Research and Development,* vol. 54, no. 2, pp. 3:1-3:12, 2010.

[38] C. Scharff, O. Gotel and V. Kul, "Transitioning to Distributed Development in Students' Global Software Development Projects: The Role of Agile Methodologies and End-to-End Tooling," in *Fifth International Conference on Software Engineering Advances (ICSEA)*, 2010.

[39] J. Lin, "Context-aware task allocation for distributed agile team," in *IEEE/ACM 28th International Conference on Automated Software Engineering (ASE)*, 2013.

[40] C. Scharff, "Guiding global software development projects using Scrum and Agile with quality assurance," in *24th IEEE-CS Conference on Software Engineering Education and Training (CSEE\&T)*, 2011.

[41] P. Anitha, D. Savio and V. Mani, "Managing requirements volatility while “Scrumming” within the V-Model," in *IEEE Third International Workshop on Empirical Requirements Engineering (EmpiRE)*, 2013.

[42] R. Phalnikar, V. Deshpande and S. Joshi, "Applying Agile Principles for Distributed Software Development," in *International Conference on Advanced Computer Control, ICACC'09.*, 2009.

[43] R. T. de Souza, S. D. Zorzo and D. A. da Silva, "Evaluating capstone project through flexible and

collaborative use of Scrum framework," in *IEEE Frontiers in Education Conference (FIE)*, 2015.

[44] E. d. Nuevo, M. Piattini and F. J. Pino, "Scrum-based methodology for distributed software development," in *6th IEEE International Conference on Global Software Engineering (ICGSE)*, 2011.

[45] F. Cannizzo, G. Marcionetti and P. Moser, "Evolution of the tools and practices of a large distributed agile team," in *Agile Conference (AGILE'08)*, 2008.

[46] S. H. Su and C. Scharff, "Know Yourself and Beyond: A students' global software development project experience with Agile Methodology," in *5th International Conference on Computer Science and Education (ICCSE)*, 2010.

[47] M. Penttinen and T. Mikkonen, "Subcontracting for Scrum Teams Experiences and Guidelines from a Large Development Organization," in *IEEE Seventh International Conference on Global Software Engineering (ICGSE)*, 2012.

[48] A. L. Peres, F. Selleri, J. B. Antunes, F. Martins, K. d. S. Brito, R. R. Wanderley, F. S. F. Soares, V. C. Garcia and S. R. de Lemos Meira, "Methods and Processes Definitions for Multiplatform Social Network Games Development with Distributed Teams," in *Brazilian Symposium on Games and Digital Entertainment (SBGAMES)*, 2011.

[49] J. M. Bass, "Scrum Master Activities: Process Tailoring in Large Enterprise Projects," in *IEEE 9th International Conference on Global Software Engineering (ICGSE)*, 2014.

[50] G. Roche and B. Vasquez-McCall, "The Amazing Team Race A Team Based Agile Adoption," in *AGILE'09 Agile Conference*, 2009.

[51] E. Hossain and M. A. P. H.-y. Babar, "Using scrum in global software development: a systematic literature review," in *Fourth IEEE International Conference on Global Software Engineering (ICGSE)*, 2009.

[52] E. Hossain, M. A. Babar, H.-y. Paik and J. Verner, "Risk identification and mitigation processes for using scrum in global software development: A conceptual framework," in *Asia-Pacific Software Engineering Conference (APSEC'09)*, 2009.

[53] G. Rodriguez, A. Soria and M. Campo, "Supporting Virtual Meetings in Distributed Scrum Teams," *Latin America Transactions, IEEE (Revista IEEE America Latina),* vol. 10, no. 6, pp. 2316-2323, 2012.

[54] J. Raszka and L. Jamroz, "Reducing human resources in management of information technology (IT) projects," in *8th International Conference on Human System Interactions (HSI)*, 2015.

[55] G. K. Ghosh, "Challenges in Distributed Scrum," in *IEEE Seventh International Conference on Global Software Engineering*, 2012.

[56] D. Damian, C. Lassenius, M. Paasivaara, A. Borici and A. Schröter, "Teaching a globally distributed project course using Scrum practices," in *Collaborative Teaching of Globally Distributed Software Development Workshop (CTGDSD)*, 2012.

