

Empirically-based Solution Framework for Distributed Engineering of Software Intensive Products

Dissertation

zur Erlangung des Doktorgrades Dr. rer. pol.

der Fakultät für Mathematik und Wirtschaftswissenschaften

der Universität Ulm

vorgelegt von

Stefan Schneider

Gutachter:

Prof. Dr. Franz Schweiggert

Prof. Dr. Leo Brecht

Prof. Dr. Tony Gorschek

Amtierender Dekan:

Prof. Dr. Werner Smolny

Ulm, 27.5.2015

to my grandfather Bruno, for being my first friend

to my mum, always caring about everyone else

Acknowledgements

At the very outset, I want to thank Prof. Franz Schweiggert for beliving in me and supporting my plan to start and finish this work. I got to know Prof. Schweiggert as an encouraging and challenging mentor and teacher. In fact, I met Prof. Schweiggert at an open day at Ulm University. I experienced his patience and support even before I started to study. Hence, I want to thank him for his support, interest and guidance that made this time so productive and exciting.

Also, I want to thank Prof. Tony Gorschek. I first met him during a conference and he invited me to do a research visit at BTH. During this visit we decided to work together on this particular research project. In the next three years I spent several weeks at BTH. Thanks to his constant support and infinite patience I learned the fundamentals of research. His guidance has made this time an enlightening, challenging and intense learning experience and I want to thank him for the time he spent on advising me.

I also want to thank Prof. Brecht for his valuable input and work to finish this PhD work.

I also want to thank the Daimler AG for offering an interesting topic and providing practical insights into real life projects and providing calibration of my work towards industry reality. Especially, I want to thank Thomas Gantner and Michael Stupperich for their support and input. Also, I want to thank Katrin Dietrich, Heike Fenster and Jörg Leuser for constantly encouraging me and never becoming tired of discussing with me. This has been extremely valuable for me and I want to thank them a lot for their time and patience.

Furthermore, I want to thank my family, especially my mum and dad and my brothers who have constantly supported me and my work. They always had an open ear for me and offered support as only family can, through good but especially through bad times. I always have and always will ask for their priceless advice.

Finally, I want to thank my friends Martin, Benedikt, Christian, Daniel, Wiebke and Stefan, Vera, Berit, Sabrina, Frederick and David. I greatly appreciate all the time you were there for me, talking and discussing with me but also occasionally distracting me.

Deutsche Zusammenfassung

In dieser Arbeit wurde ein Prozessframework für die verteilte Entwicklung softwarebasierter Systeme erarbeitet. Dabei wurden zunächst die Probleme im Bereich Global Software Engineering untersucht, darauf aufbauend ein Framework für Lösungen dieser Probleme erarbeitet und befüllt sowie in Theorie und Praxis evaluiert. Die folgenden Ergebnisse wurden erarbeitet:

i. Untersuchung der Probleme in der verteilten Softwareentwicklung sowie des Magels an leicht zu findenden Lösungen.

ii. Aufstellen der vier Dimensionen der Verteiltheit sowie deren Nutzung zur Strukturierung der Probleme in der verteilten Entwicklung.

iii. Durchführung eines systematischen Literaturreviews zur Identifikation von Lösungen.

iv. Bewertung der identifizierten Lösungen.

v. Erstellung eines auf CMMI und ISO basierenden Lösungsframeworks für verteiltes Entwickeln zur Strukturierung der identifizierten Lösungen.

vi. Anpassen des Frameworks um verteiltes Entwickeln besser darstellen zu können.

vii. Einsortieren der Lösungen in das Framework zur besseren Benutzbarkeit.

viii. Anbindung der Lösungen an das Framework durch eine zusätzliche Verbindungsschicht.

ix. Identifikation von leeren Prozessgebieten innerhalb des Frameworks für zukünftige Forschung.

x. Lösung eines Problems bei der Daimler AG in einem dieser leeren Prozessgebiete durch die Erstellung einer Methode zur Variantenreduktion und dadurch exemplarisches Füllen eines leeren Prozessgebietes innerhalb des Frameworks.

Teile dieser Dissertation wurden bereits in folgenden Fachartikeln veröffentlicht:

Schneider S., Gorschek T., Torkar R., “Solutions in Global Software Engineering: A Systematic Literature Review”, International Journal of Information Management, volume 33, issue 1, 2013, pp. 119 - 132.

Schneider S., Gorschek T., “Software Variant Reduction and Control through Factor Analysis”, submitted to the International Journal of Software Engineering and Knowledge Engineering.

Contributions

My contributions during my PhD work include

i. Studying the major problems in Global Software Engineering (GSE) and identifying the lack of easy-to-find solutions.

ii. Creating and using four dimensions of distribution to break down GSE problems.

iii. Performing a systematic literature review to identify solutions in GSE.

iv. Rate the papers found in the systematic literature review.

v. Creating a Solution Framework for Distributed Engineering (SODE) based on CMMI and ISO to structure the solutions found in the systematic literature review.

vi. Adding a process area to SODE to cover GSE related aspects better than CMMI and ISO do.

vii. Sort the papers into SODE to guide the users towards the solutions they are looking for.

viii. Adding a layer between the papers and SODE to support the users better in identifying the right solutions by linking the papers to SODE.

ix. Identifying white spots within SODE as possible areas to do future work.

x. Using a challenge at Daimler AG to fill one white spot by creating a method for software variant reduction and control based on factor analysis.

Contents

Acknowledgements **5**

Deutsche Zusammenfassung **7**

Teile dieser Dissertation wurden bereits in folgenden Fachartikeln veröffentlicht: **9**

Contributions **10**

Contents **11**

List of Figures **15**

List of Tables **17**

1 Introduction **19**

1.1 Background and Related Work 20

1.1.1 Challenges in GSE 22

1.1.2 Maturity Models and Best Practice Frameworks 27

1.1.3 Research Context 28

1.2 Thesis Contribution 29

1.2.1 Research Questions 29

1.2.2 Research Methodology 31

1.2.3 Chapter Summary and Contribution 36

1.3 List of Publications 38

1.3.1 Publications included in this thesis 38

1.3.2 Publications not included in this thesis 39

2 Solution framework for Distributed Engineering (SODE) **41**

2.1 Introduction 41

2.2 Background and Related Work 41

2.2.1 Background 42

2.2.2 Frameworks and Models 42

2.3 Creation of SODE 43

2.3.1 STEP 1: Adaption of CMMI 43

2.3.2 STEP 2: Substitution based on ISO, ACM and PMBOK 44

2.3.3 STEP 3: Revision based on a systematic literature review 44

2.3.4 STEP 4: Addition of a solution layer 44

2.4 SODE 0.2 45

2.4.1 PA 1 Process Management 46

2.4.2 PA 2 Project Management 47
2.4.3 PA 3 Engineering 48
2.4.4 PA 4 Support 50
2.4.5 Conclusion 51

3 Solutions in Global Software Engineering: A Systematic Literature Review 53

3.1 Introduction 53
3.2 Related Work and Background 54
3.2.1 Related Work 54
3.2.2 Study Context 55
3.3 Study Design 55
3.3.1 Research Questions 56
3.3.2 Systematic Literature Review (SLR) 57
3.4 Results & Quantitative Analysis 66
3.4.1 RQ1: Process Areas in SODE 69
3.4.2 RQ2: Map solutions against SODE 69
3.4.3 RQ2.2: New PAs 72
3.4.4 RQ3: Maturity of the Solutions 74
3.5 Qualitative Analysis with Solution Characterization 78
3.5.1 PA 1 Process Management 80
3.5.2 PA 2 Project Management 80
3.5.3 PA 3 Engineering 82
3.5.4 PA 4 Support 83
3.6 Conclusions 83

4 Evaluation 85

4.1 Introduction 85
4.2 Evaluation of SODE 85
4.2.1 STEP5: Improvement through specific expert interviews 85
4.2.2 STEP 6: Improvement through diversified validation 86
4.3 Evaluation Design 86
4.3.1 Expert interviews 86
4.3.2 Workshops 89
4.4 Validation – Results and Analysis 91
4.4.1 RQ1: Is SODE complete? 92
4.4.2 RQ2: Is SODE usable? 94
4.4.3 RQ3: Is SODE useful? 98
4.5 Conclusion 102

5 Software Variant Reduction and Control through Factor Analysis 103

5.1 Introduction 103
5.2 Background and Related Work 104
5.2.1 Daimler AG 104
5.2.2 Related work 105
5.3 Factor Analysis Method (FAM) 106
5.3.1 FAM – Step 0: Prerequisites and data collection 108
5.3.2 FAM – Step 1: Calculate VCV matrix 110
5.3.3 FAM – Step 2: Calculate eigenvalues and eigenvectors 112
5.3.4 FAM – Step 3: Sort eigenvalues and eigenvectors 113
5.3.5 FAM – Step 4: Calculate factor loadings matrix 114
5.3.6 FAM – Step 5: How many factors to keep 117
5.3.7 FAM – Step 6: Analysis of the factors 119
5.3.8 FAM – Step 7: Continuous monitoring 122
5.4 Conclusions 125
5.5 Future Work 126
5.6 Connection to SODE 126

6 Conclusions 127
6.1 Results 127
6.2 Future work 128

Appendix A: Questionnaire A 131

Appendix B: Questionnaire B 137

Appendix C: SODE 1.0 143

Appendix D: SLR Ratings 189

Appendix E: List of SLR Papers 195

Bibliography 231

List of Figures

Figure 1: number of solutions published per year 20
Figure 2: Research Method 32
Figure 3: Creation Steps of SODE 45
Figure 9: Generic structure of SODE with example 46
Figure 4: SODE 0.2 top level structure 46
Figure 5: SODE: Process Management 47
Figure 6: Project Management 48
Figure 7: Engineering 49
Figure 8: Support 51
Figure 10: systematic review process 58
Figure 11: Histogram of all PAs 70
Figure 12: Overview over the PAs 70
Figure 13: PA 1 Process Management 71
Figure 14: PA 3 Engineering 71
Figure 16: PA 4 Support 72
Figure 17: PA 2 Project Management 72
Figure 17: PAM with added PA 73
Figure 18: Ranks of the new PAs as discussed in 3.4.3 75
Figure 19: Ratings of sample PAs 76
Figure 20: Distribution of the ratings 77
Figure 21: Expert Interviews 87
Figure 22: Workshops 90
Figure 23: Coverage of SODE 93
Figure 24: Usability of SODE with (dark bars) and without (bright bars) training 95
Figure 25: Importance of previous knowledge 95
Figure 26: Is the structure of SODE understandable (bright) and adequate (dark)? 96
Figure 27: Overview with (dark) and without (bright) training 97
Figure 28: Problems (bright) vs. Intuitiveness (dark) of navigating through SODE 98
Figure 29: Decision support provided by SODE 99
Figure 30: Is the coverage adequate? 99
Figure 31: Quality of the solutions offered by SODE 100
Figure 32: Time to identify a solution 101
Figure 33: The convex bend in the Scree Plot 119
Figure 34: Location of NewVariants 119
Figure 35: Location of a third NewVariant with respect to the first two NewVariants 120

Figure 36: Different scenario of the location of the third NewVariant 121
Figure 37: Sort FAM into SODE 126

List of Tables

Table 1: Problems in GSE 26
Table 2: Research Questions 31
Table 3: Overview over Chapter 2 37
Table 4: Overview over Chapter 3 37
Table 5: Overview over Chapter 5 38
Table 6: Overview over Chapter 4 38
Table 7: Research Questions for the SLR 57
Table 8: Search strings 60
Table 9: Data Extraction Form 61
Table 10: Scale for Validation ratings 62
Table 11: Scale for Context ratings 62
Table 12: Scale for Design ratings 63
Table 13: Original ratings 64
Table 14: ratings and ranks 64
Table 15: Ranks instead of ratings 65
Table 16: Number of primary studies 67
Table 17: Top ranked papers 67
Table 18: Top ranked papers in Validation 68
Table 19: Details about SODE 74
Table 20: Empirical correlation between ratings 77
Table 21: Overview PAs in relation to amount of solutions and their quality 79
Table 22: Research Questions for the evaluation of SODE 88
Table 23: SODE impact summary for RQ1 94
Table 24: SODE impact summary for RQ2 98
Table 25: SODE impact summary for RQ3 101
Table 26: Function Plan 109
Table 27: Variance-Covariance Matrix 111
Table 28: Covariance Matrix 111
Table 29: Eigenvalue Matrix 113
Table 30: Eigenvector Matrix 113
Table 31: Sorted eigenvalues and eigenvectors 114
Table 32: Factor loadings and explained variance 116
Table 33: Overlapping between variants 121
Table 34: Overlapping control over time 123
Table 35: New Factor scores 124

1 Introduction

Globalization is a reality in today's connected business world. Almost no company today, not even small ones, are independent of the world wide economy. This has particularly been shown in the financial crisis in the past two years. For the software business, globalization is nothing new, though. The first small steps to a globally-distributed software industry were done in the 1960s [1]. At first, many companies went abroad for cost reasons [2], [3]. Therefore, low-wage countries like India have been strongly involved in the global software industry [4].

While cost reasons did not always pay off, the trend to develop software in globally-distributed teams was still growing. One reason for this might be that it is very easy to transport software from one end of the world to the other with very small cost, not like physical products such as cars that need to go by ship or plane. This makes it fairly easy to just try on "Global Software Engineering" (GSE) [1]. Having made the first step into GSE, many companies realized that there are many other reasons to stay involved in these projects even though cost have shown to be much higher than expected [2][3]. Reasons to keep on working globally distributed reach from proximity to market [5] to the access to qualified engineers [6].

However, there are challenges in GSE. Most of these challenges can be reduced to coordination or communication problems. Having a virtual team that is separated by time and distance and has to work closely together in one project, communication can suffer quite badly [7][8]. With bad communication, coordination of project tasks and workload cannot be distributed as instantly as in traditional, co-located teams. Another problem in GSE is creating the awareness of remote team members [9][10]. It happens quite easily that team members in remote locations are left behind. One of the major problem areas in GSE, however, is cultural issues [11][12][13]. As there might be different nationalities involved in a team, some might have to switch to a non-native language. Also, cultural differences might lead to misinterpretation or offense within the team. The pitfalls in GSE projects are various and interdisciplinary.

Obviously, there are multiple problems that arise in GSE projects and these problems are also published by practice [14][15][7][16][17][18][19][20][21]. This shows, that these problems are real, occurring on many projects and bringing big trouble to industrial projects. There does not seem to be enough solutions out there or not enough support for industry to identify solutions for their context. Yet, the number of published solutions has risen in the past few years. In Figure 1, there is clearly a strong increase visible in the number of papers

published in the last 5 years. These numbers had been gathered until March 2009 which means that the data for the second half of 2009 is missing when comparing to the years before. But, there seems to be a strong research interest in this area considering that the 2009 bar is only representing the first three months of 2009.

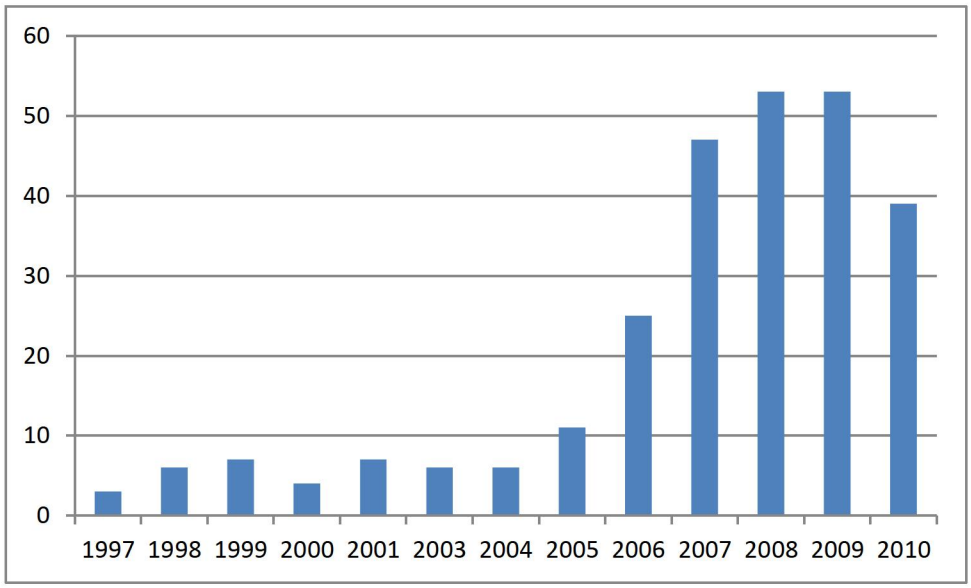

Figure 1: number of solutions published per year

This indicates that GSE and the problems in GSE projects are real. The close cooperation with an industry partner, Daimler AG Research and Development, confirmed the presence of these projects and problems in practice. Therefore, a SOlution-based framework for Distributed Engineering (SODE) has been created and will be presented as the major result of this thesis. This framework provides practitioners with relevant solutions for their daily GSE work. These solutions have been identified by an extensive systematic literature review. One of these solutions has been developed especially for Daimler AG and has been integrated into SODE. To ensure relevance for practice, the usability and usefulness of SODE have been evaluated in academia and practice.

1.1 Background and Related Work

Globally-distributed work is everyday business in some companies today. However, there is still potential for conflicts. One main reason is unclear terms and names for similar activities. Distributed work is being referred to as Global Software Development (GSD) when talking about software development only[22]. But the same problems arise in distributed engineering or any kind of virtual teams. Virtual teams are working closely together on the same project but are separated by time and/or distance [23][11][24][25][26].

As there are many names for the same thing, a clear definition of what this thesis is dealing with is needed. Here, the term GSE will be used.

Global Software Engineering…

… is traditional software engineering with the added complexity of virtual or distributed teams. That is teams, that are spread over different sites, organizations, timezones and cultures. The complexity of GSE problems directly arises from the added complexity of this work environment.

This means, that GSE combines the problems arising in distributed work, independent which product is developed, and traditional software engineering.

Another term that will be used a lot in this work is solution, which is not used in a mathematical way but rather describing anything that can be used to tackle problems arising in GSE projects. This could be methods, e.g. for requirements elicitation, or tools, e.g. video conferencing tools, or techniques that simplify daily work in a distributed team.

Talking about GSE brings along a lot of terms that are usually referred to as GSE. These are some of the main types of GSE that will be part of this work, at least for the industry partner Daimler AG.

- Outsourcing[3]
  Companies give inhouse functions to a supplier that takes care for these functions from then on. This happens a lot with IT services. Companies might outsource everything that does not belong to their core business.
- Off-shoring[17]
  Companies give projects to remote locations like India or Ireland. In the remote location, other companies can be involved but there can also be subsidiaries. In this setting, cultural problems are observed in many projects.
- Near-shoring[17]
  This is similar to offshoring with the difference, that the companies are not located too far apart. For example, the USA near-shore to Ireland which is just a few time zones away and the culture (e.g. language, religion) is comparable.
- On-shoring[27]
  This describes a trend to retract the activities once offshored into the same continent and time zone. It is based on the problems that arise in the above-mentioned settings, especially those involving cultural issues and time zone differences.

In this list, outsourcing is a little different to the rest. The last three types are clearly separated by the distance between the collaborating partners. Outsourcing is independent of the distance between the partners and describes a business practice. To make things even worse, there is more complexity to this system. Every type of GSE can be performed by either a company's own employees who are just working abroad or strangers who have been hired to do this job from the start and never worked in this company before. There is a huge difference with regards to integration of these people. In this case, the employees, may they be new or veterans, belong to the company and should get the feeling that they do. Another possibility in all types of GSE is to use a subsidiary or a supplier for the project teamwork. Then, the people working on this project do not belong to the company but have a temporary work assignment. This also makes a huge difference in the cooperation within the team.

All of this results in quite complex settings and various different working styles in GSE projects. However, the overall problems and challenges in GSE are quite similar.

1.1.1 Challenges in GSE

So, GSE seems to be a reality in today's industry in many different shapes and types. But it also seems to create a whole lot of new problems. To provide some insight into problems related to GSE projects, problems that are often mentioned in relation to GSE projects are collected in Table 1. Each challenge is numbered and well referenced to show that this is a problem in reality.

ID	Title	Description	Reference
Ch 1	Complexity	Different hierarchy at different sites/partners leads to complex structures. These result in problems in common projects, e.g. responsibilities and redundancy.	[17] [28] [29] [30] [31]
Ch 2	Inconsistency	Different processes lead to inconsistency. This may happen when work products, developed at different sites, turn out to be incompatible.	[17] [32] [33]
Ch 3	Lack of trust	Globally-distributed teams often face a lack of trust and commitment within the virtual team. This is often due to a lack of social ties.	[14] [15] [7] [16] [31] [34]

Ch 4	Cost	Increased cost (logistics, travel) while planning to decrease cost by, e.g., offshoring, is a huge problem in GSE. Often, projects fail due to budget problems.	[35] [36] [14]
Ch 5	Terminology	Terminology differences due to different level of education hugely hinder communication between team members and slow down project work.	[17] [37] [38] [39] [40] [8]
Ch 6	Fear	The fear of being replaced or fired due to the outsourcing turns out to be a big problem for motivation and job performance.	[41] [42] [31]
Ch 7	Effort Estimation	Wrong effort estimation for distributed projects is one of the major problems for project management. These problems often arise from underestimating the efforts for transferring a local project to a distributed project.	[43] [2] [44] [45] [3]
Ch 8	Shared vision	A lack of common goals and shared vision in virtual teams can often be seen due to missing or weak social ties in virtual teams.	[41] [46] [47] [48]
Ch 9	Time zone differences	Time zone differences complicate communication as it often reduces to asynchronous communication. Answering questions often takes a day or more.	[17] [18] [19] [20] [21] [31]
Ch 10	Training of remote teams	The training of teams is more complex in a distributed environment. It results either in extensive and expensive traveling or in using non face-to-face communication.	[49] [50] [51] [52] [53]
Ch 11	Cultural issues	Cultural differences are a problem for group cohesion and communication. Overcoming cultural differences is very time-consuming and complicated.	[17] [54] [55] [56] [57], [41]
Ch 12	Missing social ties	Lack of social ties is slowing workspeed	[58] [59]

		down as e.g. personal strengths are not obvious for the entire team. This slows the completion of the task down.	[60] [61] [62], 75
Ch 13	Coordination problems	Coordination and communication are problematic in distributed teams as it takes additional effort to keep remote team members in the loop all the time.	[17] [63] [64] [65] [66] [52] [67] [68]
Ch 14	Motivation	Motivation of remote team members is hard as they often feel like communication is bypassing them and decisions are made without them.	[46] [69] [70] [71]
Ch 15	Awareness	Awareness of remote team members is necessary to keep good team work alive. The possibility to assign the right tasks to the right persons is a complicated problem in remote teams.	[72] [37] [73] [74] [75] [76] [31] [77] [78]
Ch 16	Communication	Eliciting needs from distributed customers is a challenging task. Due to different educational knowledge, some implicitly integrated features might not be included in the product due to communication problems.	[27] [79], [80] [81] [82] [83]
Ch 17	Up-to-date work environment	Changes, e.g. in requirements, are hard to keep track of in distributed teams. The awareness to communicate everything relevant for the project to all team members has to be present.	[27] [84] [85] [86] [87] [83]
Ch 18	Product integration	Product integration in distributed engineering is more complex as a lot more preparation in advance is necessary to ensure compatibility.	[88] [89]
Ch 19	Distributed reviews	Performing reviews in distributed teams is more laborious due to the fact that it is hard to get additional information for the product. This makes the review process more time intensive.	[90] [91] [92]

Ch 20	Slower work speed	Change requests are resolved slower in virtual teams. In average, it takes more time gathering the information to resolve an issue if the team members are located all over the world.	[22] [10] [93]
Ch 21	Work environment	Differing work and project environment are problematic and happen a lot in distributed teams. This means, that people working in a distributed team are often members of a local team as well. The switch between these contexts can be very hard.	[94] [95] [96] [97]
Ch 22	"Not invented here" problems	Solutions developed remotely are often not implemented correctly or do not fit perfectly. Sometimes, these solutions are even actively blocked.	[94] [98] [11]
Ch 23	Barrier in communication	There is an information barrier between remote team members. This starts with communication issues and reaches to technical problems like remote access to networks etc.	[94] [61] [8] [99] [31] [100] [101]
Ch 24	Language problems	Language switch-backs are hindering communication. This happens a lot in sub-teams with common language. These tend to isolate themselves and use their native language.	[94] [19] [91] [68]
Ch 26	Transparency	Limited transparency (knowledge, work products, lack of solutions, etc.) brings along problems in speed and quality of work due to a non-ideal distribution of workload etc.	[17] [102]
Ch 27	Solution focus missing	There are mainly problem descriptions present in GSE while there is a lack of good solutions. The solutions present in GSE often arise from solutions that have not even	[103] [104] [105]

		been designed especially for GSE.	
Ch 28	Guidance missing	While there are solutions for GSE, there is a lack of guidance relevant for practice. This means, solutions helpful for GSE need to be presented and provided in a way that actually helps industry to use them.	[103] [104] [105] [31] [106] [107]

Table 1: Problems in GSE

These problems are all real world problems and mentioned a lot in literature. Therefore, the problems in this area seem to be hugely complex and of various sources. This is motivation enough to go into more details and work on solutions for at least some of these challenges. However, as the problems seem overwhelmingly huge, one of the first steps to be made is finding a way to structure these problems in a more handy way.