[57] J. Lewis and K. Neher, "Over the Waterfall in a Barrel - MSIT Adventures in Scrum," in *Agile Conference (AGILE'07)*, 2007.

[58] E. Collins, G. Macedo, N. Maia and A. Dias-Neto, "An Industrial Experience on the Application of Distributed Testing in an Agile Software Development Environment," in *IEEE Seventh International Conference on Global Software Engineering (ICGSE)*, 2012.

[59] S. Jalali and C. Wohlin, "Agile practices in global software engineering-A systematic map," in *5th IEEE International Conference on Global Software Engineering (ICGSE)*, 2010.

[60] D. F. Rico and H. H. Sayani, "Use of Agile Methods in Software Engineering Education," in *Agile Conference (AGILE'09)*, 2009.

[61] M.-W. Chung, S. Nugroho and J. F. J. Unson, "Tidal Wave: The Games Transformation," in *Conference Agile (AGILE'08)*, 2008.

[62] F. Maurer and S. Martel, "Extreme programming. Rapid development for Web-based applications," *IEEE Internet computing,* vol. 6, no. 1, pp. 86-90, 2002.

[63] M. Md Rejab, J. Noble and G. Allan, "Distributing Expertise in Agile Software Development Projects," in *Agile Conference (AGILE'14)*, 2014.

[64] B. Sheth, "Scrum 911! using scrum to overhaul a support organization," in *Agile Conference (AGILE'09)*, 2009.

[65] G. Tabunshchik and S. Malyuk, "Technique of operating time enumeration in the agile-projects," in *International Conference on "Modern Problems of Radio Engineering, Telecommunications and Computer Science"(TCSET)*, 2008.

[66] M. A. Ashraf, S. Shamail and Z. Rana, "Agile model adaptation for e-learning students' final-year project," in *IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, 2012.

[67] E. Sink, "Stories from My Experiences Learning Scrum," in *Agile Conference (AGILE'11)*, 2011.

[68] F. Cannizzo, G. Marcionetti and P. Mose, "The Toolbox of a Successful Software Craftsman," in *15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS)*, 2008.

[69] M. Esbensen, P. Tell, J. B. Cholewa, M. K. Pedersen and J. Bardram, "The dBoard: A Digital Scrum Board for Distributed Software Development," in *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces (ITS '15)*, 2015.

[70] N. Nikitina and M. Kajko-Mattsson, "Consequences of business growth on software processes," in *Proceedings of the 11th International Conference on Product Focused Software (PROFES '10)*, 2010.

[71] E. Toews, B. Satchwill, R. Rankin, J. Shillington and T. King, "An Internationally Distributed Cloud for Science: The Cloud-enabled Space Weather Platform," in *Proceedings of the 2nd International Workshop on Software Engineering for Cloud Computing (SECLOUD '11)*, 2011.

[72] C. Scharff, S. Heng and V. Kulkarni, "On the difficulties for students to adhere to scrum on global software development projects: preliminary results," in *Proceedings of the Second International Workshop on Collaborative Teaching of Globally Distributed Software Development (CTGDSD '12)*, 2012.

[73] I. Bosnić, F. Ciccozzi, I. a. D. N. E. a. F. J. Čavrak and R. Mirandola, "Introducing SCRUM into a Distributed Software Development Course," in *Proceedings of the 2015 European Conference on Software Architecture Workshops (ECSAW '15)*, 2015.

[74] M. Paasivaara, C. Lassenius, D. Damian, P. Räty and A. Schröter, "Teaching students global software engineering skills using distributed scrum," in *Proceedings of the 2013 International Conference on Software Engineering (ICSE '13)*, 2013.

[75] M. Bourimi and R. Tesoriero, "Non-Functional Requirements for Distributable User Interfaces in Agile Processes," in *Proceedings of the 2014 Workshop on Distributed User Interfaces and Multimodal Interaction (DUI '14)*, 2014.

[76] C. Scharff, "An evolving collaborative model of working in students' global software development projects," in *Proceedings of the 2011 Community Building Workshop on Collaborative Teaching of Globally Distributed Software Development (CTGDSD '11)*, 2011.

[77] A. Tuli, N. Hasteer, M. Sharma and A. Bansal, "Empirical investigation of agile software development: cloud perspective," *SIGSOFT Softw. Eng. Notes,* vol. 39, no. 4, pp. 1-6, 2014.

[78] J. Coldewey, J. Link and K. Marquardt, "Agility unlimited?," in *Companion to the 22Nd ACM SIGPLAN Conference on Object-oriented Programming Systems and Applications Companion (OOPSLA '07)*, 2007.

[79] E. Brechner, "Journey of Enlightenment: The Evolution of Development at Microsoft," in *Proceedings of the 27th International Conference on Software Engineering (ICSE '05)*, 2005.

[80] B. Bruegge, S. Krusche and L. Alperowitz, "Software Engineering Project Courses with Industrial Clients," *Trans. Comput. Educ.,* vol. 15, no. 4, pp. 17:1-17:31, 2015.

[81] J. Rubart, "Extending shared hypermedia workspaces to face-to-face multitouch experiences," *SIGWEB Newsl.,* vol. Summer, no. Summer 2014, pp. 2:1--2:5, 2014.

[82] M. Paasivaara, K. Blincoe, C. Lassenius, D. Damian, J. Sheoran, F. Harrison, P. Chhabra, A. Yussuf and V. Isotalo, "Learning global agile software engineering using same-site and cross-site teams," in *Proceedings of the 37th International Conference on Software Engineering - Volume 2 (ICSE '15)*, 2015.

[83] P. Gestwicki, "The Entity System Architecture and Its Application in an Undergraduate Game Development Studio," in *Proceedings of the International Conference on the Foundations of Digital Games (FDG '12)*, 2012.

[84] M. Esbensen and P. Bjørn, "Routine and Standardization in Global Software Development," in *Proceedings of the 18th International Conference on Supporting Group Work (GROUP '14)*, 2014.

[85] S. Goeschl, M. Herp and C. Wais, "When agile meets OO testing: a case study," in *Proceedings of*

the 1st Workshop on Testing Object-Oriented Systems (ETOOS '10), 2010.

[86] J. Vlietland, R. van Solingen and H. van Vliet, "Aligning codependent Scrum teams to enable fast business value delivery: A governance framework and set of intervention actions," *Journal of Systems and Software,* 2015.

[87] E. Scott, G. Rodríguez, Á. Soria and M. Campo, "Are learning styles useful indicators to discover how students use Scrum for the first time?," *Computers in Human Behavior,* vol. 36, pp. 56-64, 2014.

[88] M. E. Grimheden, "Can agile methods enhance mechatronics design education?," *Mechatronics,* vol. 23, no. 8, pp. 967-973, 2013.

[89] A. Martini, J. Bosch and M. Chaudron, "Investigating Architectural Technical Debt accumulation and refactoring over time: A multiple-case study," *Information and Software Technology,* vol. 67, pp. 237-253, 2015.

[90] M. Kajko-Mattsson, G. Azizyan and M. K. Magarian, "Classes of distributed Agile development problems," in *Agile Conference (AGILE'10)*, 2010.

[91] S. Jalali and C. Wohlin, "Systematic literature studies: database searches vs. backward snowballing," in *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement*, 2012.

List of tables

Appendix A – Problem groups

Table 1: Incorrect execution of Scrum

#	Problem	Ref
1	It is unclear who is responsible for updating backlogs	[5]
2	All sites need to understand Scrum	[5]
3	Requirements engineering when "throwing the requirements over the fence" does not allow developers and tester to participate	[10]
4	No up-to-date test cases	[12]
5	Multiple Product Owners – Lack of responsibility with team because of messy road map	[13]
6	Difficult leveraging available resources, best practices are often deemed proprietary, time consuming and difficult to maintain	[17]
7	The team that gets to pick from the backlog first gets all the interesting stories	[19]
8	Inadequate team structure	[20]