In literature [17], [27], several ways to structure GSE-related problems have been presented. The following four dimensions of distribution are the best approximation to the Daimler AG context in several ways:

- Temporal
  Temporal distribution occurs if the project team is spread over several time zones or different shifts. Examples are Ch9, Ch13.
- Geographical
  Geographical distribution occurs if the project team is spread over several locations. These locations might be located in one time zone. Examples are Ch7, Ch15.
- Cultural
  Cultural distribution can be found in teams who do not speak the same native language, do not come from the same cultural background or might not be familiar with the others' traditions. Examples are Ch3, Ch5, Ch11.
- Organizational
  Organizational distribution occurs if several companies work together on one project. This might result in communication gaps as the information might not be shared. Examples are Ch1, Ch18, Ch22.

Most of the structures for GSE problems have much more sophisticated dimensions in cultural issues. For example, cultural dimensions by Hofstede [108] have been used widely and break down the cultural dimension used in this work into four sub-dimensions. However, cultural issues are not the focus of this work, so there is no point of going into detail here. Much more important for the industry partner is adding an organizational dimension to this

structure. This is due to the presence of several subcontractors in this project [109], [110]. Temporal and geographical distribution is mentioned in almost all publications focusing on GSE [111][112][113][104] and an obvious part of GSE problems. These four dimensions fit the Daimler AG context best.

However, the structuring of these problems helps understanding them and even supports finding a solution, but it is not really scalable. This means, if there are several hundreds of solutions, the structuring according to these dimensions is not detailed enough. These dimensions, however, do a good job in structuring the input from an industry partner to identify the most urgent problems at hand. To structure the solutions themselves, there has to be something better.

So, either the structure is refined a lot, which would mean way beyond Hofstedes cultural dimensions [108], or there could be other structures available that already provide this sort of structure. A combination of the dimensions and the types of GSE (see Section 1.1) might be a good structure. It turns out, though, that there is no clear mapping between the problems and this new structure. Obviously, there is a difference in geographical distribution between off-shoring and on-shoring that this structure would not be able to represent. Also, this structure would lead to enormous redundancy in the structure as many problems arise in any type of GSE. As there is no ideal and straight forward structure for GSE problems, existing frameworks and models have been studied. Ideally, these should already be well-known and used in industry as well as academia.

1.1.2 Maturity Models and Best Practice Frameworks

When developing commercial software, process models are widely used in industry. Therefore, different widely-used process models have been used as an input for this work.

1.1.2.1 The Capability Maturity Model Integration

The *Capability Maturity Model Integration* (CMMI) is a systematic preparation of practices and processes that provide guidance for improving the company [114]. CMMI can also be used to assess the processes used in a company to rate the company according to maturity levels.

1.1.2.2 ISO/IEC 12207

Another process model in this area is the *ISO/IEC 12207* [115]. This norm contains software lifecycle processes that can also be tailored to a specific company. The process assessment standard SPICE ISO/IEC 15504 is closely connected to this norm.

1.1.2.3 The Guide to the Project Management Body of Knowledge

The *Guide to the Project Management Body of Knowledge* (PMBoK Guide) [116] is a widely-used project management standard especially in the US. The PMBoK Guide is also used for the certificate Project Management Professional.

1.1.2.4 The Requirements Engineering Process Maturity Model

In addition to these process models that include the complete lifecycle of a software development, solution-focused process models have been used as an input for this work as the focus of this work is to provide solutions to industry.

The Requirements Engineering Process Maturity Model (uni-REPM) [117] is one of those. It provides practitioners with practice relevant actions. The main difference to this work is the focus on a specific part of the lifecycle, requirements in this case.

1.1.2.5 GSD Knowledge Base

The GSD Knowledge Base [17][118] provides best practices from GSE projects. This project focuses on organizational learning and knowledge management. It also contains a Risk Barometer that tries to predict project risks based on historic analyses of GSE projects. According to these risks, practices are provided as guidance for project managers.

It turns out, though, that none of the above-mentioned frameworks and models can completely fulfill the needs of this work. This, to some extent, is due to the fact that the industry partner has special needs for a framework supporting their work. Another reason is the special problems that arise in distributed projects. To develop a framework fitting the needs of industry better, the context of the industry partner has to be closely examined.

1.1.3 Research Context

The research presented in this thesis included a close cooperation with an industry partner. Different stages of this work, e.g. problem description and evaluation, needed industry input to ensure practice relevant results. These steps have been performed at Daimler AG Research and Development.

1.1.3.1 Daimler AG

Daimler AG is a German premium car, van, bus and truck manufacturer. Group Research within Daimler AG closely works together with well-known brands like Mercedes Benz Cars and Mercedes Benz Trucks, among others.

In a modern automobile, software is a fundamental part of the product. An estimated 90% of the innovation in a car is related to software, and about 80% of a car's features are based

on electronic systems [110]. To implement these functions, about 10,000,000 lines of code distributed over about 70 control units have to be written. Up to 40% of the cost of a car lies in software and electronics [119]. Software projects in the automotive domain are very extensive, which is evident at Daimler AG. Therefore, problems and complexity pertaining to variants are part of daily work in software engineering at Daimler AG, which is also complicated in a distributed environment.

1.1.3.2 Daimler AG and GSE

Mercedes Benz Trucks is an internationally-oriented department within Daimler AG. Working at different sites in Germany, the USA and Japan, they are developing software for various markets.

Daimler employs about 10,000 people in Research and Development at 20 sites in nine countries. The locations reach from Palo Alto, CA, USA, to Bangalore, India, to Bejing, China and Tokyo, Japan, to mention some. The number of people employed on these sites is mostly around 50–200.

1.2 Thesis Contribution

Having stated the main problems and background for this research, it is now time to formulate the main Research Questions (RQs) in Section 1.2.1. After that, the methods used to answer these questions are shown in Section 1.2.2. In the end, all these thoughts are combined in Section 1.2.3 to outline the structure of this thesis. This section will provide a map of this thesis based on previously mentioned facts to guide the reader through the structure of this thesis.

1.2.1 Research Questions

In this section, these main RQs will be explained. They will be stated and explained referencing to the sections in this thesis where these questions will be answered. Detailed information on the process, analysis and results for each RQ can then be found in these sections. Also, the RQs will be mapped to the challenges in Table 1.

RQ	Description	Challenges
RQ1: What solutions exist that address the GSE challenges?	Many empirical studies and systematic literature reviews focus on the identification of challenges. Section 1 presents the first SLR collecting and analyzing solutions associated with GSE by also evaluating the level of empirical validation. The solutions identified have also been rated regarding the context and design description. The overall results of the systematic review reveals that over 330 papers containing 127 solutions were identified. The detailed results can be found in section 3.4.	Ch 26, Ch 27, Ch 28
RQ2: To what degree can the solutions found in RQ1 be formed into a GSE reference model using well-established models as a base?	In Section 2 a GSE model is introduced, designed to categorize solutions into process areas, useful for the analysis of the research community's contribution to state-of-the-art and identifying fundamental gaps in research. This model has been based on well-known process frameworks like CMMI [120] and ISO/IEC 12207 [115]. The reason for this is to increase acceptance and productivity.	Ch 2, Ch 4, Ch 13
RQ3: Given the combination of reference models into a GSE reference model, to what degree do new process areas need to be created, and to what degree are solutions missing?	The model categorizes the solutions from the SLR. It is populated with references and good examples, useful for practitioners, who can use the model to find solutions to overall challenges in various process areas, and researchers, who can identify gaps in research. The detailed results of empty spots and added process areas can be found in Section 3.4. The analysis is separated into a quantitative part and a qualitative part.	Ch 2, Ch 11, Ch 26

RQ4: To what degree is the created GSE reference model usable and useful?	The evaluation of the model can be found in Section 4. The evaluation has been separated into two parts, expert interviews and evaluation workshops. The expert interviews mainly reviewed the completeness of the framework contents. The evaluation workshops presented the framework to a larger group of people focusing on the usability and usefulness of the framework.	Ch 2, Ch 3, Ch 15, Ch 23, Ch 28

Table 2: Research Questions

1.2.2 Research Methodology

Having defined the direction of the research for this work, the way to achieve the results to answer the RQs has to be described now. As there is no possible way to plan for good ideas, there is a way to ensure that the results produced along the way are meaningful for both industry and academia.

This section focuses on the path that has been taken to produce the results presented in this work and therefore has a temporal structure.

1.2.2.1 General Research Approach

Having stated the RQs in Section 1.2.1, this section will explain how these questions were answered. The process of gathering the results can be found in Figure 2 [121]. In this figure, the high level process to gain the results presented in this thesis is shown. There is a separation between industry and academia represented by a dotted line. This is used to indicate in which area the results produced in these steps have mainly been created. For some steps, there has been close cooperation between industry and academia, therefore, these steps are shown on both sides of the border.

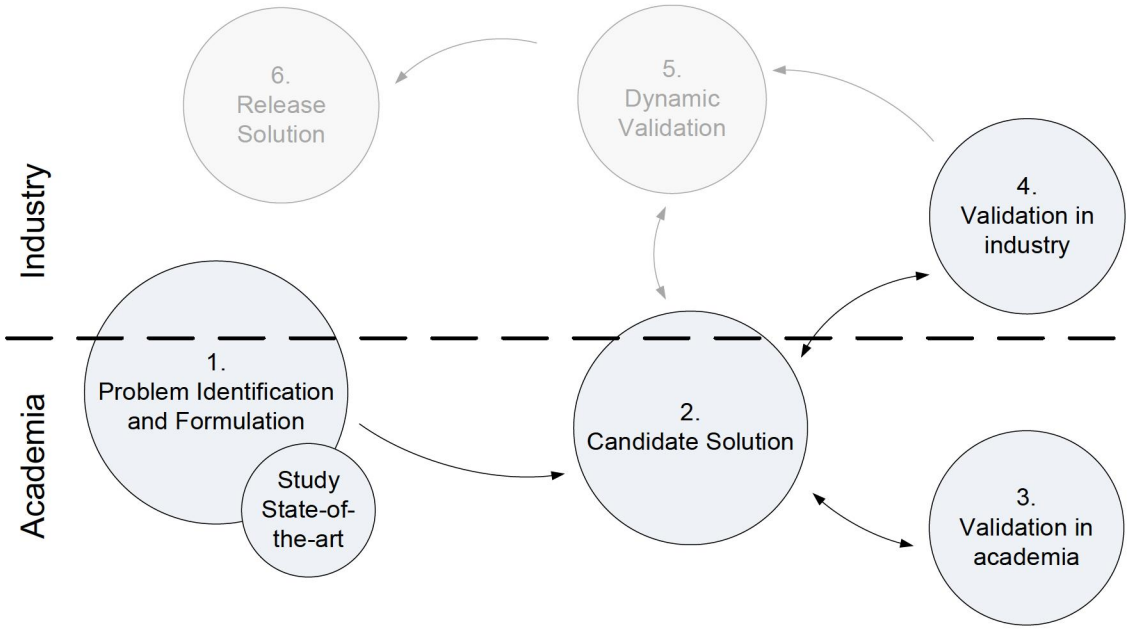

Figure 2: Research Method

The steps that have led to this work are now explained in more detail and are also connected to the structure of this thesis.

Step 1. Problem identification and formulation

First, it is very important to find out about the problems that are present in industry. By identifying problems, relevant for maybe different parts of industry, research can be focused and conductive. After identifying problems in industry, a concrete problem formulation can be stated. This is important to narrow down a big research field into a concrete problem that can be solved.

In this step, it is also important to check whether this problem has already been solved by studying state-of-the-art.

Chapter 3 is the result of this step. It shows the systematic literature review (SLR) and its results. The SLR was performed according to the guidelines of Kitchenham [122]. This is a good way of getting an overview of a research area and was mainly developed for medicine. However, it has been adapted to Software Engineering and has been used several times already providing helpful insight into various areas [73], [123].

Step 2. Candidate Solution

Based on the concrete problem formulation, a candidate solution can be developed. This is a solution based on theoretical research that is supposed to solve the problems found and stated in Step 1.

As shown in Chapter 2, SODE was created to guide practitioners in GSE problems. As an input for this, several well-known and widely-used process standards have been used [114]–[116]. With the added information of the SLR performed in Step 1, SODE has been well prepared for its validation.

Step 3. Validation in Academia

The candidate solution has to be evaluated to prove its relevance. This is first done in academia. This evaluation is used to find out whether the solution is based on good scientific research and if this research is used in an adequate way to produce a meaningful candidate solution.

The results and feedback on this step are shown in Chapter 4 and are used to refine the candidate solution. The method used to validate SODE in academia was a semi-structured interview [124]. This has been chosen to give the interviewer enough freedom to interact with the interviewee but provide a structure that guides the interviewer towards the most important questions to be answered.

Step 4. Validation in industry

This validation step is done in industry by presenting the candidate solution to the industrial clients. Yet, this validation is done in a controlled environment. This means that the solution is not yet used on a real project but in a workshop.

The results and feedback from this step are used to refine the candidate solution to match industry needs.

The results of this step can be found in Chapter 4. Other than Step 3, this validation has been done in group workshops [124]. Due to limited time resources in industry and the wish to gather as many people as possible, a workshop has been the best choice. Along with questionnaires, these workshops have provided the input from practice to evolve SODE even further and tailor it towards the needs of industry.

Step 5. Dynamic Validation

In this step, the candidate solution is implemented on a real life project and used by the team working on this project. This enables in depth analysis about the usefulness of the candidate solution by keeping track whether the solution is actually used or not.

The results and feedback on this step are used to refine the candidate solution.

Step 6. Release Solution

Finally, these steps result in the release of a solution. Having created and constantly refined the candidate solution, it should lead to a solution which meets the original problem identified in Step 1.

In this work, this process has been followed until Step 4. Step 5 and Step 6 will be performed at Daimler AG in cooperation with the University of Hannover and the project GloSE.

1.2.2.2 Research Methods Utilized

As the goal of this thesis was to do research that is relevant for industry, the methods used to identify and solve problems were mainly empirical. This is due to the fact that practitioners from industry have been integrated in the research in early stages and the input had to be gathered in a structured way to ensure traceability along the way. The methods used in this thesis to gather data were surveys, action research and a systematic literature review.[125][126][122][127]

Surveys are a method used to systematically collect input from a selected sample of individuals in order to be able to use statistical methods to analyze these input data and draw conclusions. The selection of this sample is important as it limits the generalizability of the conclusions drawn from the analysis. Usually, the generalizability of results depends on whether the selected group is representative or not. In theory, a randomly selected group of people is ideal from a generalizability point of view, however, in practice the target group of people usable, usefull and willing to take part in surveys may be limited. Also, the basic population might be limited to include a special group of people, e.g. subject matter experts, into a otherwise randomly selected sample. The generalizability may be limited, but the results might be more meaningful in order to develop results relevant for practice. [128][129]

Action Research describes a research method used especially in industry related. Action research observes and studies social systems and then changes parts of these systems to improve them. This method is always based on four main steps:

1. Planning of the intervention
2. Intervention
3. Observing the effects of the intervention
4. Reflection and planning of further actions

Action Research is focused mainly on a steady change of processes and work environments to improve and optimize in close collaboration with practitioners. As the four steps show, action research can be done iteratively to improve and optimize in several loops. [129][130][131]

A *Systematic Literature Review* is a scientific method to identify peer reviewed, published literature relevant for a specific area and/or problem. The method used in this thesis is described by Kitchenham [122] and widely used in empirical research. Here, the systematic literature review is combined with a mapping of the identified solutions into process areas to structure the results and make them easily usable. These methods can be used to structure a relatively new field of research and to identify potential research focus areas in white spots on the map. Also, the methods can help to identify why some areas have no need for further research which means that some white spots on the map might be left white intentionally as there is no need for research in this area. [127]

1.2.2.3 Data Collection Methods Utilized

To validate the results of this thesis, qualitative as well as quantitative data have been collected. These data, especially the combination of these data, can be used to produce valuable insight into the results and the possible improvement of the work done. Qualitative data is mainly used to ensure validity of the data, quantitative data is mainly used for measurable data analysis. In this thesis, mainly the following three methods have been used.[129][132]

Interviews are a widely used method of data collection. This method is very flexible as there are many different ways of conducting interviews, e.g. depending on the types of answers the interviewee can give (closed vs. open questions). The interview is especially helpful in situations where there is some guidance needed to reach a certain point or answer a specific question but offer the interviewer the possibility to raise follow-up questions along the way. Semi-structured interviews offer this flexibility but, on the downside, produce a huge amount of data that needs to be analyzed. In this thesis, interviews were mainly used to conduct a walkthrough of the model and collect improvements with respect to the model structure as well as the model contents. So, the interview offered the structure to cover the area of interest for this interview and still provides the flexibility to go into detail wherever needed.[129], [132]

Questionnaires are a form of data collection that is used to gather data unobserved, filled in by the individuals on their own. Open and closed questions can be used in questionnaires. The danger in questionnaires results from misunderstandings of questions and thus wrong or misleading answers. Therefore, closed questions are preferred in questionnaires. Also, closed questions can be transferred to numbers and can therefore be analyzed a lot easier than the unstructured answers in interviews. Also, these data can be used in a wide range of visual as well as statistical interpretation of data. [129]

Workshops can be seen as discussions and are used to bring different opinions, roles, organizations etc. to the same table. This method has been used to gain insight into areas that have shown to be unclear or that need further discussion. The complexity of workshops shows while setting up the meeting and getting everybody into one room at the same time as well as handling and interpreting the unstructured and huge amount of data collected. [129]

1.2.2.4 Data Analysis Methods Utilized

The data collected during the creation and evaluation steps of the thesis have to be analyzed to draw conclusions and gather meaningfull results. This is especially true for results aimed to industrial practice to prove the benefits resulting from the changes proposed.

Quantitative Analysis uses descriptive statistics and other mathematical methods to describe and combine data. This is especially helpful to identify outliers or other deviations within the data that might lead to improvement or further understanding of the factors driving the data. These methods have been used in the development of FAM (see Chapter 5) as well as the evaluation and validation of SODE. [126], [133]

All methods used along with more detailed reasoning on the methods chosen as well as the validity threats are discussed in the chapters they are used. There, the connection of the methods to the results they produce are given directly and are much easier to follow. Following now is a summary of the contents and contribution within the chapters of this thesis.

1.2.3 Chapter Summary and Contribution

In this section, the main commitments of this thesis will be connected to the structure of this thesis. Everything mentioned in the last few sections will be connected here and mapped to the structure of the thesis. The challenges in Section 1.1.1 that led to the RQs in Section 1.2.1 are now connected to the contributions on page 10. This is done to provide a complete overview and a short summary of this thesis.

1.2.3.1 Solution Model on Global Software Engineering

First and foremost, the creation of a framework to guide practitioners has been the goal of this work. The initial creation of SODE is shown in Chapter 2. As this was a main result, it answers RQ2 as shown in Table 3. This RQ asked for a solution-focused framework for GSE projects and was based on several challenges. These challenges mainly focus on process-based problems and project management tasks. Therefore, some of the contributions of this work answer RQ2 and solve some challenges of GSE problems.

Contribution (page 10)	Challenge (Section 1.1.1)	RQ (Section 1.2.1)	Chapter
ii, v	Ch 2, Ch 4, Ch 13	RQ 2	2

Table 3: Overview over Chapter 2

Chapter 2 will give detailed insight into the creation of SODE. This work was based on well-known and widely-used process frameworks such as CMMI [120] and ISO 12207 [115]. This initial creation was a raw version of SODE, SODE 0.2.

1.2.3.2 Literature Review and impact on SODE

Based on the results of Chapter 2, SODE 0.2, a systematic literature review has been conducted. The process and results are presented in Chapter 3. The SLR is the answer to RQ1 which was based on several challenges. RQ1 asked if there are any solutions for GSE problems presented in literature and whether these can be used or not. Based on challenges that focus on the guidance of practitioners and presentation and structure of solutions, the main amount of contributions has been produced in this section as can be seen in Table 4.

Contribution (page 10)	Challenge (Section 1.1.1)	RQ (Section 1.2.1)	Chapter
i, iii, iv, vi, vii, viii, ix	Ch 26, Ch 27, Ch 28	RQ 1	3

Table 4: Overview over Chapter 3

Literature reviews in GSE mainly focused on either small sub-areas of GSE or on problems in GSE. Therefore, the systematic literature review was a major task for the completion of SODE. Along with other contributions such as rating of the solutions or the mapping of these solutions, the SLR was the main scientific contribution of this thesis.

As a result, SODE 0.4 was developed. This was mainly the combination of the SLR and SODE 0.2 and provided real added value by providing practitioners with solutions that are integrated into a well-known process framework. These solutions have also been rated for relevance and validity.

1.2.3.3 Finding and filling the gap

Having searched the literature for solutions in GSE, some areas have remained empty in SODE. One of these areas has been chosen in close cooperation with the industry partner to be filled with a solution especially designed to solve a problem for the industry partner.

The problem chosen to be solved was the reduction and control of variants in software products. This has proven to be a problem in distributed teams. This can be taken as an example for the answer of RQ3 that asked for empty spots or missing process areas in SODE. RQ3 has been based on several challenges dealing with the need for solutions in GSE as can

be seen in Table 5. The solution developed as an example how to fill these empty areas is another contribution of this work.

Contribution (page 10)	Challenge (Section 1.1.1)	RQ (Section 1.2.1)	Chapter
i, x	Ch 2, Ch 11, Ch 26	RQ 3	5

Table 5: Overview over Chapter 5

Detailed results on the search for empty spots and missing process areas can be found in Section 3.4. The solution that has been developed for the industry partner can be found in Chapter 5.

1.2.3.4 Evaluating SODE

SODE 0.4 has been filled with theoretical knowledge and developed to a state that enables practitioners to work with it. However, to ensure easy usability and practice relevant usefulness, SODE has been evaluated in academia and practice. This answers RQ4 that asked whether SODE was usable and useful and solves challenges of GSE projects. The challenges and contributions to solve those can be found in Table 6.

Contribution (page 10)	Challenge (Section 1.1.1)	RQ (Section 1.2.1)	Chapter
vi, viii, ix	Ch 2, Ch 3, Ch 15, Ch 23, Ch 28	RQ 4	4

Table 6: Overview over Chapter 4

Detailed results on the evaluation of SODE 0.4 can be found in Chapter 4. Also, the impact on SODE and the development to SODE 1.0 can be found there. SODE 1.0 is the first result that has been checked for relevance in practice and has resulted in quite promising results. It has been evaluated with the main industry partner as well as other industry and academia partners.

1.3 List of Publications

1.3.1 Publications included in this thesis

Schneider S., Gorschek T., Torkar R., “Solutions in Global Software Engineering: A Systematic Literature Review”, International Journal of Information Management, volume 33, issue 1, 2013, pp. 119 - 132.

Schneider S., Gorschek T., “Software Variant Reduction and Control through Factor Analysis”, submitted to the International Journal of Software Engineering and Knowledge Engineering.

1.3.2 Publications not included in this thesis

Stupperich M., and Schneider S., “Process-focused Lessons Learned from a multi-site development project at Daimler Trucks”, ICGSE 2011, Helsinki, Finland.

Li K., and Schneider S., “A Special Singular Eigenvalue Problem”, Hawaii International Conference on Statistics, Mathematics and Related Fields 2007, Honolulu, Hawaii, USA.

2 Solution framework for Distributed Engineering (SODE)

2.1 Introduction

Chapter 1 has shown that globalization is a fact that industry has to face and software engineering has been affected by offshoring and outsorcing[1]. The Global Software Engineering (GSE) research community has produced many studies focusing on challenges and needs of distributed development environments. Many of these focus on problems and issues in GSE, though[103]. As it is important to know the pitfalls of GSE, it is much more important to solve these problems and especially provide practitioners with industry relevant solutions for their daily work.

The normal way to solve those challenges is to use frameworks, models, best practice frameworks and other guidelines. However, the ones available do not take care of the problems present in industry [115], [120]. Therefore, this chapter proposes an extensive GSE framework called Solution framework for Distributed Engineering (SODE). This framework was developed in close collaboration with Daimler AG, where it was also validated as can be seen in Chapter 4. SODE builds on established models such as CMMI [120] and ISO 12207 [115], but focuses on GSE topics. This ensures the compatibility with processes at Daimler AG. SODE also establishes a solution focus which means that the framework uses state-of-the-art research and practices to structure them. The structure along with the solutions offered in SODE result in a benefit for practitioners that use SODE as it offers them real life solutions within their known structures.

This chapter presents SODE and how it was developed and refined iteratively through validation at Daimler AG. In Section 2.2, the background and related work will be described. In Section 2.3, the research methodology will be explained and the path to the creation of the model will be described. In Section 2.4, the initial version of SODE 0.2 along with the additions and changes within the framework will be described.

2.2 Background and Related Work

This section discusses related work, elaborates on the industrial collaborating partner and the work context in relation to GSE. Background and motivation to the study and the creation of SODE is also described.

2.2.1 Background

Sa et al. [134]present a possibility to combine different processes for GSE. While presenting a prototype support framework to define a unified process, they focus on the technical challenges to set up a unified process over company boundaries.

Lane et al. [135] present an experience-based framework for GSE projects. This framework presents and analyzes experiences in GSE and structures these experiences according to several dimensions like distance and process. However, they do not enable integration into other frameworks like CMMI or ISO.

Salger et al. [136] present an assessment framework for industrial projects in GSE. They especially focus on the tailoring to different settings in GSE projects like offshoring, outsourcing and others. They also focus a lot on requirements specification. The framework is based on long-term work with industry and is mainly fed by industry input. However, they do not offer solutions to practice.

Aalst et al. [137] present a framework for agile approaches in GSE. Becoming increasingly popular, agile approaches are hard to implement in an distributed environment. However, Hossain et al present a case study that shows some best practices to implement agile approaches in GSE. However, the authors only cover a very small piece of GSE and are therefore a part of the puzzle but not the complete picture.

Islam et al. [138] present a framework for risk management in GSE projects. Risk Management, being a core part of GSE projects, is integrated in a framework to assess and manage risk in such projects strictly but with reasonable cost. The framework can also be tailored to different types of GSE projects.

These are all important steps on a way to managing a GSE project, but, they are only parts of the picture. There is no experience based framework that provides industry with relevant solutions and best practices for GSE. All results are either only solving a small, yet important part of GSE or lacking solutions that are relevant for practice. Therefore, there is a need to create a framework that provides industry relevant solutions for GSE projects as a whole so there is no need to use different frameworks for different tasks. Also, the connection to widely-used frameworks and models like CMMI and ISO are important and often missing.

2.2.2 Frameworks and Models

As a basis for SODE, CMMI [120] was used because of the easy connectability of CMMI to the V-Model which is extensively used at Daimler AG to develop Software [110]. Primarily, CMMI is used to improve the work of an organization. It can, however, also be used to find out the maturity level of this organization and identify areas with room to improve even further. As

a basis for SODE, CMMI is the ideal choice because it is widely spread and well established in practice and at Daimler AG.

Another input for SODE was ISO 12207 [115]. This has also been used because it is widely spread and well established and therefore is an important input for SODE for completeness reasons. ISO 12207 has a strong focus on software lifecycles. It defines a process structure and terminology but does not force specific development methods. The focus of ISO and CMMI is not to tell the user "how" to do things but "what" to do to get a high quality product.

Also, PMBoK [116] has been used as input for SODE because of its relevance in industry. This framework is more a collection of best practices and knowledge for project management. Yet, it is very process focused. Also, it is not limited to software development as e.g. ISO 12207 is. This brings another input for SODE which helps to make sure nothing was left out and to make it more relevant for practice.

2.3 Creation of SODE

SODE was created in several steps. In this section, a first insight into SODE will be given to enable the reader to understand the following chapters. A complete overview of SODE can be found in the appendix[1]. There, the last version of SODE 1.0 can be found including all the changes that will be taken in the next chapters. The steps that have been performed to create SODE are now described in more detail and can also be seen in Figure 3.

2.3.1 STEP 1: Adaption of CMMI

As a basis for SODE, CMMI [120] was chosen. As CMMI is already widely used at Daimler AG and elsewhere in industry, this was the best way to ensure compatibility with established processes as well as user acceptance. The main process areas (process management, project management, engineering and support) have completely been taken from CMMI as these give the first structure to the model. On lower levels, the process areas have been integrated into the model by analysing them in a scientific context (please note Section 2.3.2)

SODE can be seen in its final stage in the appendix[1]. The dark Process Areas (Pas) are the ones taken from CMMI. For example, *PA 1.1 Organizational Training* has been integrated into SODE. This means that trainings have to be done distributedly at Daimler AG and that there is always room to improve. Trainings have also been identified to be a challenge by [49].

[1] Appendix C: SODE 1.0

2.3.2 STEP 2: Substitution based on ISO, ACM and PMBOK

To complete the structure of SODE with theoretical input, ISO 12207 [115], PMBOK [116] and ACM [139] have been used to check the contents. There have been examples where ISO PAs have been fitting better than the ones that have been taken from CMMI. The brighter PAs in SODE have been taken from ISO 12207 and have replaced or added PAs to the structure of SODE. For example, *PA 4.4 Acquisition Process* has been chosen over Supplier Agreement Management from CMMI as the ISO PA is easier to interpret in a GSE way. Supplier Agreement Management focuses on Suppliers only whereas Acquisition Process gives more margin for interpretation.

The ACM [139] and PMBOK [116] reference did not add new PAs to SODE but was used to thoroughly check the structure of SODE for any white spots. This was a very good way to check back on the PAs integrated into the model and map them to two different ways of structuring PAs. It was used to refine the interpretation and description of the structure but did not integrate changes in the structure itself.

2.3.3 STEP 3: Revision based on a systematic literature review

Based on the theoretical structure of SODE after these two steps, a systematic literature review (SLR) (see Chapter 3) has been used to refine the structure with practical input. In the SLR, solutions for global software engineering have been searched. These solutions have then been integrated into the structure of SODE. By doing this, white spots within SODE have been identified. For example, *PA 2.5 Distributed Team Management* has been added to SODE as there was no really fitting PA to integrate some of the solutions into. The solutions not fitting into SODE mostly described softer problems in coordination or communication issues or building awareness within a distributed team. This was not easily mappable to the SODE structure and therefore a new PA had to be created.

The solutions along with their rating can be found in the appendix.

2.3.4 STEP 4: Addition of a solution layer

Having mapped the solutions found in the SLR into SODE, there was a big gap observed between the theoretical structure of the model and the practical focus of the solutions. Therefore, a connecting layer has been designed to connect the solutions to the model. This layer gives an easy understandable name and description to every solution. Several publications found in the SLR can form one solution together or one publication can contain several solu-

tions. The solutions and the link to the model can be found in the appendix[2]. These links have been created for every part of SODE that contains publications found in the SLR.

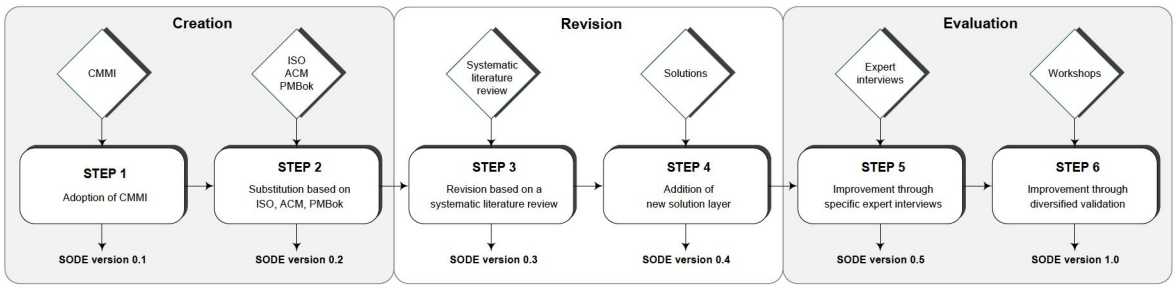

Figure 3: Creation Steps of SODE

2.4 SODE 0.2

The first version of SODE acted as an initial structure for mapping the solutions. The initial version of SODE was constructed using several sources, CMMI [120], ISO 12207 [115] and 26262, but also with input from Daimler AG. As a second step, SODE was extended and changed, in an iterative way, as the solutions themselves, caught in the SLR, evolved SODE as described in Chapter 3.

SODE consists of several Process Areas (PAs). Each PA contains one or more child PAs, thus indirectly refining the parent PA. Each of the child PAs might, additionally, contain one or more PAs, if further refinement is needed. The structure can be seen in Figure 4. Three simple rules govern the creation and subsequent refinement of SODE:

Rule 1. PAs may be refined by another layer of PAs.

Rule 2. The hierarchy below two different PAs may differ in depth.

Rule 3. Every child PA may only belong to one parent PA.

Rule 4. When splitting a PA to two or more child PAs. The original PA must have a ranking above average. A structural change to SODE should always be based on well-ranked PAs. (The ranking will be covered in detail in Section 3.3.3)

As SODE was designed to match the GSE context within Daimler AG, there might be some PAs missing that might be relevant for other domains. For example, as the automotive industry mainly produces embedded software, there might be differences to e.g. non-embedded domains.

To exemplify the generic structure an example has been incorporated into Figure 4. The example has been taken from the PA of Engineering and shows a PA structure containing Requirements Engineering as a first level child PA. This is refined by the *PA Elicitation*, which is

[2] Appendix E: List of SLR Papers

then refined into Workshops and Interviews, which are end leaves and considered as explicit solutions.

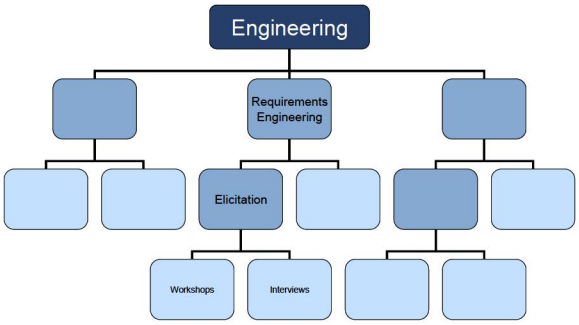

Figure 4: Generic structure of SODE with example

The idea behind SODE is to offer a practitioner several choices, that is, several solutions can, and preferably should, be offered. It is up to the practitioner and the individual needs, dictated by the domain and context, which of the solutions to choose.

As a result, SODE 0.2 is populated with PAs relevant to Daimler AG. The model structure enables Daimler AG to efficiently store and use the solutions as they can easily be found within the model. SODE 0.2, as a skeleton structure, was created based on several standards in software engineering, such as CMMI and the ISO 12207.

Looking at the structure, SODE 0.2 consists of four top level PAs, each containing several child PAs. The structure of SODE 0.2 can be seen in Figure 5. As Figure 5 shows, all top level Process Areas were taken from CMMI.

Each top level PA will be explained in more detail in the following subsections. All the details of SODE can be found in the appendix[3].

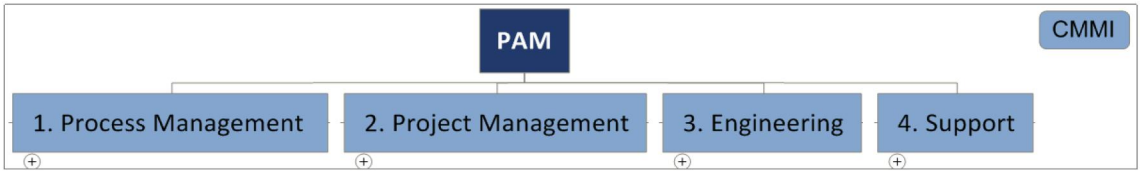

Figure 5: SODE 0.2 top level structure

2.4.1 PA 1 Process Management

The *PA 1 Process Management* contains the cross-project activities related to defining, planning, deploying, implementing, monitoring, controlling, appraising, measuring, and improving processes. The structure of SODE within Process Management can be seen in Figure 6.

[3] Appendix C: SODE 1.0

PA 1 Process Management has three child PAs, *PA 1.1 Organizational Training, PA 1.2 Organizational Process Definition* and *PA 1.3 Organizational Innovation and Deployment*.

PA 1.1 Organizational Training addresses the need to train employees to be able to work in a globally-distributed project. Therefore, *Organizational Training* is divided into *PA 1.1.1 Establish an Organizational Training Capability* and *PA 1.1.2 Provide Necessary Training*.

The purpose of *PA 1.2 Organizational Process Definition* is to establish and maintain the organizational process assets. This PA also covers the rules and guidelines for integrated teams. It is divided into *PA 1.2.1 Establish Organizational Process Assets* and *PA 1.2.2 Enable IPPD[1] Management*.

The purpose of *PA 1.3 Organizational Innovation and Deployment* is to select and deploy improvements that measurably improve the organizations processes. It is divided into *PA 1.3.1 Select Improvements* and *PA 1.3.2 Deploy Improvements*.

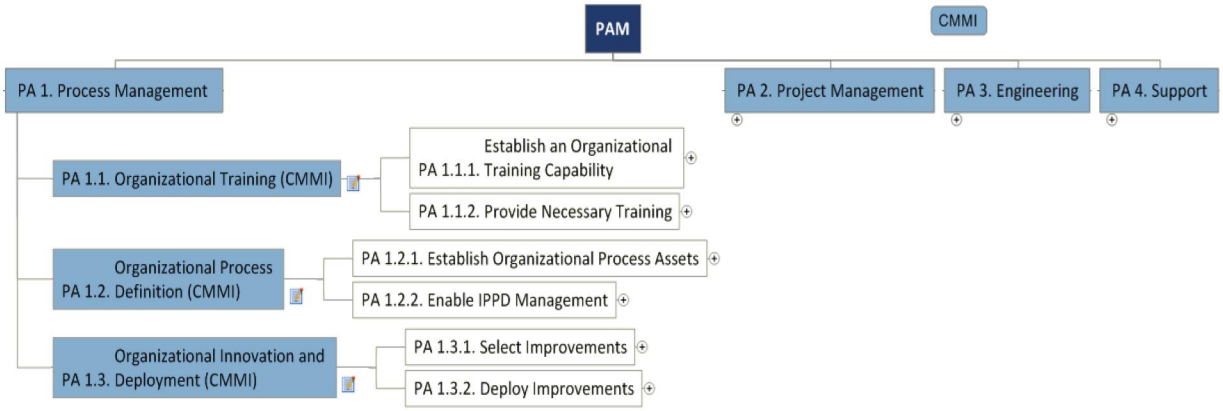

Figure 6: SODE: Process Management

2.4.2 PA 2 Project Management

PAs belonging to the *PA 2 Project Management* cover the project management activities related to planning, monitoring, and controlling the project. The structure of SODE within *PA 2 Project Management* can be seen in Figure 6.

PA 2 Project Management is subdivided into the four PAs, *PA 2.1 Project Planning*, *PA 2.2 Project Monitoring and Control*, *PA 2.3. Management Process* and *PA 2.4 Documentation Process*.

PA 2.1 Project Planning is supposed to establish and maintain plans that contain project activities. It is divided into *PA 2.1.1 Establish Estimates, PA 2.1.2 Develop a Project Plan* and *PA 2.1.3 Obtain Commitment to the Plan*.

The purpose of *PA 2.2 Project Monitoring and Control* is to give an insight into the project's progress and to identify corrective actions if the project is not on the right path. It is divided into *PA 2.2.1 Monitor Project against Plan* and *PA 2.2.2 Manage Corrective Action to Closure*.

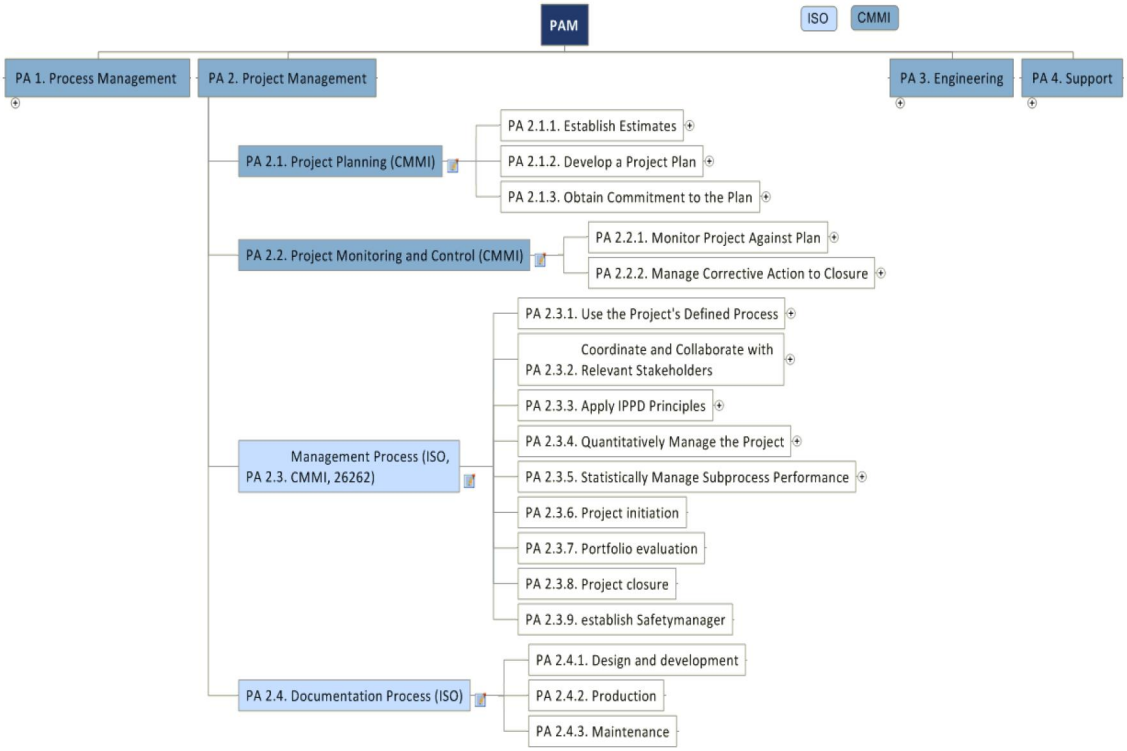

Figure 7: Project Management

The purpose of *PA 2.3 Management Process* is to establish and manage the project and also integrate the relevant stakeholders. It also covers a shared vision for the project and the establishment of integrated teams. It is divided into *PA 2.3.1 Use the Project's Defined Process*, *PA 2.3.2 Coordinate and Collaborate with Relevant Stakeholders*, *PA 2.3.3 Apply IPPD Principles*, *PA 2.3.4 Quantitatively Manage the Project, PA 2.3.5 Statistically Manage Subprocess Performance, PA 2.3.6 Project initiation, PA 2.3.7 Project evaluation, PA 2.3.8 Project closure* and *PA 2.3.9 Establish Safetymanager according to ISO 26262*.

The purpose of *PA 2.4 Documentation Process* is to record the information that is produced by the life cycle process. It is divided into the three phases *PA 2.4.1 Design and Development*, *PA 2.4.2 Production* and *PA 2.4.3 Maintenance*.

2.4.3 PA 3 Engineering

The PAs belonging to *PA 3 Engineering* cover the development and maintenance activities that are shared across engineering disciplines. The structure of *PA 3 Engineering* within SODE can be seen in Figure 8.

PA 3 Engineering consists of the four PAs which are *PA 3.1 Requirements Engineering/Management*, *PA 3.2 Technical Solution*, *PA 3.3 Product Integration* and *PA 3.4 Validation*.

PA 3.1 Requirements Engineering/Management has been combined as otherwise there would have been a Requirements Management PA with only one sub area. The purpose is to produce, analyze and manage customer requirements. It is divided into *PA 3.1.1 Develop Customer Requirements, PA 3.1.2 Develop Product Requirements, PA 3.1.3 Analyze and Validate Requirements* and *Manage Requirements.*

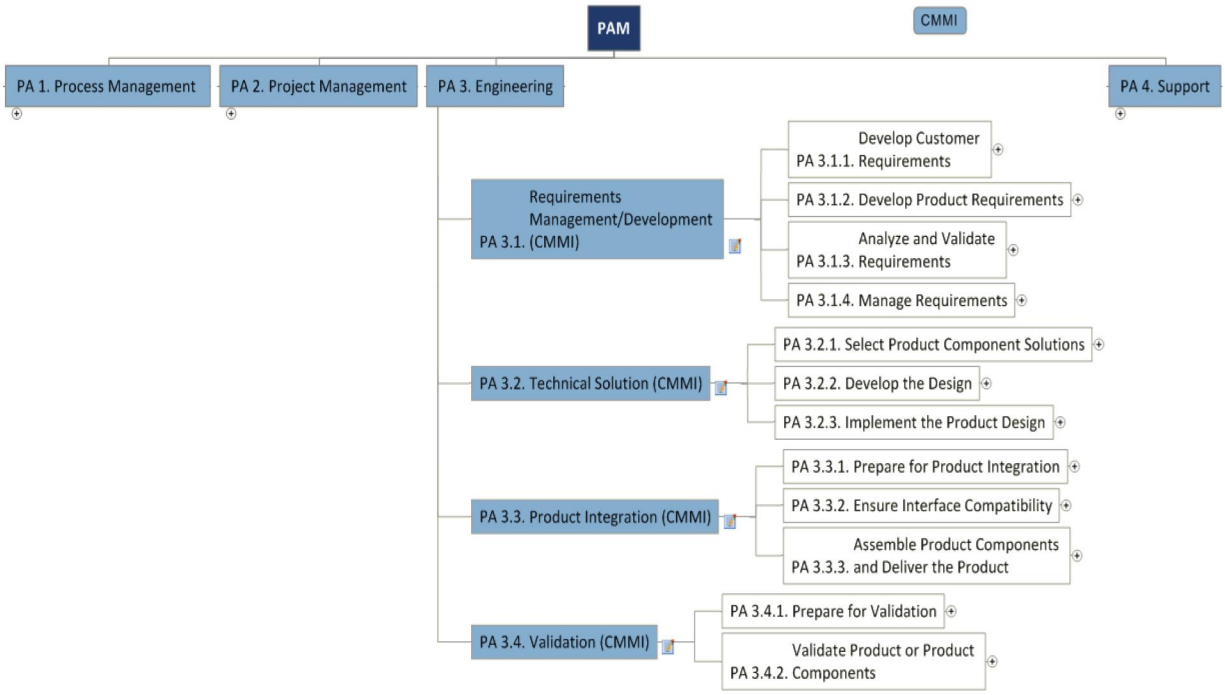

Figure 8: Engineering

The purpose of *PA 3.2 Technical Solution* is to design, develop and implement solutions according to requirements. It is divided into *PA 3.2.1 Select Product Component Solutions, PA 3.2.2 Develop the Design* and *PA 3.2.3 Implement the Product Design.*

The purpose of *PA 3.3 Product Integration* is to assemble the product. It is divided into *PA 3.3.1 Prepare for Product Integration, PA 3.3.2 Ensure Interface Compatibility* and *PA 3.3.3 Assemble Product Components and Deliver the Product.*

The purpose of *PA 3.4 Validation* is to demonstrate that the product fulfils the requirements. It is divided into *PA 3.4.1 Prepare for Validation* and *PA 3.4.2 Validate Product or Product Components*.