Table 2: Not communicating all information to team

#	Problem	Ref
1	Long teleconferences are exhausting	[5]
2	Volatile specification	[12]
3	Lack of formal document for requirements	[20]
4	Difficult to fully communicate client nuances, context and priorities	[22]
5	Misunderstanding between the Product Owner and development team	[25]
6	Requirements written for solution domain which is not understood	[27]
7	Documented requirements only for the “tip of the ice berg”	[27]

Table 3: Misunderstanding

#	Problem	Ref
1	It takes time to learn to report information that is useful for others	[5]
2	Difficult to recognize speakers when their faces are not visible	[5]
3	Misunderstanding requirements - This is undetected until the demo	[8]
4	Reduced cooperation arising from misunderstandings	[18]
5	Cultural differences	[22]
6	Difference in communication resulting in miscommunication	[25]

Table 4: No syncing between sites

#	Problem	Ref
1	Working in silos - rarely any knowledge sharing	[9]
2	Synchronize work between sites	[17]
3	Keeping track of progress	[19]
4	Resistance to communicating long term plans to development teams	[21]
5	Coordinating the continuous changes and refinements of the project	[28]

Table 5: Planning a meeting with everyone present is difficult

#	Problem	Ref
1	Time zone differences make it challenging to arrange meetings, especially long ones	[5]
2	Team collaboration when working in different time zones - not everyone involved in status meetings	[9]
3	Meetings with everyone present with different time zones	[19]
4	Hard to plan demos with all stakeholders present	[23]
5	Planning with really big teams	[26]

Table 6: Hardware and tools not sufficient

#	Problem	Ref
1	Sound quality not always good enough	[5]
2	Wiki editor is awkward to use	[5]
3	No possibility for videoconference - The network connection between offices is not fast enough for videoconferencing	[8]
4	Losing communication bandwidth (tone and emotion) due to the usage of video conferencing and phone calls	[19]
5	Having the right tools for communication and collaboration	[22]

Table 7: Difference in reporting impediments

#	Problem	Ref
1	Cultural differences e.g. about reporting impediments	[5]
2	Differences in interests during retrospective	[13]
3	Problems are not shared; one location only shares the positive things	[19]
4	Difference in Openness of society	[25]

Table 8: Product Owner not present

#	Problem	Ref
1	Proxy Product Owner is not always present leading to delays and missing information	[14]
2	Due to distance between Product Owner and the team missing information was delaying sprint goals	[14]
3	Product Owner was project manager, therefore stayed with the client	[14]
4	Waiting long time for the Product Owner to reply for a simple question	[19]

Table 9: Integration difficulties

#	Problem	Ref
1	Legal barrier between the onshore and offshore location, a person from outside the EU is not allowed to access financial systems of a company within the EU	[10]
2	Process adaption in DSD is slower and more difficult than in regular collocated Scrum	[12]
3	Hard to ensure that the product satisfies the market needs and has the required documentation and has the best quality	[26]
4	Increasing time needed for integrating code	[27]

Table 10: Lack of focus

#	Problem	Ref
1	Low software quality and lack of focus in sprint	[12]
2	Multiple Product Owners – Teams miss intrinsic motivation because of a messy road map	[13]
3	Communication barriers resulting in a lack of team commitment	[20]

Table 11: Scrum of Scrums not effectively used

#	Problem	Ref
1	In the Scrum of Scrums problems are not reported because the reporter assumes they will receive no help anyway	[11]
2	In the Scrum of Scrums problems are not reported because the reporter assumes it is not relevant	[11]
3	Scrum of Scrum meetings are not always useful because a lot of information is not relevant for everyone	[11]

Table 12: Coordinating in multiple time zones is difficult

#	Problem	Ref
1	Having people in three time zones is difficult bordering on insane	[16]
2	Increased coordination costs (due to temporal distance)	[18]
3	Coordinating between 4 time zones	[26]

Table 13: Different holidays

#	Problem	Ref
1	Different religious or other holidays in different countries cause synchronization challenges	[5]
2	Holidays where not overlapping	[14]

Table 14: Silence / passivism

#	Problem	Ref
1	Cultural and language differences may cause some participants to be more silent	[5]
2	Silence caused by distance - Norwegian team asked many questions, Malaysian team preferred to listen	[8]