2.4.4 PA 4 Support

The PAs belonging to *PA 4 Support* cover the activities that support product development and maintenance. The structure of *PA 4 Support* within SODE is illustrated in Figure 9.

PA 4 Support consists of six PAs, *PA 4.1 Causal Analysis and Resolution, PA 4.2 Decision Analysis and Resolution, PA 4.3 Process Quality and Assurance, PA 4.4 Acquisition Process, PA 4.5 Maintenance Process* and *PA 4.6 Joint Review Process.*

The purpose of *PA 4.1 Causal Analysis and Resolution* is to identify causes of defects and other problems. It is divided into *PA 4.1.1 Determine Causes of Defects* and *PA 4.1.2 Address Causes of Defects*.

The purpose of *PA 4.2 Decision Analysis and Resolution* is to analyze decisions and the identification of possible alternatives.

PA 4.3 Process and Product Quality Assurance provides staff and management with objective insight into work products and the processes used to build these products. It is divided into *PA 4.3.1 Objectively Evaluate Processes and Work Products* and *PA 4.3.2 Provide Objective Insight*.

PA 4.4 Acquisition Process defines activities to support the acquirer in acquiring software products or services. It is divided into *PA 4.4.1 Acquisition preparation, PA 4.4.2 Supplier selection, PA 4.4.3 Supplier monitoring, PA 4.4.4 Customer acceptance* and *PA 4.4.5 Safety specific* regulations according to ISO 26262 if needed.

The *PA 4.5 Maintenance Process* is activated whenever the product undergoes changes or modifications. It is divided into *PA 4.5.1 Problem and modification analysis, PA 4.5.2 Modification implementation, PA 4.5.3 Maintenance review/acceptance* and *PA 4.5.4 Migration*.

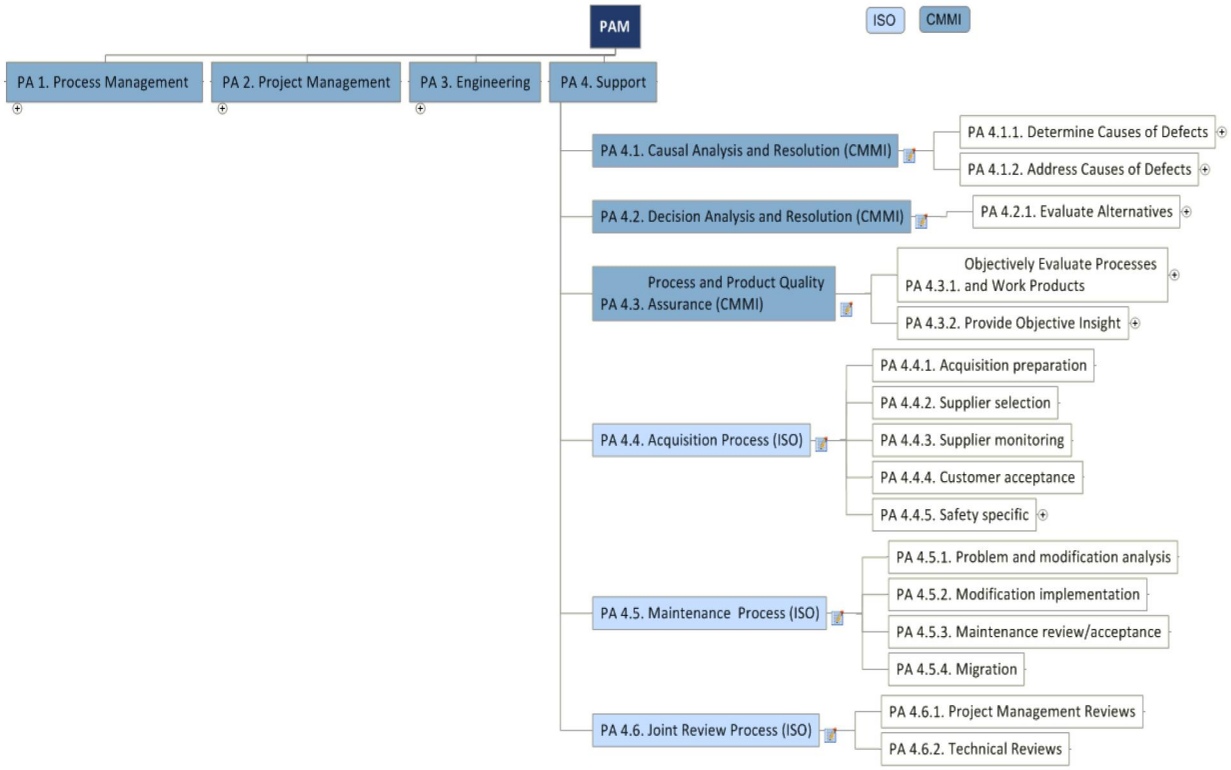

Figure 9: Support

The *PA 4.6 Joint Review Process* provides a framework for interactions between the reviewer and the reviewee. It is divided into *PA 4.6.1 Project Management Reviews* and *PA 4.6.2 Technical Reviews*.

2.4.5 Conclusion

This is the initial version of SODE, 0.2. As can be seen, the structure and most of the PAs have been taken from CMMI and combined with other standards like ISO 12207. This is the ideal initial setup as these are all well-known and widely-used standards and therefore enable easy adaption within the business context.

However, this is only the start for SODE. At this stage, SODE 0.2 needs further improvement and input to provide real added value for practice and academia. As can be seen in Figure 3, the first two steps are completed now. The next step to complete SODE is the revision of the results gathered in this chapter as can be seen in Figure 3. Chapter 3 will provide the actions that have been taken to perform the revision of SODE 0.2 and to develop it to SODE 0.4. This has to be done to add relevant input to SODE and to make it relevant for practice and academia. By now, SODE 0.2 can be seen as the skeleton, Chapter 2 will now add the contents that will be the main difference of SODE compared to other standards. This will then result in SODE 0.4 at the end of Chapter 3.

3 Solutions in Global Software Engineering: A Systematic Literature Review

3.1 Introduction

As SODE 0.2 has been created based on theoretical frameworks and models in Chapter 2, this chapter focuses on the further enhancement of SODE. This means that SODE 0.2 will be used as an input for this chapter and will be adjusted and developed to become more relevant for industry and academia.

As Challenge 27 and 28 (see Section 1.1.1) have shown, there is a need for guidance and solutions in GSE projects. Taking the structure provided by SODE 0.2, solutions that help practitioners to do their daily work was missing. Several systematic literature reviews have been published focusing on everything from applying agile in a GSE environment (see e.g. [140]), to subcontractor management issues [123], while some studies are wider and on a macro level look at rigor and level of empirical validation of solutions in GSE [103].

For industry, problem descriptions and identifying challenges is relevant, but so is offering information on possible solutions to tackle the complexities. This chapter presents a systematic literature review (SLR) [122] focusing on identifying and collecting as many solutions for GSE as possible by extracting them from peer-reviewed literature. Solution in this context means any process, tool, technique, model and so on, that claims to tackle specific GSE challenges. As the solutions found are extracted, they are put in SODE, which structures the solutions into Main- and Sub-process areas. Each of the solutions is further evaluated against different criteria, for example, level of trust a potential industry practitioner can accredit a specific solution based on the level of validation the solution has undergone.

This chapter thus presents the results of an extensive systematic review, extracting solutions in a GSE context, gauging their level of validation, and mapping them into a structure creating SODE 0.4. The use of SODE, populated by the extracted solutions, can be seen from two main perspectives. First, researchers can use SODE 0.4 to build upon, and to see which process areas are well researched, and which areas require extensive study in the future. Second, industry practitioners can use SODE 0.4 as a reference framework to evaluate and improve their development environment by finding solutions in a structured way.

The chapter is structured as follows. In Section 3.2, related work is presented along with the context of this study at Daimler AG, the main partner in the study presented in this chapter.

In Section 3.3, the design of this study is shown. In Section 3.4, results, and in Section 3.5 a solution characterization is offered. Section 3.6 concludes this chapter.

3.2 Related Work and Background

This section discusses related work, elaborates on the industrial collaborating partner and the work context in relation to GSE. Background and motivation to the study and the creation of SODE is also described.

3.2.1 Related Work

Several systematic literature reviews (SLR) have been performed previously in the area of GSE.

Babar et al. [140] studied the use of SCRUM in GSE, as agile practices are extremely popular in GSE projects. They identified about 400 papers and extracted results from about 20 of them. The main conclusion is that it is hard to provide solutions for GSE problems as the type of development distribution differs from project to project. Yet, there were some findings that seemed to help in many projects, e.g. the importance of synchronous communication.

Khan et al. [123] focused on the outsourcing part of GSE and studied success factors for e.g. supplier selection. This SLR also states some similarities and differences between cultures of offshore outsourcing.

A more general study of GSE can be found in the SLR of Šmite et al. [103]. Focusing on empirical studies in GSE, this SLR gives deep insight into the field of GSE providing useful practices and techniques. For practitioners, the SLR provides the seven most frequently discussed practices.

The SLR by Huda et al. [141] focuses on finding the key barriers for global software projects. As many problems arise during the transition from local to global organizations, the authors identify challenges to guide companies to implement appropriate organizations for GSE.

The risks of GSE in requirements engineering have been studied by Lopez et al. [142]. The authors identified risks regarding the requirements process especially with inexperienced teams. These risks are related to the shift towards globally-distributed development. The SLR also provides some safeguards that help overcome these risks; however, the safeguards have not been tested in practice.

In general, it can be seen, that SLRs in relation to GSE often focus on specific aspects of GSE, or general challenges and problems associated with GSE. So far, as as can be told from the research, no SLR has been conducted focusing on the investigation of *solutions* associated

with GSE. This work focuses on the solution perspective, mining peer-reviewed research for solutions used to address challenges. Further, the level of industrial validation and use of these solutions are also evaluated.

3.2.2 Study Context

The work presented in this chapter, the solution-focused literature review, and the creation of SODE was inspired by a gap in literature, but also directly by the needs of Daimler AG, which is the main research partner of this project. Daimler faces many challenges related to GSE. This section gives a brief motivation to the work and the context characterization of Daimler AG.

More details on Daimler AG and the context of this research within Daimler AG can be found in Section 1.1.3.

3.2.2.1 SOlution framework for Distributed Engineering (SODE)

As a response to the GSE challenges at Daimler AG, a research project consisting of collaboration and co-production between industry and academia initiated the work of creating the process area map (SODE) aimed at structuring available solutions to tackle global product development. SODE can be seen as a framework with process areas sorted in a tree structure, e.g. Requirements Engineering (a main process area) has leaves like Elicitation (sub-process area), and at the end node existing solutions are mapped into SODE. In this example, workshops or interviews can be used for actual elicitation and is thus seen as a solution (end nodes in SODE).

SODE has two purposes. First, SODE and the SLR provide an overview of which fields in GSE are most researched, where the gaps are, and what it avenues for future research. Second, SODE can be used by Daimler AG as a reference model for finding solutions to GSE-based challenges based on process area, and as new solutions are offered they can be mapped and structured into SODE.

SODE has been introduced in more detail in Section 2.

3.3 Study Design

In Section 3.3.1, a number of research questions (RQs) are presented. SODE has already been introduced in Section 2.3, as the structure of SODE is a part of the research formulation. The SLR will be introduced in Subsection 3.3.2.

3.3.1 Research Questions

The RQs that drive this study are specified and explained in Table 7. The overall idea is to perform a SLR to identify as many "solutions" in GSE research as possible, then map these solutions to SODE. SODE in turn is based on an empirical study conducted at Daimler AG, and the consolidation of several process models. It should be observed that the main purpose of this chapter is not to present SODE itself, but rather use it as a structure to sort and categorize the solutions found during the SLR.

	Research Question	Description
RQ1	**What are the main Process Areas identified at Daimler AG in relation to GSE?**	SODE is designed based on state-of-the-art process models and also in addition extended and populated based on the needs of Daimler AG. By being able to map the needs of Daimler to the structure of SODE an indirect validation of the SODE structure is achieved.
RQ2	**Studying state-of-the-art, to what level is it possible to map solutions to SODE?**	There is a need to identify how good the solutions found in state-of-the-art match into the structure of SODE, but also indirectly to the needs of Daimler AG. This RQ is divided into two subquestions.
RQ2.1	What parts of SODE do not have solutions associated to it?	The identification of blank spots within SODE is of great interest as this would mean that there are areas which do not yet have solutions, and thus are areas of future research.
RQ2.2	Is there a need to create new Process Areas in SODE based on the solutions found in state-of-the-art?	Are there process areas (PAs) missing in SODE? That is, are there solutions in state-of-the-art that fall outside the structure offered by SODE. If so, how can SODE be extended? This is also an indirect validation of SODE.
RQ3	**To what extent have the solutions in state-of-the-art been described and tested?**	The solutions found need to be rated for quality. This is as it is important to identify the right solution fast and accurately in an industrial environment. The ratings set up for this purpose are described in the sub-questions in this section.

RQ3.1	How well is the context of the solutions' use and validation described?	To what extent is the case context of the solutions' validation described? This is important to decide whether a solution can be applied to another context.
RQ3.2	How well is the research design of the solutions described?	The description of the design of a study presented in a paper is of fundamental importance to the reader in order to judge whether the results presented are reasonable and reproducible.
RQ3.3	How well validated are the solutions?	The validation of the solutions presented in the paper is also very important to the reader. Very well validated solutions might be implemented right away whereas unvalidated solutions can require additional work to assure e.g. scalability in a real industry environment.

Table 7: Research Questions for the SLR

3.3.2 Systematic Literature Review (SLR)

State-of-the-art had to be studied to fill SODE. An SLR was designed and executed for this purpose.

Figure 10 provides an overview of the SLR design, which consists of three major steps. First, the research question was formulated and the need for the SLR was examined. Second, the SLR needed to be planned. That is, developing search terms, pilot searches, test extraction of data, and subsequent refinement of all steps. Third, the SLR inclusion, exclusion and extraction rules and procedures were established and refined.

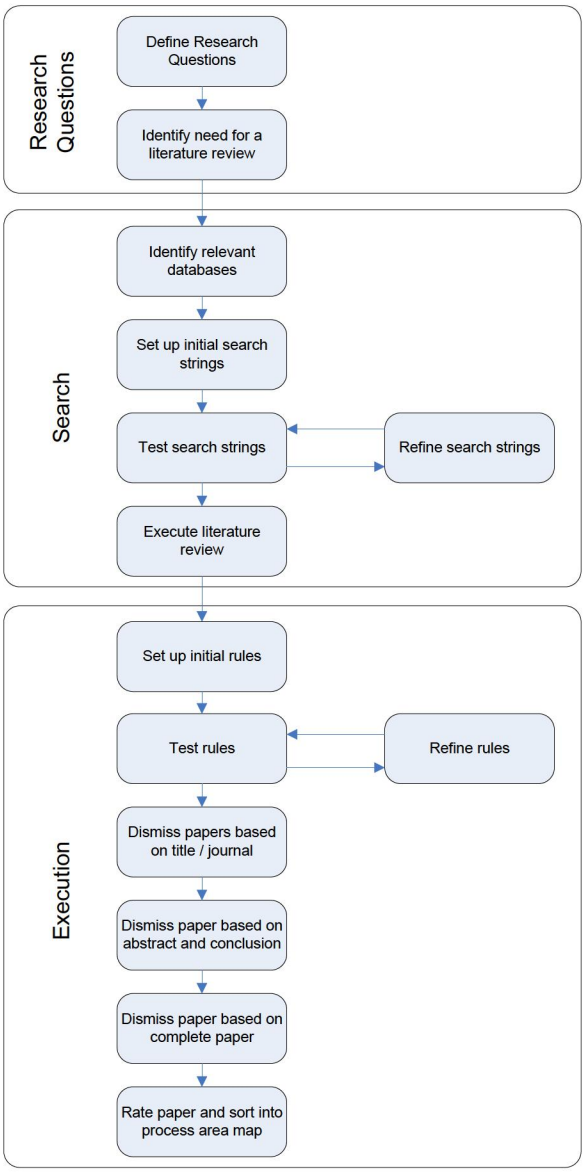

Figure 10: systematic review process

3.3.2.1 Search

The following five databases were selected as primary targets. Scopus, INSPEC, IEEE Xplore, ACM Digital Library and SpringerLink.

The search string was developed through a cycle of test and refinement. That is, the search string was tested, modified, tested again and the results were compared over iterations until adequate results were obtained. The search results were screened for solutions that could be relevant in a GSE context. (This does not mean that solutions relevant for non-GSE context were dismissed.)

The stop criterion for search string refinement was when at least 10% of the top 20 results in a search were relevant for the SLR. This procedure was replicated for all databases using the

search strings seen in Table 8. The search string with the best results was composed of two groups of search terms. The first group is related to the GSE part of the problem:

> Virtual teams, Distributed development, Offshoring, Outsourcing, GSE/Global Software Engineering

The second group of search terms consists of keywords with the purpose of finding particular solutions in GSE:

> Solution, Best practice, Lessons learned, Process improvement, Process enhancement, Process innovation, SPI

The search string combinations, as well as the number of results, can be seen in Table 8. In total, 2,861 papers were found in the search. The search covered all papers in the databases until March 2009.

Database	Search string	Results
IEEE eXplore	((((solution or best practice or lessons learned or 'process improvement' or 'process enhancement' or 'process innovation' or 'spi'))<in>ab) <and> (((virtual team or distributed development or offshor* or outsourc* or gse or global software engineering))<in>ab))	260
INSPEC	(solution OR “best practice” OR “lessons learned” OR “process improvement” OR “process enhancement” OR “process innovation” OR SPI) AND (“virtual team” OR “distributed development” OR offshor* OR outsourc* OR GSE OR “global software engineering”)	931
SCOPUS	TITLE-ABS-KEY((solution OR "best practice" OR "lessons learned" OR "process improvement" OR "process enhancement" OR "process innovation" OR spi OR support) AND (“virtual team” OR “distributed development” OR offshor* OR outsourc* OR gse OR “global software engineering”))	167
ACM Digital Library	(gse or "global software engineering" or "virtual team" or "distributed development")	619

Springer Link	(gse or "global software engineering" or "virtual team" or "distributed development") and (solution or "lessons learned" or "best practice" or spi or "process improvement")	884

Table 8: Search strings

3.3.2.2 Data Extraction

After dismissal of duplicates using Citavi[4], rules for the inclusion and exclusion of papers were set up. These criteria can be seen in column one of Table 9. The criteria are exemplified in column two mostly as questions that need to be answered. If a paper did not meet all criteria it was excluded. In the last column the data used to make this decision is given.

The inclusion/exclusion criteria were further tested to ensure the reproducibility of the SLR, by selecting 15 random papers. Three reviewers took part in the test. The test was performed on a sample set of the results of the search, containing papers from every database in approximately the share that each database had in the complete result. For example, INSPEC provided about one third of the total papers included in the SLR, so INSPEC also provided a third of the papers used for the sample set.

The test was separated into four steps:

Step 1. Set up the criteria and a sample set.

Step 2. Perform a classification on the sample set.

Step 3. Compare the results of the classification.

Step 4. If agreement level not met; refine the criteria and go to Step 1, else exit.

This test was then conducted until the agreement level between the three reviewers was not more than one paper out of all in the sample.

Criteria	Description	Information used
Area of interest	Does the paper belong to the area of interest studied in this SLR?	**Title, Journal/Conference Information, Authors**

[4] www.citavi.com

GSE Context	Does the paper belong to GSE context?	**Abstract, Conclusion**
	Does the Conclusion promise a solution in the paper?	
Solution	Is there a solution presented in the paper?	**Whole paper**
	Is there a technology in the paper that can be used as a solution in GSE?	
	Is there any input in the paper that can help software engineers to do their daily work in a GSE environment?	
Rating	Is the solution presented in the paper validated and is the validation itself presented in the paper?	
	Is the context of the solution described in the paper? That is, does the paper give an insight into the project and organization that developed this solution?	
	Is the design of the study that led to this solution described in the paper? That is, is it easy to understand how that solution has been developed?	
Populate	Sort the rated solutions into SODE.	

Table 9: Data Extraction Form

3.3.2.3 Rating and Transformation

The rating part presented in Table 9 used a grading scheme (scale) were the value 0 meant that there was no recognizable effort described in the area. That means, that there was no way to tell whether the method had been validated or not, in which context it had been developed or how the study had been designed. The scale from one to three is explained separately for each dimension below. The perspectives, as well as the scales were developed by Ivarsson and Gorschek [143].

Validation, being the most important measure for industry relevant solutions [103], [121], [143], [144], is ranked according to a scale from zero to three as can be seen in Table 10. In

this scale, there is a strong focus on validation in industry. Therefore, a validation that is done by a researcher in academia is rated with 1, whereas a validation done by a practitioner in industry can be said to be the most realistic setting for a validation, thus considered to be highly relevant for industry, is ranked with 3. Mixtures of both are ranked 2 as this is not necessarily validated in industry. It should be observed that this rating is primarily adapted to grade the level of industrial validation. Certain solutions might be validated in academic settings, and be *potentially* valid in an industry setting. These types of studies were not the main focus of this research, but industrial usability and usefulness.

Done by / Done in	researchers in academia	researchers in industry
academia	1	2
industry	2	3

Table 10: Scale for Validation ratings

The scales for Design and Context, however, are described on a linear scale with more general characterizations, as can be seen in Table 11 (c.f). To receive a good rating, information on the context in which the evaluation is performed (e.g. experience of the staff, development process) is needed. It should be observed that the rating is rather forgiving, i.e. providing some contextual information gives a rating of two (2) on a scale from zero to three.

Scale	Description Context
0	There is no description of the context at all.
1	The context is described shortly or the industry is mentioned (e.g. automotive). This gives an idea of the context but leaves many questions unanswered like typical project type, development method, size, length, etc.
2	The reader can find information on the context of the solution but questions remain unanswered.
3	It is easy for the reader to understand the context of the solution. This includes e.g. the type of distribution (e.g. temporal, geographical, cultural) or the type of organizational distribution (e.g. number of suppliers, work distribution between the suppliers).

Table 11: Scale for Context ratings

The scale for Design is quite similar, as can be seen in Table 12. To receive a good rating, the products, resources and the process used in the evaluation should be explained. However, here also, the rating is forgiving, and, hence, providing a minimum level of design description would result in a two (2).

Scale	Description Design
0	There is no description of the Design at all.
1	There is a short description of the study design (e.g. the steps taken in the study), but many questions remain unanswered.
2	The study design is described (e.g. the steps taken in the study) but some questions remain unanswered.
3	The reader can understand the study because they are given the variables measured, the control used, the treatments etc.

Table 12: Scale for Design ratings

The analysis of the results from the SLR demanded that papers and their characteristics in terms of the rating above could be grouped and compared. To enable such comparisons, e.g. papers grouped in different PAs, another scale had to be set up. The rating with the rules set up in Table 10, Table 11, and Table 12 is on an ordinal scale and therefore indicates an order but cannot be used to calculate, for example, averages. Thus, for example, it is unclear if a paper rated 2 is twice as good as a paper rated 1. Also, it is not yet obvious whether two papers rated 1 and 3 are on average as good as two papers where each is rated 2 [145], [146].

The idea was to transform the ordinal scale into a ratio scale. One way of doing so is to assign ranks to the ratings. That is, the papers in each of the rated areas (Validation, Context and Design) are ordered. Then, the rank for a specific rating is obtained by counting the papers rated below the specific rating. Table 13 shows an example of papers rated according to our three areas. The paper IDs are given in column one, the ratings are given in columns two to four.

	Context	Design	Validation
P1	0	1	3
P2	1	3	3
P3	3	0	1
P4	2	2	0
P5	0	0	2
P6	3	3	0
P7	3	2	0
P8	1	1	0
P9	2	1	2

Table 13: Original ratings

	Context		Design		Validation	
Original rating	Number of papers with this rating	Number of papers rated worse = rank	Number of papers with this rating	Number of papers rated worse = rank	Number of papers with this rating	Number of papers rated worse = rank
0	2	0	2	0	4	0
1	2	2	3	2	1	4
2	2	4	2	5	2	5
3	3	6	2	7	2	7

Table 14: ratings and ranks

First, the frequencies of the ratings for each area in Table 13 are counted and presented in Table 14.

Table 14 provides two kinds of information. It provides the original rating in column one and matches it with the areas. Each area is split into two columns. The first column gives the frequencies of the original ratings in this area. For example, in Context, we have two papers rated 0 (P1 and P5 from Table 13), while in Validation we have four papers rated 0 and so on.

The second column in each area presents the corresponding ranks of the ratings. These ranks are now used instead of the original ratings. So, In Table 15 the same information as in Table 13 will be displayed but ranks instead of ratings are used. For example P1, with rating 0 in Context will be replaced with its rank 0, the rating 1 in Design will be replaced with its new rank 2 and the rating 3 in Validation will be replaced with the rank 7. Those are the underlined numbers in Table 14 and Table 15. Those numbers give the amount of papers rated below a specific rating and can be used to calculate averages and compare different process areas. These numbers also verify how many papers are in each area and which perceived quality they have. It is therefore favorable to see lots of high ranks as this indicates PAs with high quality papers.

	Context	Design	Validation
P1	0	2	7
P2	2	7	7
P3	6	0	4
P4	4	5	0
P5	0	0	5
P6	6	7	0
P7	6	5	0
P8	2	2	0
P9	4	2	5

Table 15: Ranks instead of ratings

It is also possible to calculate a discrete distribution function [146] for each group of papers based on the rankings. In this case, the distribution function gives the probability that a paper randomly picked from that group is ranked worse for a given rating:

$$P(X < \mu) = \frac{r_\mu}{n},$$

where X is a paper randomly picked, μ is the input rating, r_μ is the rank of μ and n is the total number of papers. Therefore, the probability is just the ranking transformed into percentages. The reason for presenting this transformation from values into ranks is to enable analysis and comparisons as will be seen later in Section 3.4.4. In short, it is immaterial that GSE is a fairly new research area and, thus, also is likely to hold a lower "overall" knowledge and maturity, since a comparison, statistically speaking, within the treatment, i.e. within the research area is done.

3.4 Results & Quantitative Analysis

This section presents statistical results for each of the research questions as stated in Section 3.3.1. Additional results, in terms of a characterization of the actual content of each area/PA are presented in Section 3.5.

The bases for the results are the papers found in the SLR. The amount of papers that have been kept in each step described in Table 9 can be found in Table 16. It can be seen, that in the end 330 papers have been kept to be incorporated into SODE, i.e. 11.5% of all papers that have been reviewed in the SLR. A complete list of papers included in the SLR can be found in the appendix[5].

[5] Appendix E: List of SLR Papers

Database	Results	Area of Interest	GSE Context	Rating and sorting step
IEEE Xplore	260	116	66	65
INSPEC	931	484	88	55
SCOPUS	167	104	29	14
ACM Digital Library	619	165	82	81
Springer Link	884	311	132	115
Total	**2,861**	**1,180**	**397**	**330**

Table 16: Number of primary studies

The solutions presented in the paper have also been sorted into solution groups. These groups cluster closely related solutions together and makes it easier for the reader to identify the right solution. There are 127 solutions in SODE.

In Table 17 a list of top ranked papers can be found. As can be seen, there were only three papers that received the highest rating in all the categories of Context, Design and Validation, i.e. only about 1% of all the primary studies from the SLR. These papers can be taken as a measurement that other papers have to live up to in this rating. However, it needs to be considered that this rating is context-specific with respect to Daimler AG. A detailed list of papers and their detailed ratings can be found in the appendix[6].

Paper	Context	Design	Validation
[147]	3	3	3
[148]	3	3	3
[63]	3	3	3

Table 17: Top ranked papers

6 Appendix D: SLR Ratings

In Table 18 the "best validated" papers from the SLR can be found. These are the papers that industry should be looking for as these are the papers that often contain relevant information for practitioners. The solutions introduced in these papers have proven to be successful in particular contexts and, hence, provide valuable insight into solutions that are already used by industry.

Paper	Context	Design	Validation
[18]	2	1	3
[149]	3	2	3
[147]	3	3	3
[150]	0	1	3
[39]	0	0	3
[151]	2	1	3
[152]	1	1	3
[153]	2	1	3
[154]	1	1	3
[155]	2	2	3
[156]	1	1	3
[148]	3	3	3
[157]	3	3	3
[158]	2	2	3
[6]	2	1	3
[70]	1	2	3
[52]	2	3	3
[159]	2	2	3

Table 18: Top ranked papers in Validation

Detailed results of the SLR can be found in the appendix[7], and are not presented here due to space limitations.

3.4.1 RQ1: Process Areas in SODE

For RQ1, SODE needs to be populated with PAs relevant to Daimler AG. The model structure enables Daimler AG to efficiently store and use the solutions as they can easily be found within the model.

SODE 0.2 was used to answer this RQ. In Chapter 1, SODE 0.2 is presented. It consists of several well-known and widely-used standards like CMMI and ISO and was designed to match the requirements of this step.

3.4.2 RQ2: Map solutions against SODE

In this section, the process how the solutions found in the systematic literature review are mapped to the process areas (PAs) in the process area map (SODE) is described. The purpose of this was to identify if there are parts of the SODE that had few or no solutions in current state-of-the-art (Section 3.4.4). Additionally, the SLR was then used to validate SODE by investigating if SODE needed to be extended based on the findings from the SLR (Section 3.4.3). A complete mapping of the solutions found in the SLR and SODE can be found in the appendix[8].

One goal, when creating a map, can be to find blank spots where nobody has been and then figure out if nobody has thought of the problem yet or if there are no significant problems to expect in this area. In Figure 11, a histogram of the Process Areas of SODE is shown. The total number of papers within one PA is displayed on the *x*-axis with the corresponding number of PAs on the *y*-axis, e.g. the first bar gives the amount of PAs that have only one or two papers. That is, the PAs are categorized into groups, where every group contains the same number of papers. The first two groups, containing one to four and five to ten papers, have been split in two sub-groups, as they were quite large. In Figure 11 it can be seen that there are 17 PAs that contain one or two papers only and 11 PAs that contain three or four papers. So, there are 28 PAs with less than five papers, which means that more than 68% of all PAs in the SODE have four or less papers. Having less than five papers, however, leaves little

---

[7] Appendix D: SLR Ratings, Appendix E: List of SLR Papers

[8] Appendix C: SODE 1.0

choice when selecting different solutions. For example, context-specific tailoring of these PAs would probably result in many empty areas. A larger variety concerning the offered solutions, i.e. surely more than five papers per PA is needed, would be preferred.

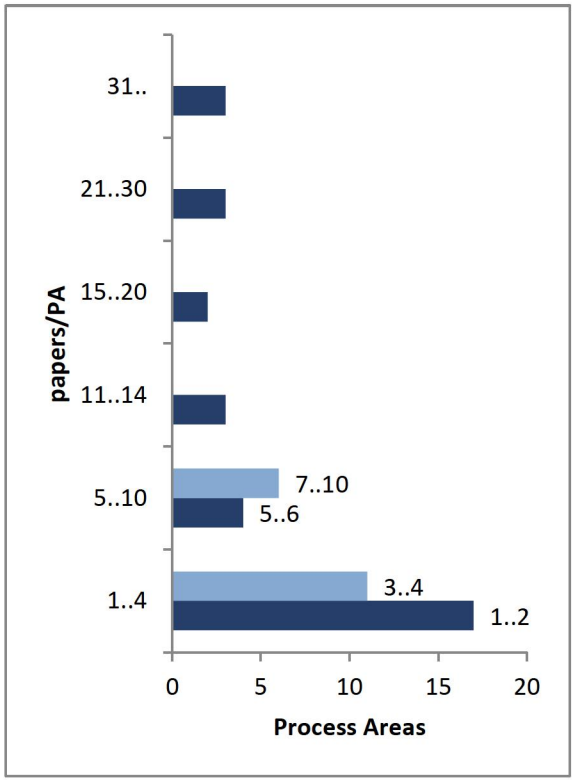

Figure 11: Histogram of all PAs

While the results shown in Figure 11 have been gathered from the leaves of the SODE, the results have also been assembled to the PA level. There are four top level PAs as shown in Figure 12. Three of these PAs seem to be fairly neglected (*PA 3 Engineering*, *PA 4 Support* and *PA 1 Process Management*) compared to *PA 2 Project Management* which is quite populous. The first three will be examined next while the last one will be examined in 3.4.3.

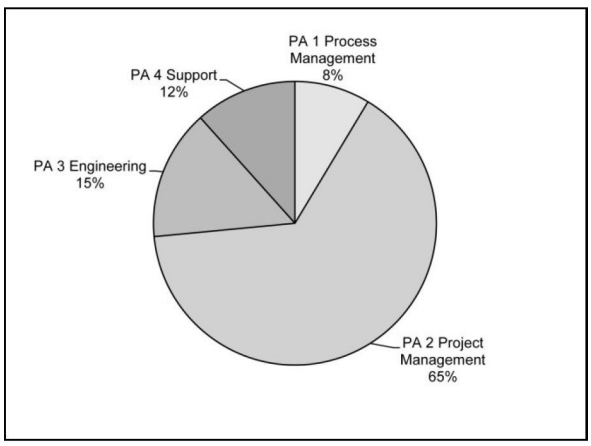

Figure 12: Overview over the PAs

In Figure 13, *PA 1 Process Management* is displayed in further detail. As is evident, there is one larger part (*PA 1.2 Organizational Process Definition*) and two parts containing fairly few papers.

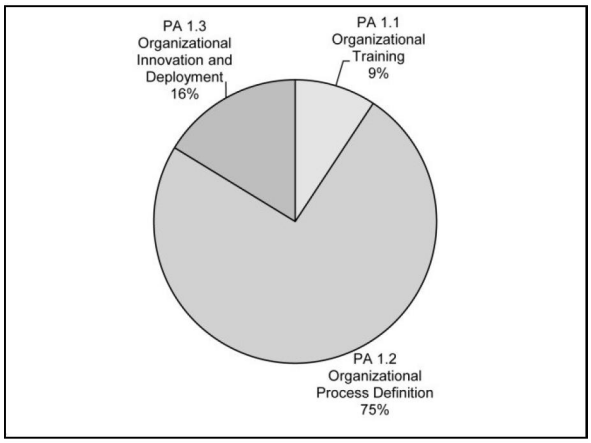

Figure 13: PA 1 Process Management

In Figure 14, *PA 3 Engineering* is presented in more detail. As can be seen, there is one larger part (*PA* 3.1 *Requirements Management/Development*). As *PA 3.1 Requirements Management/Development* is a very intense area of research within GSE, this is also one of the larger PAs. In Section 3.4.4 it will be shown that this PA is also of high quality concerning the maturity of proposed solutions. The other three parts (*PA 3.2 Technical Solution, PA 3.3 Product Integration* and *PA 3.4 Validation*) are quite small and might be in need of some additional attention.

In Figure 15, *PA 4 Support* is described in further detail. It can be seen, that there is one large part (*PA 4.4 Acquisition Process*), whereas the other parts are rather small. Especially *PA 4.1 Causal Analysis and Resolution*, as well as *PA 4.3 Process and Product Quality Assurance*, might be in need of some more attention.

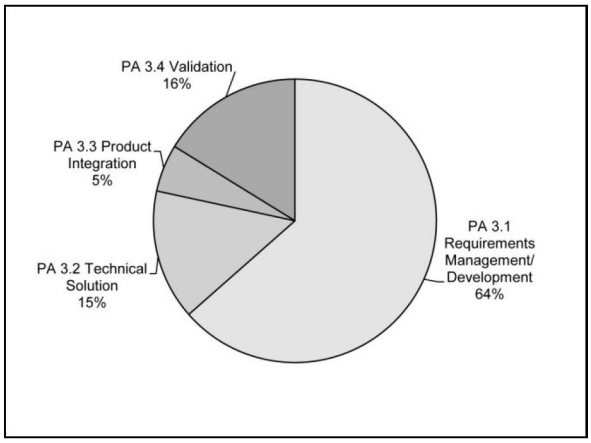

Figure 14: PA 3 Engineering

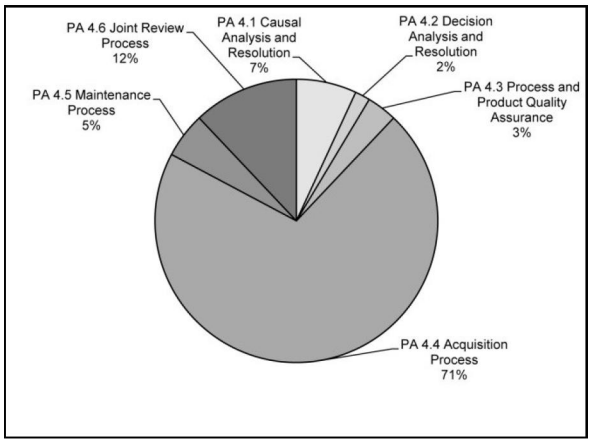

Figure 15: PA 4 Support

3.4.3 RQ2.2: New PAs

The second important question to be asked is if SODE is complete. Complete in this context means to provide solutions for Daimler AG within the GSE context. This is important to note as it may not be complete for other contexts and companies. However, looking at the distribution of papers in Table 17, it can be seen that some PAs contain a lot of papers. As in

, the total number of papers within a single PA is shown as ranges on the y-axis with the corresponding number of PAs on the x-axis. These crowded PAs are, by large, located in *PA 2 Project Management*. In Figure 16, it can be seen that there is especially one very populated area (*PA 2.5 Distributed Team and Project Management*). The other areas fade in comparison.

The next step now is to analyze if there is a PA missing or if a PA needs to be refined and, thus, the large amount of papers should be distributed across two, or more, new PAs.

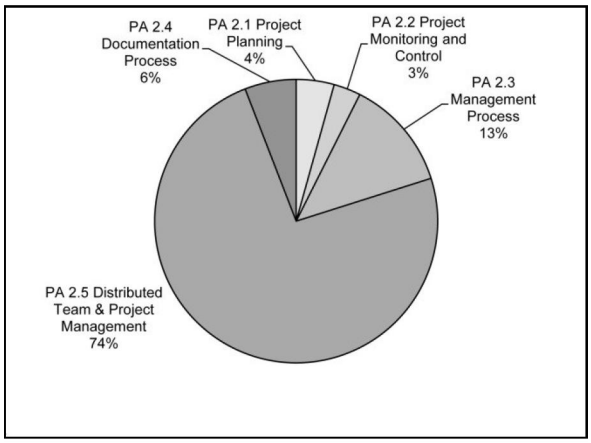

Figure 16: PA 2 Project Management

By analyzing the papers in the populated areas of SODE, it can be seen that there was an obvious need to refine *PA 2 Project Management*, by adding *PA 2.5 Distributed Team & Project Management* as can be seen in Figure 17. The PA in the lower part of the figure is newly added, combining papers with a clear connection to distributed team management as well as distributed project management, and consists of five PAs:

PA 2.5.1 Team Building/Management contains every paper that deals with how to set up a virtual team and how to manage it.

As *PA 2.5.2 Coordination/Communication* is considered to be two major problems in GSE; there was a need to create a special PA to summarize everything related to communication within a team and coordinating a virtual team into a specific PA.

Virtual teams are not necessarily motivated in the same way as traditional teams. Therefore, *PA 2.5.3 Motivation* contains everything related to motivational solutions.

If parts of a team are distributed all over the world, the team has to be aware that there are other people involved in their project beyond the team members they regularly meet. *PA 2.5.4 Awareness* contains solutions on how to accomplish that.

As there were many papers that summarized several rules on how to manage a distributed project, there was a need to collect these papers not only in the PAs they have solutions in but also in a PA that contains only "solution collections". *PA 2.5.5 Distributed Project Management* is this new PA.

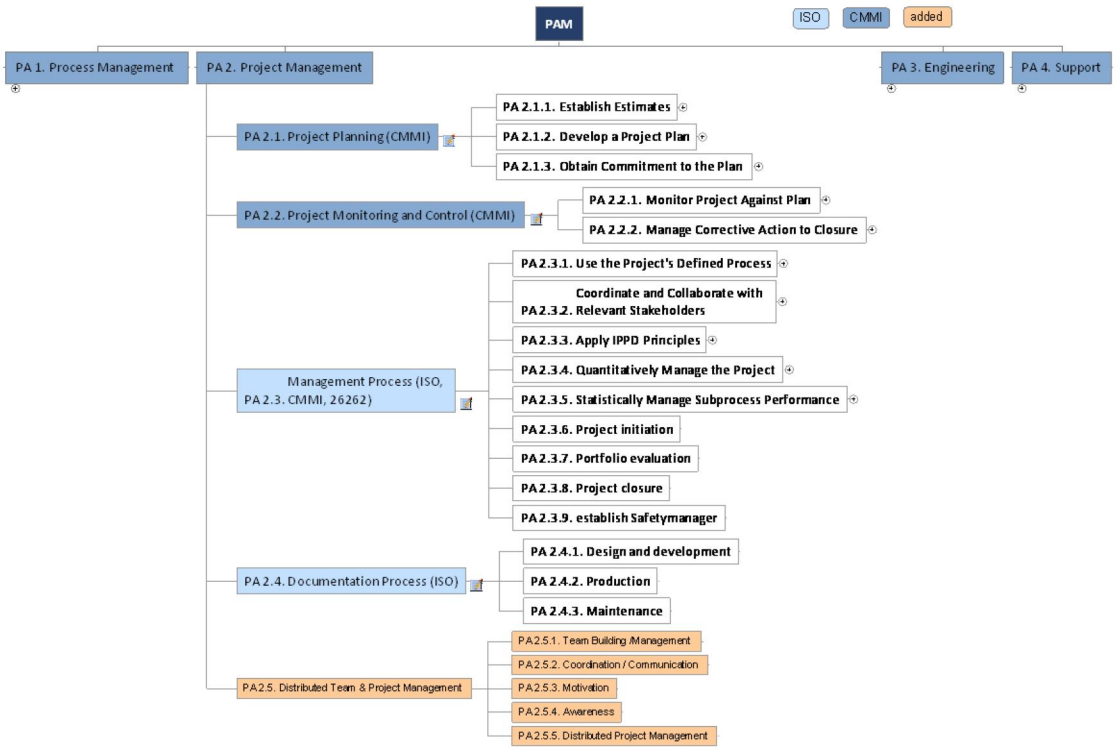

Figure 17: PAM with added PA

As these PAs are collecting a lot of papers, there might be a need to refine them in even more detail. As *PA 2.5 Distributed Team & Project Management* contains more than 100 papers in the new setting, it might also qualify as a new top level PA, especially as this might be considered the main focus of GSE.

At the moment, the empty areas are more interesting to analyze compared to the overly populated areas, but during an evaluation of the model at Daimler AG there might be some additional refinements to SODE. Table 19 shows that SODE consists of 56 PAs in total. This number provides some insight into the size of SODE. Compared to CMMI for example it is of course smaller as it focuses only on one aspect of software engineering, GSE, and on the needs of only one company, Daimler AG. It can also be seen, that there are 147 actions in SODE. Actions are the leaves in the model and these leaves contain any solutions that might exist. The average rankings of all papers (80.1) as well as the rankings for Context (80.7), Design (77.9) and Validation (81.7) can also be found in Table 19. This means, that the average paper in this area is "better" validated than 81.7 papers in the total set of papers within SODE as described in Section 3.3.2.3.

Number of PAs	56
Number of actions (PAs without children PAs)	147
Average rating of all papers	80.1
Average Context rating of all papers	80.7
Average Design rating of all papers	77.9
Average Validation rating of all papers	81.7

Table 19: Details about SODE

3.4.4 RQ3: Maturity of the Solutions

In RQ3, the average maturity of the solutions in a PA is used to describe the maturity of the PA. This average maturity can then be used to compare PAs. To demonstrate the results, the ratings for the selected PAs will now be shown. In Figure 18 and Figure 19 the rank of selected PAs is shown and the length of the bar indicates the rank (the higher the better). The

ranks indicate the number of papers rated below the average paper in this PA. The higher the bar, the more papers are left behind by an average paper in this PA. For each PA shown in these figures, three bars are grouped together indicating the rank for Context, Design and Validation. A list of all papers rated can be found in the appendix[9].

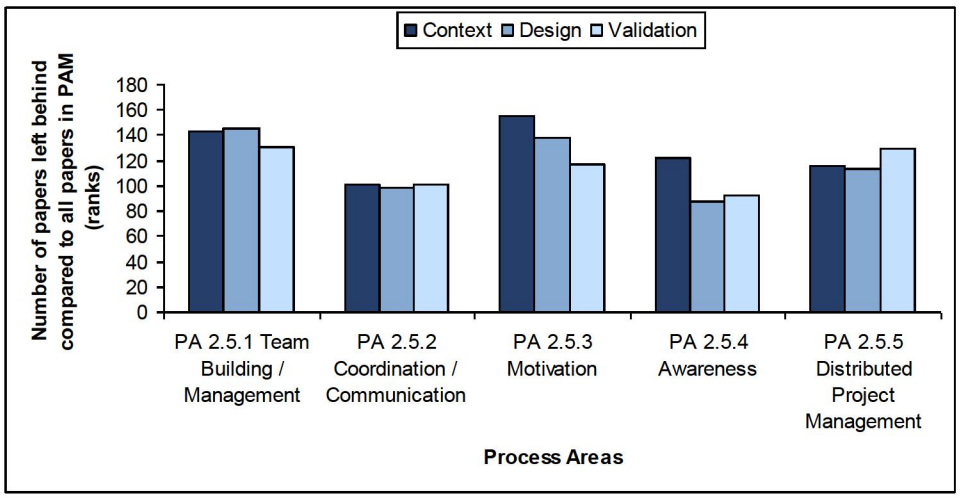

Figure 18: Ranks of the new PAs as discussed in 3.4.3

As described in 3.4.3, there are some PAs containing many papers. These PAs have been split up to make the model usable in practice. Five new PAs have been added in *PA 2 Project Management*. The average ranks for these five new PAs are shown in Figure 18. *PA 2.5.1 Team Building/ Management*, for example, is rated quite high in all three areas whereas in *PA 2.5.3 Motivation* the PA Context seems to be very well described with a rank of 143 while Validation fairs a bit worse, with a rank of 130. Compared to the averages shown in Table 19, it can be seen that all PAs, that were added, are ranked above average. It can also be seen, that *PA 2.5.5 Distributed Project Management* seems to be well validated with a ranking of about 140. Recalling the average Validation ranking of all papers from Table 19, which was 81.7, *PA 2.5.5 Distributed Project Management* seems to be better validated than compared to the average.