Table 15: Different work practices

#	Problem	Ref
1	Different approaches with sprint planning	[13]
2	Inconsistent work practices can impinge on effective coordination	[18]

Table 16:- Multiple Product Owner not in sync

#	Problem	Ref
1	Collaboration and communication between Area Product Owners is not good	[7]
2	Multiple Product Owners – Product Owners not really in sync gives a messy road map	[13]

Table 17: Informal contact is lost

#	Problem	Ref
1	Reduced informal contact can lead to lack of critical task awareness	[18]
2	Information sharing in the hallway is lost	[19]

Table 18: Meetings at the office outside office hours

#	Problem	Ref
1	Reduced informal contact can lead to lack of critical task awareness	[19]
2	Having meetings outside office hours	[19]

Table 19: No transparency between sites

#	Problem	Ref
1	Silence caused by distance - Norwegian team felt unsure whether the off-site team had really understood everything	[8]
2	Transparency for both sites not accomplished	[12]

Table 20: Features not being deployment ready at end of sprint

#	Problem	Ref
1	Test-ready features at the end of the sprint instead of deployment ready ones	[12]
2	Succeeding in sprints is difficult	[19]

Table 21: Managing customers new to agile

#	Problem	Ref
1	Management not used to agile practices	[20]
2	Managing customers new to agile	[22]

Table 22: Time differences

#	Problem	Ref
1	Difference in dinnertime caused one location to plan meetings for the other location during dinner time without knowing so	[19]
1	Having to work late at night makes one location feel as being second hand	[19]

Appendix B – Problems without group

Table 1: Problems without group

#	Problems	Ref
1	With RUP only certain people communicated with each other using video-conference	[10]
2	User stories where in German and had to be explained in English	[14]
3	Shared designing hard without overlapping hours	[23]
4	No budget for traveling	[23]
5	Difference in willingness to adopt new techniques and technology	[25]

Appendix C – Solution groups

Table 1: Apply Scrum

#	Solution	Ref
1	User stories provide help for communication	[6]
2	The Product Owner role provides help for communication	[6]
3	The Scrum Master role provides help for communication	[6]
4	Product Owner asks follow up questions	[8]
5	Monthly peer feedback from each peer in the Scrum team	[9]
6	Using Scrum all team members have face-to-face meetings which builds trust	[10]
7	Involve customer more in prioritization	[12]
8	Small manageable stories	[12]
9	No story updates during sprint	[12]
10	Use retrospective as a driver for continuous process improvement	[12]
11	User stories need to be in more detail	[14]
12	Enable reflections to review the process	[18]
13	Enable reflections to review the process	[18]
14	Enable planning activities to scope work and schedule it	[18]
15	Enable planning activities to scope work and schedule it	[18]
16	Enable planning activities to scope work and schedule it	[18]
17	Time box those (local) meetings	[19]
18	Have a 3rd party present at retrospectives	[19]
19	Use cases should be integrated with user stories	[20]
20	Assigning testers and developers to work together	[20]
21	Informal updates by the Product Owner	[22]
22	Continuously improving development practices	[27]
23	Requirements management and agile planning	[27]
24	Establishing technical and business feedback rhythms	[27]

Table 2: Travel and meet face to face

#	Solution	Ref
1	Frequent face-to-face visits	[5]
2	Frequent visits	[8]
3	Annual gathering	[8]
4	Visiting engineer during first iteration	[8]
5	Travel to strengthen communication and trust	[9]
6	Increase contact visits	[12]
7	Contact visits to improve trust and team spirit	[12]
8	Team building - Visit the other location	[14]
9	Use nearshore resources – have people in the same time zone that can meet face-to-face regularly	[16]
10	Visit the other site to enable face-to-face meetings	[18]
11	Quarterly meetings with everybody present	[19]
12	Successful Product Owners communicate bilateral	[21]
13	Travel to develop personal relationships	[22]
14	Regular traveling	[22]
15	Start with a collocated period	[23]
16	Organize face-to-face quarterly meetings	[24]
17	Cross culture understanding with more collaboration and onsite visits	[27]