[9] Appendix D: SLR Ratings, Appendix E: List of SLR Papers

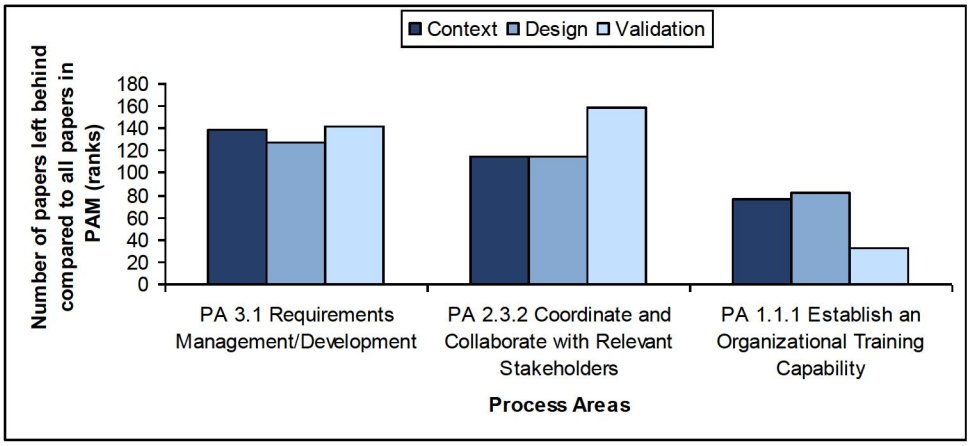

Figure 19: Ratings of sample PAs

The ranking above average is important when adding a new PA to SODE because a structural change to SODE should always be based on well-ranked PAs. This way it is ensured that the changes made to SODE are of good quality and based on information that is ranked above average compared to all papers in SODE.

Awareness, on the contrary, seems to be validated only a little above average. While this is still a good value within SODE, it is the smallest ranking within these five new PAs. However, it can be seen that the new *PA 2.5 Distributed Team and Project Management* is ranked above average as every child PA is ranked above average.

PA 3.1 Requirements Management/Development, as can be seen in Figure 19, is also an example of a very well validated PA. Context (139), Design (127) and Validation (142) are all comparatively good ranks. After all, a paper randomly picked from *PA 3.1 Requirements Management/Development* leaves 142 papers behind in Validation. This means, that papers in this PA are on average validated better than 142 papers compared to all papers in SODE. That is definitely a high ranking.

It can also be seen that *PA* 2.3.2 *Coordinate and Collaborate with Relevant Stakeholders* is ranked quite good while it is exceptionally highly ranked in Validation (158). This could be very interesting for industry as validation is the key factor indicating that a solution actually could work.

PA 1.1.1 Establish an Organizational Training Capability is an example of a PA that is ranked very low. Especially Validation with a value of 32 leaves a lot of room for improvement. It can also be seen that, compared to the averages given in Table 19, this might be a PA that really needs some additional work.

It is also interesting to know the distribution of the ratings as the distribution can tell us a lot about the quality of the papers or at least, implicitly, the ratings the reviewers gave to the

papers. The distribution should hopefully look somewhat as a bell-shaped curve so that really good papers are quite rare (as they are in reality) and really bad papers should also be rare as they should have been excluded during the SLR and by reviewers in scientific journals. As can be seen in Figure 20, the distribution of the ratings is quite bell-shaped. The ratings are centered and the tails are slim indicating only a few very good and very bad papers. This applies to both the ranks as well as the ratings, as the ranks have been calculated from the ratings only.

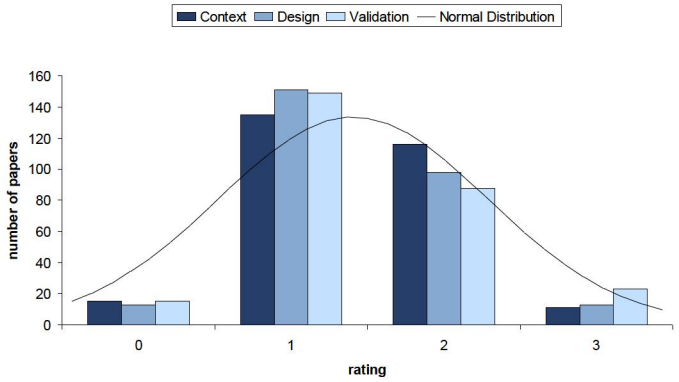

Figure 20: Distribution of the ratings

Another interesting question, arising in this context, is whether the ratings are correlated amongst each other. As can be seen in Table 20, the empirical correlation between the ratings is all positive. This means that if one of the ratings is high, the others are more likely to be high, and if one rating is low, the others will be, too. So, this could indicate that some papers are written well, and receive high grades in every dimension, while others receive lower grades in every dimension.

Dimensions	Correlation
Context/Design	0.73
Design/Validation	0.42
Context/Validation	0.44

Table 20: Empirical correlation between ratings

However, the empirical correlation seen in Table 20 is not really strong positive. For example, the empirical correlation between Context and Design (0.73) means that 53.3% (0.73^2=0.533) of the variance is common for these two areas. For Design and Validation, however, only 17.6% of the variance is in common while 82.4% is not in common. This indi-

cates that there is a good reason for keeping the three ratings and not combining them. Also, it is good to see that they are correlated positively which means that they tend to go in the same direction while not being completely superfluous.

3.5 Qualitative Analysis with Solution Characterization

Having finished the quantitative analysis of the SLR, this section will now describe and characterize the PAs through the solutions found. This section gives an overview of the contents of SODE to provide insight into the quality of the solutions.

Since the SLR found 127 solutions in total it would of course be impractical to cover each solution in detail. Instead clusters of solutions that were chosen might be of certain interest. Nevertheless, all solutions (with reference number, title of paper, abstract, and the ranking they received in the classification) can be found in the appendix[10].

Table 21 gives an overview of the PAs in SODE at the second level of abstraction (the sub-PAs chosen for the qualitative analysis are in bold). Each PA is rated in relation to Amount (of solutions within the area) and Quality (of the solutions in the area).

Name	Amount	Quality
PA 1.1 Organizational Training	**Low**	**Low**
PA 1.2 Organizational Process Definition	High	Medium
PA 1.3 Organizational Innovation and Deployment	Low	High
PA 2.1 Project Planning	Medium	Low
PA 2.2 Project Monitoring and Control	Low	Medium
PA 2.3 Management Process	High	Medium
PA 2.4 Documentation Process	Medium	Medium
PA 2.5 Distributed Team & Project Management	**High**	**High**
PA 3.1 Requirements Manage-	**High**	**High**

[10] Appendix D: SLR Ratings, Appendix E: List of SLR Papers

ment/Development		
PA 3.2 Technical Solution	Medium	High
PA 3.3 Product Integration	**Low**	**Low**
PA 3.4 Validation	Medium	Medium
PA 4.1 Causal Analysis and Resolution	Low	Medium
PA 4.2 Decision Analysis and Resolution	Low	Medium
PA 4.3 Process and Product Quality Assurance	Low	Medium
PA 4.4 Acquisition Process	High	Medium
PA 4.5 Maintenance Process	Low	Medium
PA 4.6 Joint Review Process	**Low**	**Low**

Table 21: Overview PAs in relation to amount of solutions and their quality

The amount is rated as *high*, *medium* or *low*. A PA is said to have a low amount of solutions if there are less than 10 solutions, medium if there are less than 25 solutions and high if there are more than 25 solutions.

The quality is also rated in three steps, *high*, *medium* and *low*. This measure is deducted from the average rating of the solutions in this area as described in Section 3.3.2.3. An area is said to have low quality if the average rating is below 50, medium quality if the rating is 50-100 and of high quality if the average rating is above 100.

As is evident from Table 21 PAs were chosen to be further analyzed that have either (according to the classification):

i) Low amount *and* low quality of solutions.
ii) High amount *and* high quality of solutions.

The reason for this selection is that, regarding the first category (low-low), these are sub-PAs that might be interesting to be further analyzed because of the possibility of finding future research topics, or to simply investigate their relevance in a GSE context. That is, these sub-PAs are candidates for removal from SODE based on the solutions found in literature.

With respect to the second category (high-high) we believe this analysis should be interesting for practitioners since they are able to choose between several solutions of high quality which would be a comfortable situation to be in for them. (This section also ties back to the results and analysis in Section 3.4.)

Nevertheless, all top-level PAs are shortly summarized and introduced now.

3.5.1 PA 1 Process Management

Looking at Figure 12, it can be seen that *PA 1 Process Management* is the smallest of the four PAs when it comes to the amount of solutions. Figure 13 further shows that PA *1.2 Organizational Process Definition* is the area with the most solutions within PA 1. In addition, Figure 19 shows that *PA 1.1.1 Establish an Organizational Training Capability* is ranked quite low with the validation ranking being the worst.

3.5.1.1 PA 1.1 Organizational Training

PA 1.1 has been characterized low on quality as well as quantity of solutions (Table 21). The reason for the lack of validated solutions might be that companies have started to create solutions for GSE in the recent years. These solutions might not yet be mature enough and thus be included in training for a distributed context. Therefore, this area might get additional attention in relation to validation in future work (and thus increase PA quality). As one of the most important PAs in this area*, PA 1.1.2.1 Deliver Training* focuses on how to train teams that are not co-located, e.g. how to build a global test team through training [49]. The main goal is to understand the ineffectiveness of globally distributed test teams and get the best out of them.

3.5.2 PA 2 Project Management

In Figure 12 it can be seen that *PA 2 Project Management* is the biggest PA of all in SODE. Figure 19 further shows that especially *PA 2.3.2 Coordinate and Collaborate with relevant Stakeholders* is very well validated. Also, in Figure 18 there is a broad overview of the rankings in *PA 2.5 Distributed Team & Projecct Management* and it can be observed that they are all quite high. This PA is one of the better ranked, in total, within SODE.

3.5.2.1 PA 2.5 Distributed Team & Project Management

This PA has been identified to have a high amount and high quality. So, this area seems to be very mature in GSE. Therefore, several examples are given to demonstrate state-of-the-art in this area.

PA 2.5.1 Team Building (see Figure 18) is one of the best ranked PAs in the newly created *PA 2.5 Distributed Team Management*, from all three perspectives of context, design and validation.

To build a distributed team, some hurdles have to be overcome. Language skills and terminology differences are just two examples of problems that arise in this PA [113]. The solutions covered in *PA 2 Project Management* try to take care of the problems arising in globally

distributed teams in several ways. One focus is the setup of virtual teams, with multi-culturalism and its implications on the creativity of virtual teams, as one of the major bonuses [11], [55].

Diversity-focused solutions can however lead to effort increase in team unification. Group dynamics and cohesion in virtual teams (Nicolopoulou et al. [56]) are also a challenge addressed by many of the solutions (see e.g. van der Duim et al. [39], [54]).

Another focus area is the leadership in distributed teams and the differentiation from traditional leadership [160]. Setamanit et al. [161] or Oshri et al. [62] present solutions focusing on the importance of social ties within virtual teams as a key factor of project success. Ungerleider [109] extends this with solutions for conflict resolution.

Figure 18 illustrates *PA 2.5.2 Coordination/Communication*, which is ranked below the other PAs in this area; however, it is still above average as can be seen in Table 19. This PA addresses the dominant use of asynchronous communication which is a central challenge in coordinating virtual teams [113]. Knowledge sharing solutions for this is a pre-dominant measure (see e.g. [65], [68]). Further, the extension can be seen in solutions for decision understanding commonality [52], [63], and tool support for these activities presented by Ferscha et al. [64].

It is also important to the know rules and constraints in distributed communication. The solutions range from practical lessons learned, as described by Šmite et al. [162], to the more theoretical challenges of virtual partnerships [68].

The third example under *PA 2 Project Management* is *PA 2.5.3 Motivation*. Figure 18 shows a high rank in Context, while Validation seems to be a little low, but still above average.

Motivation of virtual teams is reported to be harder than co-located teams. One reason for this could be the lack of common goals among the distributed team members [113]. Lings et al. and Lindqvist et al. present solutions focusing on defining clear and transparent project priorities and goals [70], [163]. In addition, another important factor in distributed teams is to send out a shared vision of the project to all team members. This shared vision enables the team to accomplish the work together [46], [71].

The last example to demonstrate the maturity of this area is *PA 2.5.4 Awareness* (Figure 18). Having remote team members, it is important to make all the people in the team aware of all the other members of the team [72]. If a team is separated between different sites, it is often hard to get a team feeling [113]. This feeling can be strengthened by building trust between the team members [37], [73], [76]. Awareness can also be created by a technique that is best described as virtual co-location, presented by Herbsleb and others [10], [97] as a way to address the issues of motivation lack of and team feeling.

3.5.3 PA 3 Engineering

In Figure 12 it can be seen that *PA 3 Engineering* is only about a quarter of the size of *PA 2 Project Management*. Yet, it is double the size of *PA 1 Process Management* and offers double the amount of solutions. Table 21 shows the level of quality in this area which can be seen as relatively high.

3.5.3.1 PA 3.1 Requirements Management/ Development

Looking at Figure 14, *PA 3.1 Requirements Management/Development* is one of two areas that have a large amount of solutions (and high quality solutions according to our ranking).

To produce a product that is of high value for the customer, it is very important to know what the customer wants (*PA 3.1.1.1 Elicit Needs*). In a distributed team, problems might occur related to the limited transparency of decisions especially in the early stages of development [27]. The area of elicitation in a distributed environment has got a lot of attention from researchers as it presents a unique GSE twist to the overall challenge of requirements elicitation.

Some of these solutions focus on the techniques used to elicit customer's requirements in a distributed environment [80], [164], while others focus on the organization of virtual requirements meetings themselves [55], [79]. Thus both the actual "catching" of requirements, and the subsequent handling and understanding of them are critical to enable a common understanding between the stakeholders.

PA 3.1 Requirements Management/Development (see Figure 19) is one of the best ranked PAs in SODE. This is true for all three aspects (Validation, Context and Design). Due to the lack of synchronous communication and increased "virtualness", changes to requirements can lead to problems if not all team members get instant notification, as well as a rationale for the changes [27]. The continuous tracking of changes in requirements is a challenging task in distributed teams. Heindl et al. [165] present a solution that helps project managers to ensure the correctness and traceability of the requirements documents. Välimäki et al. [85] extend this and present a possibility to transfer requirements management practices into patterns especially tailored for distributed project management.

3.5.3.2 PA 3.3 Product integration

The amount of solutions in this area is low (Table 21) and the quality seems to be lacking. One explanation for this could be that there are few GSE-specific problems in this area. As, for example, the actual product integration is not often done in distributed teams, this might

lead to the lack of solutions (and quality)[166]. The parts might get delivered from all over the world, but the integration itself is not done in a distributed way.

However, for example Paulish et al. [167], [168] identified product integration as a central research area for distributed software engineering. Possible solutions in this area quote the product line approach as a feasible solution for product integration through a tight control of core assets and their use, especially for multi-site projects and large-scale requirements sets.

The process area of product integration should not be excluded from SODE based on the low amount of solutions, rather should the area be investigated further.

3.5.4 PA 4 Support

As can be seen in Figure 12, *PA 4 Support* is one of the three small PAs in SODE. Yet, especially in *PA 4.4 Acquisition Process*, important solutions can be found as can be seen in Figure 15.

3.5.4.1 PA 4.6 Joint Review Process

This PA has been identified to have a low amount of low quality solutions. There are different possible reasons for this. One reason might be that this area is of minor interest in a GSE context, and the area is well covered (mature) in software engineering literature and that the methodology is easily transferred to a global context. However, Duarte et al. [90], [92], [156] identified this area as important and they claim there still exist several additional challenges with reviews and inspections in a distributed environment.

3.6 Conclusions

An extensive SLR was performed to identify and map solutions to software engineering challenges in a GSE context. Further, to sort the solutions, and identify what areas were well populated, SODE 0.2 was used. Further, the "quality" of the solutions identified were rated in relation to Context, Design and level of Validation. This is central as any solution identified (and the map as a whole) should be a resource for practitioners. Thus it is important to be able to judge the level of validation, as well as in which context, and how the empirical data related to the solutions was collected, speaking the credibility of the validation.

127 solutions were identified in the review, which were subsequently rated[11]. It turned out that these ratings were correlated. However, the correlation was not high enough to com-

11

bine these three ratings into a single rating. Additionally, the ratings were clearly distributed normally, having no fat tails, which is a good sign for the rating—too many extremely good and extremely bad papers and no papers in-between would have been a sign for an ill-constructed scale.

SODE 0.2 also showed to be a good approximation to reality as it needed little tweaking. Only one PA had to be added post SLR in order to fit the solutions, however this was mainly due to the focus of the review, i.e. some areas were simply overpopulated.

Some areas such as *PA 3.1 Requirements Management/Development* and *PA 2.5 Distributed Team and Project Management* are very well populated, that is, a lot of research and subsequent solutions have been presented to tackle GSE-related problems. There are however several areas, most pre-dominantly *PA 3.3 Product Integration* and *PA 2.2 Project Monitoring and Control*, that are very scarcely populated. Researchers can use this as a clear indication for future research efforts, or at least we need to confirm that the solutions for co-located development also apply in a GSE context. Little empirical evidence exists clarifying this.

In terms of quality, the rating system of Context, Design and Validation presented in the paper can also be used to preemptively improve research quality in the field of GSE research, and the well-rated papers can be used as good examples by researchers.

SODE 0.2 was used as an input for this chapter and has evolved during the SLR. This resulted in an improved version SODE 0.4. This includes the solutions identified in the SLR and the connection to SODE. This completes the theoretical creation and revision of SODE. As SODE is designed to support practitioners as well as researchers, it is now important to get the feedback of real users. This will be done in Steps 5 and 6 as displayed in Figure 3 (on page 45). These steps will now be presented in Chapter 4. During these steps, SODE 0.4 will be evolved even further to meet the expectations of real world users. To gain this insight, expert interviews and workshops with the academical and industrial partners have been conducted. The process and results will now be presented in Chapter 4.

Appendix D: SLR Ratings

4 Evaluation

4.1 Introduction

In Chapter 2, a theoretical structure based on CMMI and ISO focusing on GSE topics, SODE 0.2, has been built and presented. This structure was then used to order solutions identified in a systematic literature review described in Chapter 3. These two results combined resulted in the creation of SODE 0.4 at the end of the last chapter.

However, one main goal of industry related research is to transfer academic knowledge into industry[144]. This can be done by repeated evaluation with the users of, in this case, SODE. After each step of evaluation, there is a feedback loop to constantly improve SODE. The evaluation is split up into expert interviews and workshops[124].

In this chapter, there will be an explanation of the actions that need to be performed to complete the evaluation of SODE in Section 4.2. After that, the process that has been followed during the evaluation will be described in detail in Section 4.3. The research questions and the goals of the evaluation will also be given. In Section 4.4, the results and analysis of the evaluation will be presented including the impact analysis on SODE resulting in the final version, SODE 1.0.

4.2 Evaluation of SODE

In this section the last two steps (Steps 5 and 6) to complete SODE according to Figure 3 (see page 45) will be explained.

4.2.1 STEP5: Improvement through specific expert interviews

SODE evaluation STEP 5 aims on optimizing specific SODE contents, their design and structure. Even though the focus is on optimizing these specific SODE contents, modifications concerning SODE's design and structure which exceed specific contents (and address SODE as a whole) are still performed if necessary.

Therefore interviews with hand-picked experts are conducted to validate quality and integrity of specific SODE contents. Experts' knowledge is analyzed focusing on improving SODE contents and SODE is adjusted according to improvement aspects indicated. For a detailed description of the expert interviews please note Chapter 4.3.1. For a detailed description of the results analysis please see Section 4.4.

Validating and improving SODE contents in terms of quality and integrity based on expert opinions leads to SODE version 0.5.

4.2.2 STEP 6: Improvement through diversified validation

Additionally to validating and improving specific SODE contents as performed within STEP 5, SODE evaluation STEP 6 finally aims on improving SODE in total through widespread validation. SODE's contents, structure and design are validated by different user groups in separate workshops.

Differences within user groups and workshops are systematically applied to get diverse results. To keep the quantity of results still manageable, questionnaires are used to simplify the collection and structuring of the results. Questionnaires are evaluated by deducing aspects to improve SODE in total and SODE is adjusted accordingly.

For detailed descriptions of questionnaire design, workshop design and constellations concerning user groups, workshops and questionnaires please note Section 4.3.1. For a detailed description of questionnaire analysis please note Section 4.4.

SODE version 0.5 has to pass a widespread validation before SODE version 1.0 is finally applicable as a **SO**lution framework for **D**istributed **E**ngineering.

4.3 Evaluation Design

4.3.1 Expert interviews

To evaluate the quality of specific, rather small parts of SODE, expert interviews are conducted. This has been chosen as the experts bring the required knowledge that is needed to develop SODE even further. [124]

Fundamental for valid result quality is the structuring of interviews. They have to be prepared, conducted and analyzed (please note Figure 21).

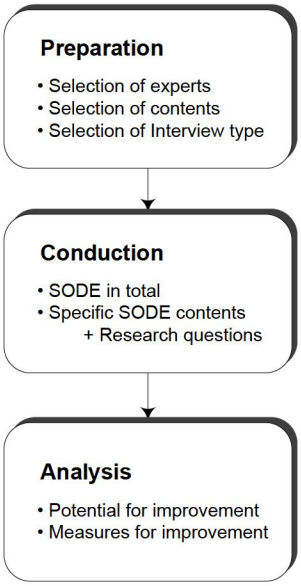

Figure 21: Expert Interviews

4.3.1.1 Preparation

The preparation is of central significance. The high effort of expert interviews reduces their number of conduction massively. Therefore not only the selection of experts per se is of high importance but also the selection of contents which are to be evaluated.

Within the evaluation of SODE five experts are interviewed on the topics “Global Software Engineering”, “Project Management”, “Process Management”, “Requirements Engineering” and “Safety”. Experts are characterized by their fields of work and their experiences within these fields.

As expert interviews aim on evaluating specific parts of SODE, RQs are predefined (please note Table 22). To keep the effort of the interviews manageable, expert interviews focus on evaluating completeness and structure of SODE using RQ1, RQ1.1, RQ1.2. The use of these research questions within semi-structured interviews enables the interviewer to combine a guided interview with an open discussion by giving them freedom in the sequencing of these predefined research questions, their exact wording and the amount of time and attention given to each of them [124].

ID	Research Question	Description
RQ 1	**Is SODE complete?**	The completeness of SODE is analyzed with a focus on the structure and the contents of SODE.
RQ 1.1	Is the structure of SODE complete?	The completeness of the structure of SODE examines missing contents, superfluous contents as well as the sorting of the solutions into SODE.

RQ 1.2	Is the content of SODE complete?	This examines the completeness of the content of SODE. Especially, white spots in SODE are examined to detect the causes for the lack of solutions in this content.
RQ 2	**Is SODE usable?**	The usability of SODE measures the ease of use when trying to find solutions for a specific problem.
RQ 2.1	Is it understandable how to use SODE?	Is SODE usable without any training other than the written instructions associated with SODE?
RQ 2.2	Is previous knowledge needed to use SODE?	To what degree does knowledge of process models such as CMMI, ISO etc. support the use of SODE?
RQ 2.3	Is the structure of SODE easy to understand?	Is the overall structure of SODE understandable? Is it easy to see the structure of SODE? Are the different layers detectable and how well are the layers understood?
RQ 2.4	Does the structure of SODE provide an easy overview of the contents?	Is this structure a good way to provide the solutions? Does the structure simplify the order of the solutions to provide a clear overview of the solutions in SODE?
RQ 2.5	Is the navigation through SODE clear?	Does SODE support an easy navigation through SODE to find the desired solutions?
RQ 3	**Is SODE useful?**	This question examines the usefulness of SODE to provide a solution with reasonable effort.
RQ 3.1	Does SODE provide decision support?	To what degree can the solutions be used and implemented in practice addressing a given problem?
RQ 3.2	Is SODE coverage adequate?	Are the solutions presented in SODE relevant to practice and do they solve the problems?
RQ 3.3	How is the quality of the solutions?	Can the solutions be used to solve problems or are the solutions either to obvious or to complicated? Could they be implemented in practice?
RQ 3.4	Was the time to find a solution reasonable?	This depends on the quality of the solution: It might be reasonable to find no solution in short time, also, it might be reasonable to search for some time to find a perfect solution.

Table 22: Research Questions for the evaluation of SODE

As a result of the preparation, experts and corresponding specific parts of SODE are selected, semi-structured interviews are decided to be used and topics in terms of research questions are defined to be covered within the interviews.

4.3.1.2 Conduction

To point out SODE's general idea and its objective to provide a SOlution framework for Distributed Engineering, each interview starts with a presentation of SODE in total, its overall structure and its classification in regard to existing models.

Afterwards specific contents and their structures mapping experts' knowledge are examined in detail. While analysing these specific contents, research questions resulting from the preparation are interspersed. Thereby the types of semi-structured interviews enable the interviewer to transform a guided interview into an open discussion.

All experts are interviewed individually. Their comments are recorded to analyse the interviews afterwards.

4.3.1.3 Analysis

Within the analysis, 'potential for improvement' is identified by evaluating the recorded comments. Therefore, 'potentials for improvement' are compared with aspects inducing SODE's as-is state and 'measures for improvement' are evaluated.

Whether SODE is adjusted according to these measures or not depends on their impact on SODE in total. A detailed description of results and analysis of the expert interviews is given in Chapter 4.4.

4.3.2 Workshops

In contrast to the purpose of the evaluation of SODE by expert interviews, workshops aim on evaluating usability and usefulness of SODE by enquiring the research questions compiled by RQ2 and RQ3. For that reason, SODE in total is evaluated and not only specific contents of it. This stretch of coverage is accompanied by an increase of potential participants and evaluation results. Therefore, SODE is evaluated by focus groups within workshops.

Even though the necessity of experts is no longer given, group size and composition of the focus groups are still significant for high quality results [124].

Within SODE's evaluation, focus groups are intentionally kept small (5 to 9 participants each group) to strengthen interaction with the workshop participants. They are categorized as "Managers", "Researchers with knowledge of global software engineering", "Researchers without knowledge of global software engineering" and "Skilled workers". This categorization aims on evaluating whether previous knowledge is helpful to understand SODE or not (RQ2.2).

But previous knowledge could be generated shortly through training. Furthermore, a specifically-tailored training might lead to an even better understanding of SODE and its use.

To pay attention to these assumptions, two different workshop concepts are designed (please note Figure 22).

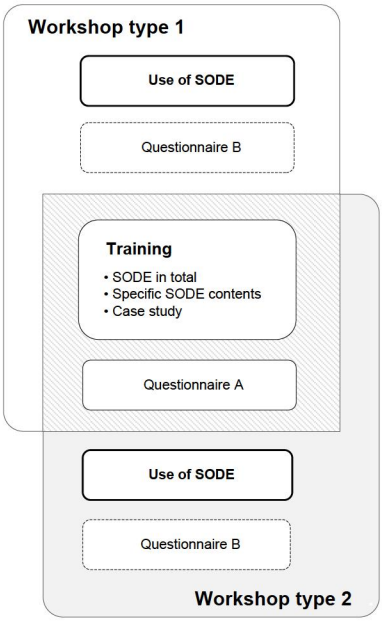

Figure 22: Workshops

The main difference between workshop type 1 and workshop type 2 is that focus groups attending workshop type 2 are given a training of SODE before using it while focus groups attending workshop type 1 are not. Second group participants have to face SODE by themselves and are given the training afterwards.

According to these two workshop concepts, two different questionnaires are used to simplify collection and structuring of the results. Questionnaire A[12] evaluates the usability of SODE by addressing RQ2.1, RQ2.2, RQ2.3 and RQ2.4. It also enquires the necessity of training for using SODE. Focus groups of both workshop concepts have to complete it directly after the training.

Contents of the training are the same within both workshop concepts and comparable with the presentation of SODE within expert interviews.

First SODE's general idea and its objective to provide a SOlution framework for Distributed Engineering is pointed out. Therefore SODE in total, its overall structure and its classification in regard to existing models are shown.

[12] Appendix A: Questionnaire A

Afterwards specific contents and their details are shown exemplarily, not as in-depth as within expert interviews, but to give an idea about the structure, design and amount of informational content of SODE.

Finally, a case study demonstrates the use of SODE, starting with a fictional problem concerning global software engineering and illustrating the entire process to find a solution with the help of SODE.

The completion of questionnaire A by focus groups passing workshop type 1 (who have received the training after using SODE) reveals differences between the experiences participants made with SODE and SODE's real intent.

The completion of questionnaire A by focus groups passing workshop type 2 (who have only received the training but not used SODE so far) provides information about their suggestion whether the training got them to know how to use SODE and whether this information will simplify their use of SODE or not. The correctness of their suggestions is validated by questionnaire B which is completed after they used SODE by themselves.

Questionnaire B[13] focuses on the usefulness of SODE and its contents by using RQ 2.5, RQ3.1, RQ 3.2, RQ 3.3 and RQ 3.5.

Both questionnaires are designed the same way to make results comparable. At the beginning, data characterizing participants like "previous knowledge of process models" or "experience in projects concerning global software engineering" is collected to structure results according to focus groups categorization.

Following questions are close-ended using Likert scales with an uneven number of check boxes covering an acceptance range from "absolutely" to "not at all".

For a detailed description of questionnaires analysis please note Chapter 4.4.

4.4 Validation – Results and Analysis

The results of the evaluation will be structured according to the Research Questions given in Section 4.3.1.1. In each section there will be the feedback from all relevant workshops and expert interviews to answer this question.

30 people were involved in the evaluation of SODE. The participants of workshops were given questionnaires to gather their feedback and input. The return rate of the questionnaires was over 80%.

[13] Appendix B: Questionnaire B

4.4.1 RQ1: Is SODE complete?

One of the most important things to find out in the evaluation was whether SODE was complete or not. The completeness of SODE was reviewed in two contexts, the structure and the contents.

4.4.1.1 RQ1.1 Is the structure of SODE complete?

4.4.1.1.1 Overview and Analysis

The completeness was mainly evaluated with expert interviews as it is quite time-intensive to get an overview of the structure of SODE. Within the expert interviews, the area of expertise was the main parts that have been covered to evaluate the completeness.

The GSE expert raised some issues about the redundancy of some topics that could be integrated into IPPD add-ons to CMMI as well as the newly-created *PA Distributed Team Management*. It was therefore decided to leave the IPPD add-on out of the scope of SODE and integrate everything in one place, the newly-created PA for simplicity. Also, some of the empty areas in SODE have been confirmed to be important areas in GSE projects, like *Organizational Training*.

The Project Management and Process expert could also not find parts that have been forgotten in SODE. He suggested to move risk-related solutions from *Distributed Team Management* to risk specific areas of SODE like *Identify Project Risks*. He also suggested to remove the PAs *Establish the Projects Objectives, Compose the Defined Processes* and *Select the Subprocesses that will be statistically Managed* from SODE because, at least at Daimler AG, these do not really differ from traditional projects.

The Requirements expert mainly raised an alternative structure for *Requirements Management/Development* because of one empty PA in the present structure. The new structure was not implemented as it would also have one empty area and would therefore not solve any problems. For consistency reasons, the old structure was kept.

The Safety expert (ISO 26262) suggested to use the CMMI +SAFE add-on for SODE. This was implemented as it was the best solution to integrate safety specific contents into SODE.

The Management Group complained that no legal issues are covered by SODE. As this was out of the scope of CMMI as well, these otherwise interesting topics were left out of SODE as well.

4.4.1.2 RQ1.2 Is the content of SODE complete?

4.4.1.2.1 Overview and Analysis

Additional input was mainly provided by the Safety expert. Some solutions were also provided by the Project Management expert and the academic group. All experts were very surprised of the amount of solutions offered by SODE.

As can be seen in Figure 23, the coverage of SODE was mostly reported to absolutely cover the problems raised by GSE projects. Seven people answered positively about the coverage of SODE (see the two left bars). The four people (the two right bars) that tend towards the

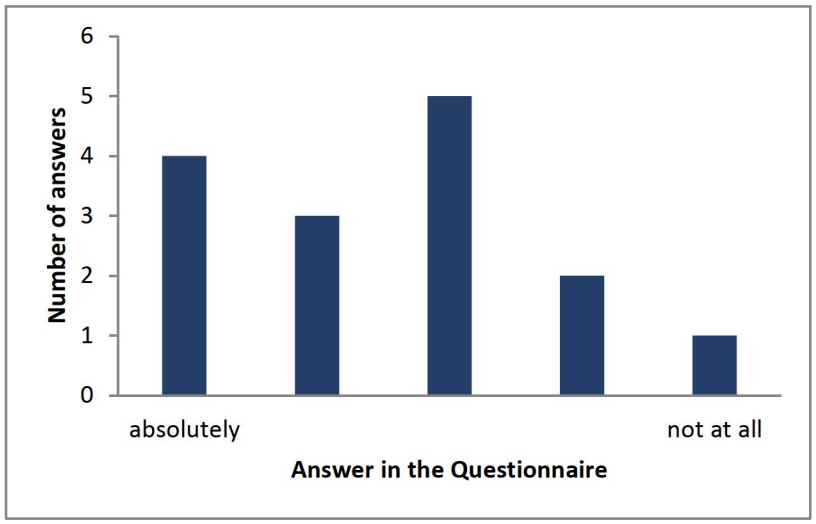

Figure 23: Coverage of SODE

solutions not at all addressing the problems mostly stem from the management group that missed the legal issues.

4.4.1.3 SODE impact summary

The impact of RQ 1 and how SODE has been changed and refined is described in Table 23.

PA ID	Action
Enable IPPD Management	Deleted
Apply IPPD Principles	Deleted
Establish the project's Objectives	Deleted
Compose the Defined Process	Deleted
Select the Process that will be statistically managed	Deleted
PA 2.3.8 Safety Management	Added
PA 2.3.8.1 Develop Safety Plans	Added
PA 2.3.8.2 Monitor Safety Incidents	Added

PA 2.3.8.3 Manage Safety related Suppliers	Added
PA 2.4.5 Distributed Project Management	Restructured
Documentation Process: Production	Deleted
Documentation Process: Maintenance	Deleted
PA 3.5 Safety Engineering	Added
PA 3.5.1 Identify Hazards, Accidents and Sources od Hazards	Added
PA 3.5.2 Analyze Hazards and perform Risk Assessment	Added
PA 3.5.3 Define and Maintain Safety Requirements	Added
PA 3.5.4 Verification	Added
PA 3.5.5 Safety Analysis	Added
PA 3.5.6 HW/SW Integration	Added
PA 3.5.7 System Integration /Verification	Added

Table 23: SODE impact summary for RQ1

4.4.2 RQ2: Is SODE usable?

Usability is also a very important issue of a model like SODE. If a model is dependent on input from practice, people have to use it. SODE can offer the best solutions to its users. If they are not able to find them or find them in a reasonable time, SODE will not be used.

4.4.2.1 RQ2.1 Is it understandable how to use SODE?

4.4.2.1.1 Overview and Analysis

This RQ is answered in two steps. The workshops were designed to provide some groups with the training before the use of SODE, other groups had to use SODE without any training.

In Figure 24, the results without training can be found as bright bars. There are quite a few persons who are not able to use SODE in a way that leads them to the solutions they are looking for. However, it turns out that seven of these eight people, which is over 80%, had no experience in using process models. This can be seen in the two right bars. It is no real surprise that they were not able to use SODE without problems.

If training is involved, the results can be found in Figure 24 as dark bars. These results look very much more promising. Yet, there are still some people who are not able to use the model. However, it turns out that these people mostly were not able to find a solution at all.

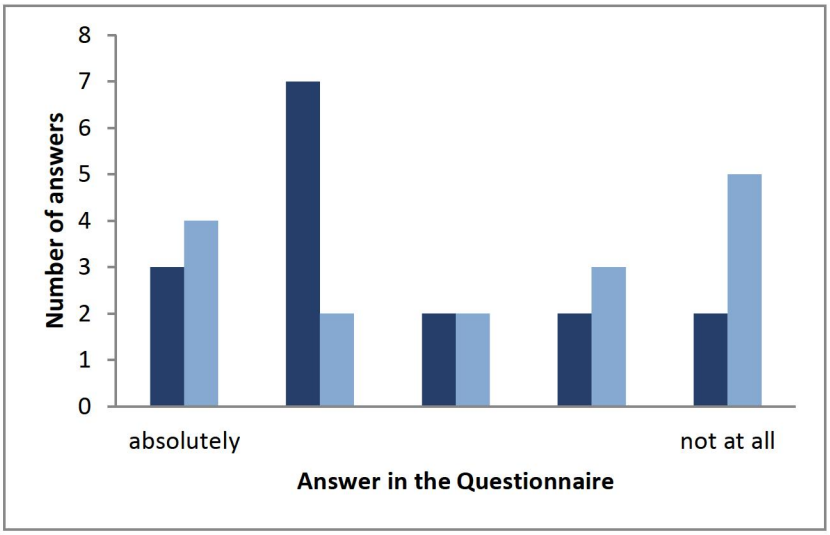

Figure 24: Usability of SODE with (dark bars) and without (bright bars) training

But, if there are blank spots in SODE, some problems just do not have solutions within SODE. So, these people might have used the model in the right way but do not know or feel so as they did not end up with a solution.

4.4.2.2 RQ2.2 Is previous knowledge needed to use SODE?

4.4.2.2.1 Overview and Analysis

As indicated in the answer to RQ2.1, previous knowledge is a bonus when using SODE. Figure 25 shows, that the users feel just the same way. Indeed, most of the people think that previous knowledge is needed especially with respect to process models. Previous knowledge for

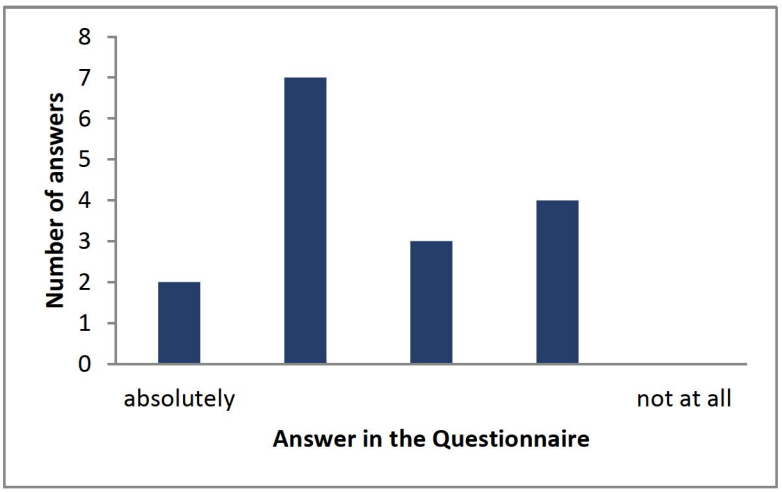

Figure 25: Importance of previous knowledge

GSE topics is not needed as the solutions offered by SODE are very well described according to the workshop participants.

4.4.2.3 RQ2.3 Is the structure of SODE easy to understand?

4.4.2.3.1 Overview and Analysis

The structure of SODE was part of this evaluation as there was an added layer to connect the solutions in the papers to the model. Because of this, the users were asked whether the structure was easy to understand. As can be seen in Figure 26, most of the participants think that the structure of SODE is indeed easy to get and use.

Figure 26 also shows that the participants were not sure whether this is the best structure to display solutions. Most of the complaints came from users who did not have previous process model knowledge and therefore had problems using SODE.

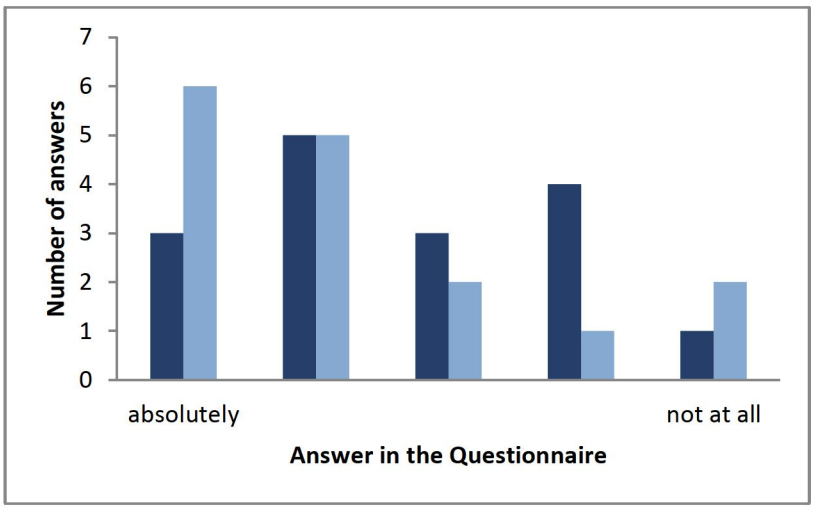

Figure 26: Is the structure of SODE understandable (bright) and adequate (dark)?

As a result, tool tip texts were implemented to guide the user through the first three layers of the model. These texts help to understand the contents that are located underneath a special process area.

4.4.2.4 RQ2.4 Does the structure of SODE provide an easy overview of the PAs?

4.4.2.4.1 Overview and Analysis

This RQ is especially interesting as the differences between the treatment of the two groups, with and without training, can be compared.

In Figure 27 it can be seen that the users got a good overview of the structure of SODE even without training. There are only very few people complaining about the size of SODE which, of course, leads to some problems. Comparing this to the red bars in Figure 27 the overview is even better with the training. As this is no surprise, it is interesting that there is about the same amount of people in the groups that received trainings that did not see the structure compared to the groups with training. So, the people with training profit a lot from it while there seems to be some people who did not benefit from it.

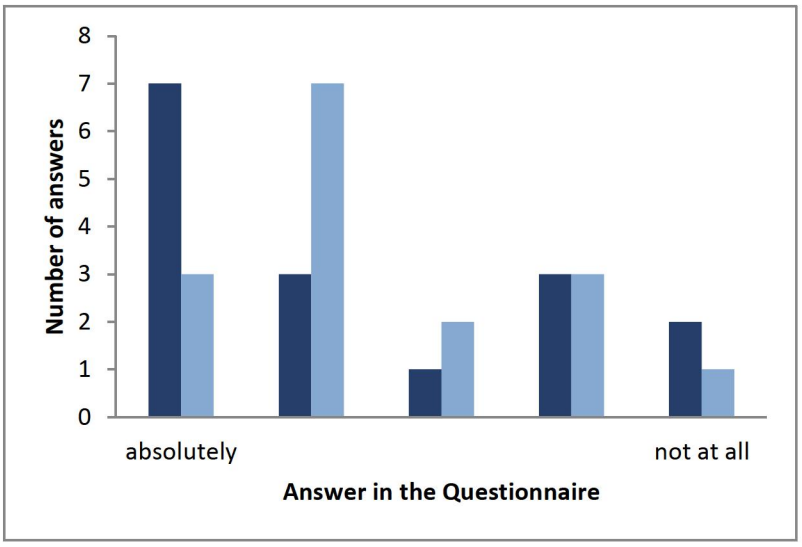

Figure 27: Overview with (dark) and without (bright) training

4.4.2.5 RQ2.5 Is the navigation through SODE clear?

4.4.2.5.1 Overview and Analysis

In this RQ, the results also differ a lot between the two different groups. In Figure 28, groups without training state that, although having no big problems, the navigation was challenging. This resulted in a change of technologies for the user interface. First, Excel was used as user interface for SODE but this was changed after the workshops to a much more intuitive web User Interface. The results can be found in Figure 28 in the dark bars. After the change, the model was found to be much more intuitive.

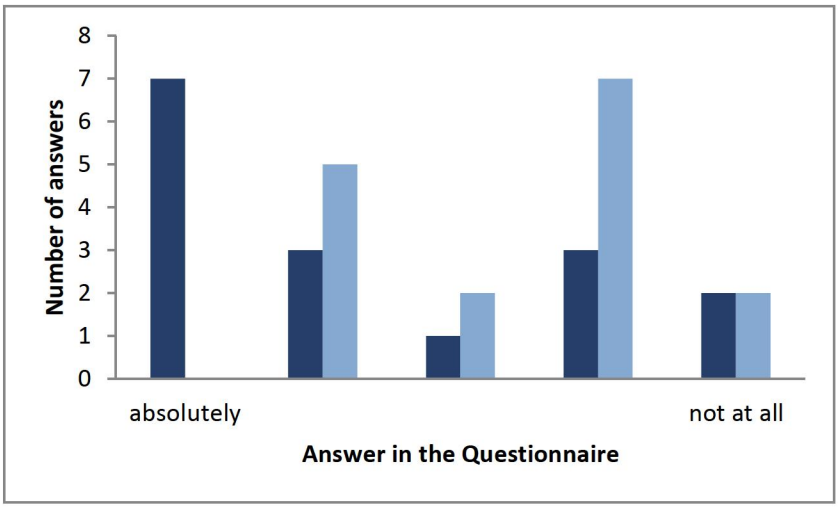

Figure 28: Problems (bright) vs. Intuitiveness (dark) of navigating through SODE

4.4.2.6 SODE impact summary

The impact of RQ 2 and how SODE has been changed and refined is described in **Error! Reference source not found.**. The results in this section are mainly based on the data that were gathered within the questionnaires. This includes the comments section for each question.

PA ID	Action
All top-level PAs	Added ToolTipText
All solutions items in the connection layer	Added description
SODE (complete)	Web interface for better usability

Table 24: SODE impact summary for RQ2

4.4.3 RQ3: Is SODE useful?

The last RQ covers the usefulness of SODE. Of course, a complete and usable model is of no value if it is not useful. Therefore, all three parts together provide a good overview of the quality of SODE.

4.4.3.1 RQ3.1 Does SODE provide decision support?

4.4.3.1.1 Overview and Analysis

Decision support is what the solutions should offer to the users. If the solutions offered by SODE do not offer support in the decision what to do in a special problem setting, the solution is not worth very much.

As can be seen in Figure 29, most of the users think the solutions provided offer decision support. Yet, the results could be better. This means that most of the solutions offered by SODE do not completely address practical problems. This should become better once SODE is used and filled in a real life project. It is a good sign, though, that the initial set of solutions offered in SODE already support practitioners.