Table 3: Different communication channels

#	Solution	Ref
1	Have many different possibilities to communicate between sites	[5]
2	Have multiple lines of communication	[16]
3	Constant availability of team members in instant messaging	[12]
4	Use of tools that enable synchronous communication between teams (e.g. conference calls and video conference)	[18]
5	Use of tools that enable asynchronous communication between teams (e.g. email and instant messaging)	[18]
6	Use of tools that enable synchronous communication between teams (e.g. conference calls and video conference)	[18]
7	Use of tools that enable asynchronous communication between teams (e.g. email and instant messaging)	[18]
8	Use of tools that enable synchronous communication between teams (e.g. conference calls and video conference)	[18]
9	Use of tools that enable asynchronous communication between teams (e.g. email and instant messaging)	[18]
10	Use of tools that enable synchronous communication between teams (e.g. conference calls and video conference)	[18]
11	Use of tools that enable asynchronous communication between teams (e.g. email and instant messaging)	[18]
12	Always-open Skype connections	[22]
13	Shared wiki updated at end of day	[23]
14	Use video conferencing over email and other forms of communication	[24]

Table 4: Non-Scrum practices

#	Solution	Ref
1	10-15% of sprint time is reserved for sharing knowledge and learning new skills	[9]
2	All relevant informal communication needs to be appended to the feature ticket in the issue tracking tool	[12]
3	Give test site the right to reject stories	[12]
4	Review of test cases for critical user stories by a developer	[12]
5	Incremental inclusion of test site	[12]
6	Sunshine cases should work when a story is passed to the tester so the tester can focus on corner cases	[12]
7	Write the answers to the daily Scrum meeting down and email this before the daily Scrum	[17]
8	Resolve questions regarding stories by mail before the daily Scrum	[17]
9	Be aware of the loss of informal information sharing in the hallway and compensate with other practices	[19]
10	While piloting agile practices the team must document Scrum meeting minutes	[20]
11	Add a project manager to manage the client and deliver a clear backlog	[22]
12	Scrum of Scrums not used	[29]
13	Sprint review not done	[29]
14	Sprint review done with only Scrum Masters, project master and customer present	[29]

Table 5: Proxy Product Owners

#	Solution	Ref
1	Area Product Owners	[7]
2	On-site system expert	[8]
3	Proxy Product Owner was used to have a Product Owner on site	[14]
4	Proxy Product Owner on site with the client	[14]
5	The chief Product Owner meets regularly with all Product Owners	[17]
6	Have someone who regularly syncs with the Product Owner who can answer those questions for that location (a proxy PO)	[19]
7	Use a Proxy Product Owner	[21]
8	Assign a Proxy Product Owner with in debt understanding of both cultures	[25]
9	Multiple Product Owners	[26]
10	Move the Product Owner to the same room as the developers and make part of the team	[27]
11	Move the Product Owner to the same room as the developers and make part of the team	[27]
12	Move the Product Owner to the same room as the developers and make part of the team	[27]

Table 6: Use the daily Scrum

#	Solution	Ref
1	Daily Scrum creates contacts and encourages informal communication, especially between the sites	[5]
2	The daily Scrum provides coordination	[6]
3	Dedicated daily collaboration of 45 min	[9]
4	Using Scrum, the daily interactions enable the team to find misunderstandings early in the process	[10]
5	Daily standup meeting with video conference technologies	[15]
6	Standup done with whole team present at start of the Dutch work day	[22]
7	Daily standup of Indian team with proxy member of US team	[23]
8	Daily Scrum distributed via Skype	[29]
9	Daily Scrum done locally	[29]
10	Two daily Scrums, one locally one distributed	[29]
11	Daily Scrum done locally	[29]

Table 7: Shared backlog and resources

#	Solution	Ref
1	All team members can access, pick items and follow progress with shared backlogs	[5]
2	Shared electronic workspaces (e.g. Sharepoint)	[16]
3	Chief Product Owner and Chief Scrum Master to manage Scrum of Scrums and backlog management centrally	[17]
4	Share pictures of the physical Scrum boards on a wiki	[19]
5	Use a "global" task board	[20]
6	Electronic shared backlogs and a wiki to share information and documentation	[22]
7	Have one big backlog	[26]
8	Backlog shared between locations	[29]
9	Backlog shared between locations	[29]
10	Backlog shared between locations	[29]
11	Backlog shared between locations	[29]

Table 8: Weekly Scrum of Scrums

#	Solution	Ref
1	Weekly Scrum of Scrums distributes information between the teams	[5]
2	Weekly Scrum of Scrums opens discussion channels and encourages informal communication between teams	[5]
3	Weekly Scrum of Scrums	[8]
4	Weekly 30 min Scrum of Scrums	[9]
5	Bi-Weekly 60 min Scrum of Scrums	[9]
6	Scrum masters of all sites meet for a Scrum of Scrums each Monday	[17]
7	Architects are allocated to the Scrum teams and also meet each Monday	[17]
8	Scrum of Scrums distributed done weekly	[29]
9	Scrum of Scrums done weekly combined with sprint planning and review	[29]
10	Scrum of Scrums distributed done weekly	[29]

Table 9: Frequent communication

#	Solution	Ref
1	Frequent communication with Scrum practices bring transparency to a distributed project	[5]
2	Frequent communication with Scrum practices reveals problems early on	[5]
3	Sprints provide frequent monitoring opportunities between the sites	[5]
4	The unified Scrum team ensured only one party was communicating to the team in India	[10]
5	Code Buddy as a bi-directional proxy to facilitate inter-team communication (1 different team member stay's in the German office for one week)	[13]
6	Enable frequent communication	[18]
7	Enable frequent communication	[18]
8	Enable frequent communication	[18]
9	have the senior members update one another with a call from time to time	[19]
10	A project news gazette after every iteration	[22]

Table 10: Sprint planning with all sites present

#	Solution	Ref
1	Sprint planning meetings give a possibility for team members from all sites to participate, to ask for clarification, to understand tasks and to commit to shared goals	[5]
2	Common sprint planning with representatives	[7]
3	Distributed sprint planning meetings	[8]
4	Using Scrum all team members participate in requirements engineering	[10]
5	Sprint planning done with whole team present	[22]
6	Sprint planning done with whole team	[23]
7	Global structure, working with clearly distributed daily Scrum and planning to manage the work focus and boundaries	[27]
8	Sprint planning modified with pre- and post-planning meetings with onshore team	[29]
9	Sprint planning modified with additional pre-planning meeting	[29]
10	Sprint planning modified with additional pre-planning meeting	[29]

Table 11: Non-Scrum meetings

#	Solution	Ref
1	Requirement and design workshop before sprint planning	[7]
2	Unofficial distributed meetings	[8]
3	Feature Scrum of Scrum meetings for the teams working on the same feature	[11]
4	Increased up-front planning and specification to identify problems, corner cases and impact on existing software early and improve estimation of team	[12]
5	On-demand specification meetings with members on both sites	[12]
6	Meetings with the customer before sprint	[12]
7	The War Room – before a story is given to the Product Owner the development team tries to “break” the feature, and thus discover failures early	[13]
8	Daily sync meeting next to daily Scrum for technical details	[13]
9	Standup once a week done with whole team	[23]

Table 12: Meetings with local team

#	Solution	Ref
1	Hold retrospectives asynchronous and post result in the shared electronic workspace	[16]
2	Do the meetings locally	[19]
3	Have retrospectives with local teams	[19]
4	Demo done with customer and Dutch members, Indian members briefed after	[22]
5	Retrospective done locally every 2 weeks	[29]
6	Retrospective done locally every 4 weeks	[29]
7	Sprint review done with onshore team	[29]

Table 13: Iterate / use sprints

#	Solution	Ref
1	Deliverables at end of sprint build relations and trust	[6]
2	Synchronized 4-week sprints	[8]
3	Build all aspects of a story including non-functional requirements (e.g. stability and performance)	[12]
4	Continuous deployment to customer every sprint	[12]
5	Use sprints to empower and motivate the development team	[15]
6	Iterate to monitor progress and resolve issues	[18]
7	Iterate to monitor progress and resolve issues	[18]

Table 14: Successful team composition

#	Solution	Ref
1	Get the right people and get them talking	[16]
2	Have teams in two time zones and split work accordingly	[16]
3	Scrum master needs to be a strong negotiator	[20]
4	Communicate long term vision to maintain team commitment	[21]
5	Team culture aimed at openness and direct communication	[22]
6	Have a local collocated team to clear roadblocks that can only be done by a local team	[22]
7	Favor small teams of 4 to 6 over larger teams	[24]

Table 15: Sprint demo with all sites present

#	Solution	Ref
1	Sprint demo ensures the understanding of the requirements, especially regarding the offsite	[5]
2	Sprint demo prevents problems by providing a frequent monitoring opportunity between the sites	[5]
3	Common sprint demo	[7]
4	Distributed sprint demo's	[8]
5	Demo done with whole team	[23]
6	Sprint review done distributed	[29]

Table 16: Increase overlapping hours

#	Solution	Ref
1	Increase overlapping work hours between sites to enable synchronous communication	[18]
2	Increase overlapping work hours between sites to enable synchronous communication	[18]
3	Increase overlapping work hours between sites to enable synchronous communication	[18]
4	Increase overlapping work hours between sites to enable synchronous communication	[18]
5	Alternate the schedule so everyone has to work late at night at some point	[19]

Table 17: Retrospective with all sites present

#	Solution	Ref
1	Common retrospective	[7]
2	Distributed retrospective meetings	[8]
3	Retrospective done with whole team present	[22]
4	Retrospective done with whole team	[23]

Table 18: Daily Scrum of Scrums

#	Solution	Ref
1	Daily Scrum of Scrums builds relations and trust	[6]
2	Daily 20 min Scrum of Scrums	[9]
3	Daily Scrum of Scrums with Scrum Masters after standup	[22]
4	Have a daily Scrum of Scrums meeting with another higher level Scrum of Scrums meeting above it	[26]

Table 19: Sprint planning prepared in separate meeting

#	Solution	Ref
1	Sprint planning prepared separately	[23]
2	Sprint planning modified with pre- and post-planning meetings with onshore team	[29]
3	Sprint planning modified with additional pre-planning meeting	[29]
4	Sprint planning modified with additional pre-planning meeting	[29]

Table 20: Give all sites Scrum training

#	Solution	Ref
1	Give all sites a proper Scrum training	[5]
2	Specialized coaching at each location	[21]
3	Training the teams in knowing the problem domain	[27]

Table 21: Scrum Board and burn down charts

#	Solution	Ref
1	The Scrum Board provides coordination	[6]
2	Visible Scrum Boards provide help for communication	[6]
3	Burn down chart builds relations and trust	[6]

Table 22: Use universal language to increase clarity

#	Solution	Ref
1	Explain user stories face-to-face in English	[14]
2	Explain the German user stories to the German speaking site manager via screen sharing in telephone conferences	[14]
3	Translate user stories in English	[14]

Table 23: Communication strategy

#	Solution	Ref
1	Have a strict communication plan	[16]
2	Communication protocols	[27]
3	Use well defined mechanisms for collaboration	[28]

Table 24: Separate backlogs for each team

#	Solution	Ref
1	Separate backlogs for each team	[8]
2	Put stories on the backlog for specific teams based on local business value	[19]

Table 25: Retrospective not done

#	Solution	Ref
1	Retrospective not done	[29]
2	Retrospective not done	[29]

Appendix D – Solutions without group

Table 1: Solutions without group

#	Solutions	Ref
1	Bug fix iteration (should be avoided)	[12]
2	No informal story updates	[12]
3	Staffing: new Business Analyst	[12]
4	Talk about cultural differences	[19]
5	Do not follow Scrum practices religiously	[19]
6	Company culture of openness with an equal value	[22]
7	Do not be afraid to make mistakes, but be ready to implement changes when you see something is not working	[24]
8	Create a validation team that defines customer configurations	[26]
9	One global technical infrastructure	[27]
10	Sprint planning modified, held weekly with project manager and sub-team coordinators	[29]

TUD-SERG-2016-008
ISSN 1872-5392