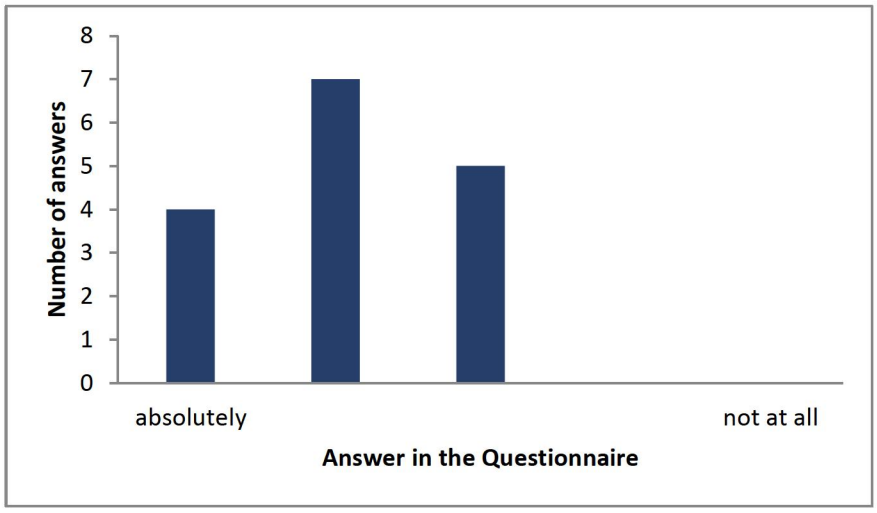

Figure 29: Decision support provided by SODE

4.4.3.2 RQ3.2 Is the coverage of SODE adequate?

4.4.3.2.1 Overview and Analysis

The coverage of SODE, regarding the solutions offered in SODE, is another key quality issue. Figure 30 shows that the coverage seems to be very good. So, neglecting the blank spots, the solutions in a process area are considered absolutely usable in practice.

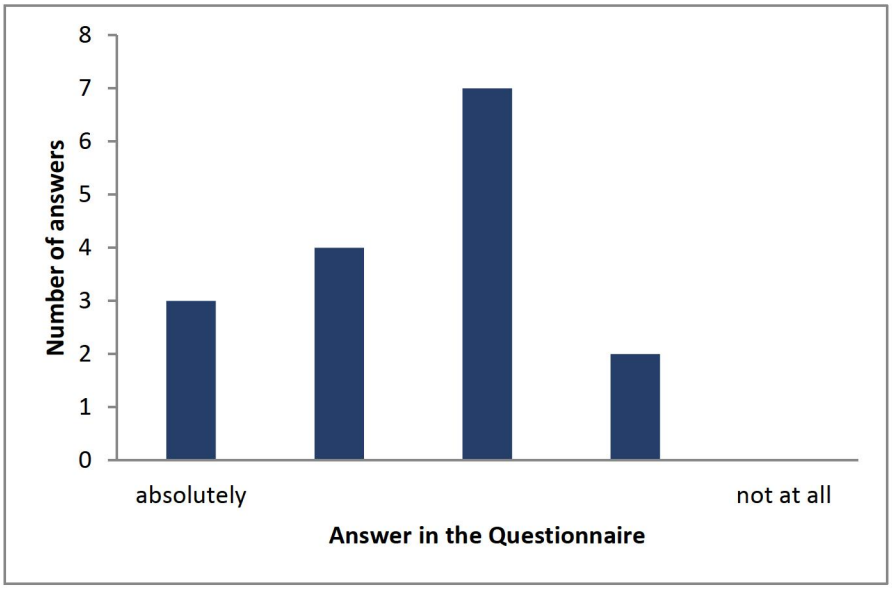

Figure 30: Is the coverage adequate?

4.4.3.3 RQ3.3 How is the quality of the solutions?

4.4.3.3.1 Overview and Analysis

While the solutions offered were considered usable in practice, this does not reflect the quality of the solutions. As can be seen in Figure 31, the quality of the solutions can be considered improvable. This means that the solutions are based on papers and are not really usable out of the box. These solutions still need work to implement the actual solution and this costs some additional effort that not everybody is willing to spend.

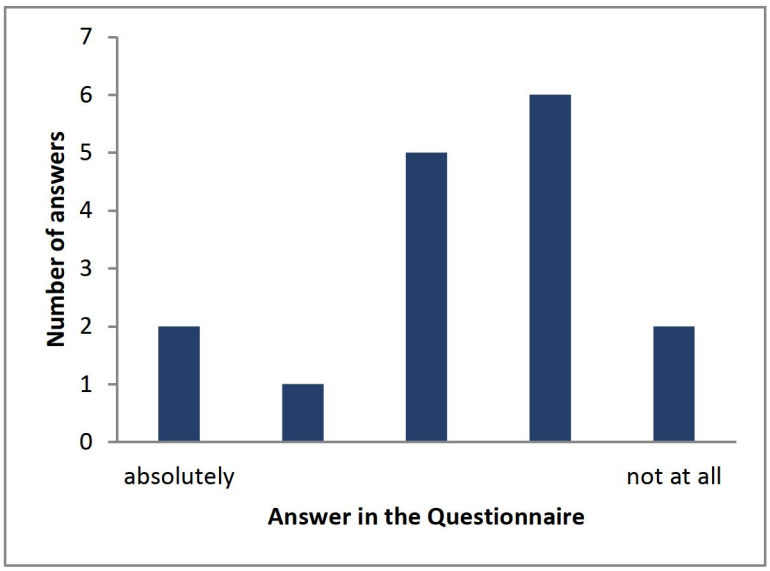

Figure 31: Quality of the solutions offered by SODE

4.4.3.4 RQ3.4 Was the time to find a solution reasonable?

4.4.3.4.1 Overview and Analysis

The time to identify the solution that solves the problem at hand is very important to users. Figure 32 shows that all solutions were identified within 15 minutes. Most of the solutions were identified even faster and about 40% of all solutions were found within five minutes. This is quite impressive as this was the first time the users had the chance to work with SODE. Taken into account that these times will improve over time, SODE is really usable.

It is interesting to see, though, that 80% of the users found the time to identify solutions to

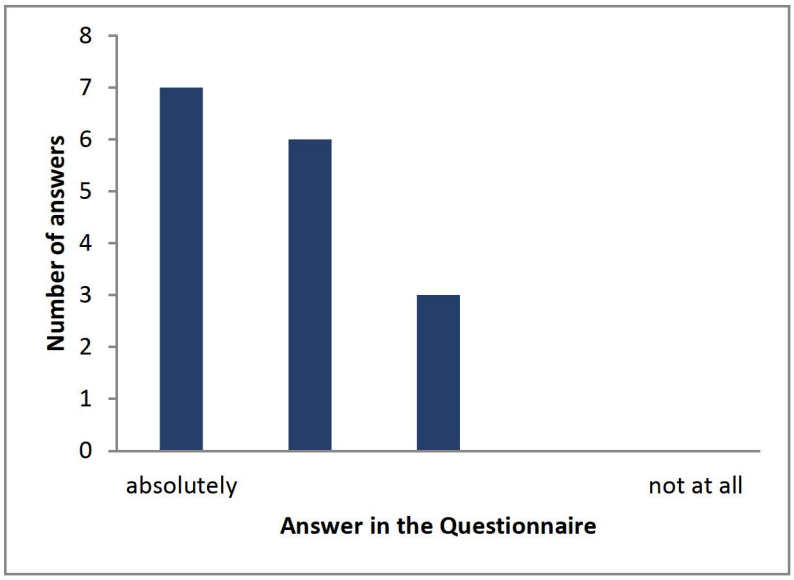

Figure 32: Time to identify a solution

be reasonable. So, people generally think the time spent on using SODE is well spent. Although, one user was very unhappy with the time he had to spend looking for an answer. From the comments it can be told that this user looked for a solution for a special GSE task where there is no solution. The problem, however, was that he thought he had identified a solution because of a misleading formulation of a solution. This solution has been renamed to describe the solutions it contains better.

4.4.3.5 SODE impact summary

The impact of RQ 3 and how SODE has been changed and refined is described in

PA ID	Action
All solutions	Display the abstract of the papers
All solutions	Possibility for links between solutions (shortcuts)
All solutions	Display ratings to indicate the quality of the solution
All solutions	Possibility for user ratings

Table 25: SODE impact summary for RQ3

4.5 Conclusion

In this chapter, the evaluation of SODE, a solution-focused framework for distributed engineering, has been presented. SODE has been based on several well-known and widely-used process frameworks and models to ensure easy adaption to the daily work life of practitioners as presented in Chapter 2.

These frameworks, however, did not fully match the needs of distributed work. Therefore, SODE was adjusted to fit those needs better. This was mainly done with a systematic literature review as described in Chapter 3. Based on Figure 3 (see page 45), these two chapters give the result of the first four steps. The last two steps are mainly used to check SODE 0.4 with respect to relevance for practice and academia. This evaluation has been presented in this chapter using interviews and workshops to get the input of practitioners and researchers.

The results of these interviews and workshops along with the changes they brought to the model are also presented in this chapter. This input was then used to improve SODE 0.4 and to develop it into a usable and useful tool. The result, SODE 1.0, is a framework that supports practitioners in industry. It also helps researchers in industry as well as academia to identify future research. This completes the steps that have been designed to create a solution framework for distributed engineering, SODE.

However, one of the big complaints about SODE during the evaluation was the existence of white spots. These are areas that do not offer solutions because the systematic literature review did not identify any.

During the collaboration with the industry partner, there has been a need for a solution in a globally-distributed engineering project at Daimler Trucks. The problem was the number of variants in this project and the tendency to create new variants for remote markets even if the need to create a new variant was questionable.

The solution for this problem has been developed in close cooperation with the industry partner. The next chapter explains how new solutions can be created and sorted into SODE presenting the example to this problem. This can be used as a roadmap to exemplify how SODE needs to be used and filled to provide support for practitioners.

5 Software Variant Reduction and Control through Factor Analysis

5.1 Introduction

Having finished SODE 1.0, there is still work left to refine it. To keep SODE alive, it needs to be filled with solutions constantly. The creation and sorting of one solution for Daimler AG will be shown in this chapter.

Software variants are very common during product development of software intensive systems. In complex projects they might even occur in the end product, or be an intended part of the overall product offering, like in the case of product line engineering [57], [169], [170] . Variants of software cover the same area of expertise but may differ in functionality and features. For example, Microsoft Office comes in different versions (XP, 2003, 2007), and also different variants (Professional, Student, and Home).

This is all well and good if the variants are created intentionally, however, as a product grows, and the software complexity increases, the number of variants may increase unintentionally. This can present a large software engineering challenge [170], [171] . Some of the problems that arise with a growing number of variants are parallel work to fix bugs, keeping track of the features in each variant, as well as the overall maintenance cost related to many variants [172]–[174] . In addition, as versions of products increase, and the necessity to offer several of these versions simultaneously to the market, an increase in cost follows.

The problem with unintentional variant creation is omnipresent at Daimler AG, with their complex code base, and offering of multiple products and multiple versions of said products. For example, building an Engine Control Unit for different markets, targeted at globally distributed sales and distribution, demands variants, but at the same rate increases complexity of development and maintenance. Add to this different brands, that partly use the same software, and a globally-distributed development organization. This is the reality of Daimler AG where there is a need for variants, but also for the control, and reduction of superfluous variants continuously. The aim being to reduce duplicates, which is aggravated in large-scale distributed development, but also controlling variant amount maintaining cohesion and lowering maintenance costs [57], [169], [170].

This chapter focuses on presenting a method, developed and currently in use at Daimler AG, aimed at providing decision support to engineers and managers looking to reduce, and even merge, variants. That is, to decide whether variants in a piece of software can be reduced by

combining some of them into one variant, or simply not introducing a variant but expanding/changing a present one. The method presented in this chapter, simply called the Factor Analysis Method or FAM for short, uses statistical analysis of the software and its features to map structure and features. Specifically, factor analysis [146], [175] is used to identify the structure of the variants, i.e. identifying closely related variants, which can imply that the variants are candidates for merger, and offer decision support for managers during development.

As the method is used, related variants are mapped and an alternative variant structure is suggested through the presentation of candidates for merger. However, in practice the method does not claim automation, rather the production of alternative structures as decision support for engineers and managers. Additional methods for selecting between different candidates [174], [176] are thus needed.

The chapter is structured as follows. Section 5.2 gives background information by introducing the context of the case, i.e. Daimler AG, and the projects utilizing the method. In Section 5.3, the factor analysis method (FAM) is explained and exemplified. Every step of the method will be explained in detail, and an example case will be used for exemplification. The necessary theoretical background to understand FAM will also be described in more detail. However the mathematics will be limited to what is needed in a software engineering context. The detail of the method is presented to ensure that the reader is able to reproduce the results of the analysis presented in this chapter, and can use the method in their own context. It also gives researchers the transparency to judge the usability and usefulness of the method. Section 5.4 presents the conclusions and experience from project work using FAM. Due to confidentiality it should be observed that the exemplification of FAM, for the purposes of this chapter, is on synthetic data (although a realistic, close to real data example is used).

5.2 Background and Related Work

5.2.1 Daimler AG

In the car industry, it should be noted that the lifetime of a variant, and the features in this variant, can be very long, up to 15 years of production time (lifetime even longer), and with the introduction of 6 variants per car generation, and almost 500,000 vehicles in 2008, the number of variants can become large, aggravated by the introduction of unintentional, superfluous and/or duplicate variants. Adding to this is the size and distribution of the devel-

opment organization, making it impossible for maintaining an overview of all products and variants.

5.2.2 Related work

The development of variant analysis is closely related to research on duplicates, like the work of Koschke [177] shows. Koschke has done extensive research in the appearance of redundancy, cloning, duplication and similarity of software products. Baker [178] has also worked on finding duplicates in software products, aided by specialized applications focusing on textual duplicates in source code. Daimler AG could not use this solution as their developers seldomly work on source code level, but on models, or on natural language descriptions.

Frenzel [179] presented a method to reconstruct the architecture to identify variants. Architecture and design reconstruction, however, is not the problem in the case of Daimler AG as they are already on a model level. Furthermore, the identification of variants is not the issue, rather the identification of variants that are overlapping, enabling reduction. Duszynski [180] showed a way to visualize software variants for comparison purposes. This is an interesting visualization method, however it leaves the reduction problem untouched.

A method to examine software variants according to their features, to enable and give decision support as to which variants to discontinue and/or combine and to reduce the number of variants was missing. FAM (see Section 5.3) is based on factor analysis, and supports finding statistically significant overlaps between variants. The prioritization of requirements (and indirectly variants in some cases), as well as the overall product management business decisions on what variant to go with, is out of scope for FAM and this chapter per se. However, FAM can be used by product managers as a decision support tool as it identifies variant proximity and candidates for merger.

Similar use of factor analysis as in the case of FAM can be primarily found in applied statistics. Stadel [181] showed that a similar approach could be used in medicine to investigate cancer rates and environmental factors. In addition, in the field of psychology, Bartlett [182] used factor analysis to study mental factors in human behaviour. The fact that the Bartlett study was published in 1954, and Stadel published his work in 2007, shows that factor analysis has been in use over a long period, and is still relevant for data analysis where the actual application area is the main focus for development. FAM thus has a solid and proven base, and is applied on a software engineering problem which in this chapter is illustrated through examples taken from the substantial application of FAM at Daimler AG, where embedded, software intensive, products are the primary focus.

In a software engineering context the use of principal component analysis (PCA) [183], for variant functionality overlap, can be seen as similar to FAM. However, there are two central arguments for factor analysis and FAM. FAM as PCA described the data, but FAM also models the data as well as possibly enabling rotation (see Section 5.3.6.1), which in turn offers better visualization of data sets where correlation is hard to find. The main advantage of FAM is however that it allows for groupings of variants into separate factors ensuring that variants in different factors are not correlated [146] . If this was the case (like in PCA), variants grouped in different factors could have strong correlations – and this is one of the things to be avoided when trying to figure out what variants could be candidates for merger [146], [184].

FAM can also be put in relation to software reuse in general[185]. However, it should be noted that the main focus of FAM and variant reduction analysis is to find variants that are candidates for merger (or identify variants that are close to each other). Furthermore, this is on existing software, thus software reuse as such falls out of scope for this chapter.

5.3 Factor Analysis Method (FAM)

FAM identifies variants that overlap in their functionality. A list of features is fed into FAM, factor analysis is performed on the data, and finally a data correctness check is performed.

The essential purpose of factor analysis is to describe, if possible, the covariance relationships among many variables in terms of a few underlying, but unobservable, random quantities called factors. Basically, the factor model is motivated by the following argument, assume that variables can be grouped by their correlations. That is, assume that all variables within a particular group are highly correlated among themselves, but have relatively small correlations with variables in a different group. Then it is conceivable that each group of variables represents a single underlying construct, or factor, that is responsible for the observed correlations. Factor analysis can be considered as an extension of principal component analysis. Both can be viewed as attempts to approximate the covariance matrix [146].

Please note, that this chapter will not give a complete insight into factor analysis. Illustrations, such as proofs in relation to the methods and theorems used in factor analysis, or descriptions of factor analysis in a mathematically complete matter, are out of scope for this chapter. This chapter focuses on the use of factor analysis, and can be seen as a guide to use factor analysis to reduce the total amount of variants in a product. Therefore, the aim of this chapter is to enable the reader to use factor analysis on basically any piece of software. It also gives an overview as how the results from the use of factor analysis can be interpreted to successfully reduce the number of variants. The complete framework, preparation of ma-

terial for factor analysis, the actual analysis, and post decision support is called FAM. For background information on factor analysis see Johnson and Wichern [146]. Background information on the numerical methods used to compute the factor analysis can be found in Golub's Matrix Computation [186]. Theoretical background will only be provided where it is absolutely necessary in order for the reader to understand the steps that have to be taken when using FAM. Yet, the steps to perform FAM are complete in a way that the reader is enabled to use FAM without further information. That is, the math presented in this chapter is needed to calculate the results, but Step 1 through 5 can be automated using simple tools like Excel.

FAM, as shown in this chapter, is based on factor analysis, but contains steps that are not originally part of factor analysis. Thus FAM consists of seven steps to get to a possible solution of the variant problem (each explained and exemplified in detail below):

- In Step 0, the input for the factor analysis is described. This is done to ensure that the reader can design their data to fit (can be input to) FAM easily.
- In Step 1, the Variance-Covariance Matrix is set up based on the input from Step 0 to perform the factor analysis.
- In Step 2, the eigenvalues and eigenvectors of the matrix are calculated. The eigenvalues can be thought of as the weight for each factor indicating the amount of information this factor holds.
- In Step 3, the eigenvalues and eigenvectors are sorted.
- In Step 4, the factor loading matrix is calculated providing the factors. These factors represent the possible variants.
- In Step 5, the number of factors (and therefore number of new variants) will be calculated.
- In Step 6, the factors are analyzed and it is assessed whether they qualify to be variants in reality.
- In Step 7, possible ways to continuously control the success of this method are presented.

FAM

The Steps 1 to 5 are related to pure factor analysis. Steps 0, 6 and 7 are additional steps that have to be performed as a part of FAM to avoid the risk of factor analysis cancelling out input dimensions.

At a glance, FAM seems to be complex, however it should be noticed that steps 1-5 have been completely automated, and done without human interaction (and they can be performed by almost any statistical software, or with MS Excel with an add-in, as in our exam-

ple). Step 6 and 7 need human interpretation, but these steps are mostly reduced to the interpretation of descriptive statistics (figures, see e.g. Figure , Figure , Figure).

Below, each step of FAM is explained in detail for the purposes of transparency of the method.

5.3.1 FAM – Step 0: Prerequisites and data collection

Most projects at Daimler AG are evolutionary. That is, the products developed are based on predecessors, and the functionality of the variants has been growing for years, taking the previous variants, with their functionality as a starting point, and addition new functions and variants for each project. Normally, several meetings are held with function responsibles, where new functions for the project are "collected". Functions [121], [144] in this context are, for example, "fuel injection". This function depends on different input parameters, such as the pedal position, the current speed, and the selected gear. It also gets input from systems such as the electronic stability program that might overrule the pedal position. This is the granularity of functions examined using FAM.

As can be seen in Table 26, the example given here is based on a function development plan from the automotive domain. This plan contains the functions of the software project and connects them to the variants that exist in the software at present. Therefore, the following format is the basis:

- Different functions in rows
- Different variants in columns
- The functions for each variant have to be selected (denoted by setting the cell to "1")
- There could possibly be another column, indicating for example the costs for a specific function (e.g. number of days to implement this function).

The first action to be taken is to set "1" for all selected functions and "0" for all unselected functions. It is also possible to change the selected functions into costs. Observe that the factor analysis has to be performed on numerical data.

	Engine Platform 1			Engine Platform 2		
	NAFTA	Europe	Japan	NAFTA	Europe	Japan
Function 1 to n	1	1	0	0	0	0
	0	1	0	0	1	1
	0	0	1	0	1	1
	0	1	0	0	0	0
	0	0	0	1	1	0
	1	0	1	0	1	1
	0	1	0	1	0	0
	1	0	0	1	1	0
	...	...	...	...	...	...

Table 26: Function Plan

A random matrix to compute the results is set up. The variants in this example will be grouped in two different engine platforms for heavy duty (Engine Platform 1, EP1) and medium duty (Engine Platform 2, EP2) engines. Several emission standards exist for each of these engines. In this example, Europe, Japan and NAFTA (North American Free Trade Agreement) are considered. Two engine platforms, comprising three markets each, make a total of six variants that will be examined in this chapter.

While there have been reasons to set up these variants in the past, there was a need to reduce the complexity of the product. Therefore, the variants needed to be examined to see whether there were potential candidates for merger.

The quality of the output is directly dependent on the quality of the data put into the Function Plan (Table 26). One thing a potential user of FAM has to decide is the level of detail of the functions. Of course, functions can be split up in sub-functions or grouped in function units. This influences the result, and the ideal amount of functions is not easy to find but has to be subject to expert opinion relating to what is appropriate for the case at hand.

There is also no weighting of special functions, as this would add huge complexity to data interpretation, as will be evident later.

The correlations between these markets and engines as well as the identification of variants that can be combined are the focus from now.

5.3.2 FAM – Step 1: Calculate VCV matrix

The Variance-Covariance matrix (VCV) is computed on the function plan described in Section 5.3.1. The Variance-Covariance matrix has the variances (all variances may be standardized to 1) on the diagonal and, as off-diagonal entries, the covariances of Variant i and Variant j in row i and column j. Therefore, it is easy to see that this matrix is symmetrical, as the covariances of Variant i and Variant j can be found in row i and column j as well as in row j and column i. The covariances are the entries that merit attention as they contain the correlation between two different variants. This is especially true if the variances are standardized to 1. Positive covariances indicate positive correlation, negative values indicate negative correlation. So, the values close to 1 indicate two variants that are closely related and values close to -1 indicate variants that do not at all belong together. If, for example, the correlation is close to 1 between the two Variants i and j, it is very likely that if Variant i contains a specific function, the same function can be found in Variant j. If the correlation is close to -1, it is very likely that Variant j will not contain this function. That will give a first insight into the problem.

First, the function plan can be standardized. This standardized matrix will be called Θ with entries θi. The standardization is done by subtracting the mean μi and dividing by the standard deviation σi for each of the six variants and each function within the variant according to:

$$\theta_{i,j}^{new} = \frac{\theta_{i,j}^{old} - \mu_i}{\sigma_i}$$

Once this standardization is done, computing the Variance-Covariance matrix Σ is done by multiplying this matrix in the following way:

$$\Sigma = \frac{1}{n}\Theta^T\Theta$$

Thus, it can be verified that Σ is a symmetric real matrix with real non-negative eigenvalues and orthogonal eigenvectors (as can be seen in [186]). To compare the covariances, the Σ matrix can be standardized. This means that all the variances on the main diagonal will be equal to 1, and therefore the basis for the covariances are the same. This is done by computing the coefficients of correlation ρ according to:

$$\rho_{i,j} = \frac{Cov(V_i, V_j)}{\sqrt{Var(V_i)Var(V_j)}}$$

Where the $\rho_{i,j}$ are the entries of Table 28, $Cov(V_i,V_j)$ are the covariances of Variant i, and Variant j and $Var(V_i)$ is the variance of Variant i. The variances and covariances are given in Table 27. Working on a standardized matrix Θ, the diagonal will only contain the value 1 as can be seen in Table 28.

Variance-Covariance Matrix					
0.33	0.17	0.00	0.50	0.00	0.50
0.17	**0.50**	1.00	0.50	0.33	1.00
0.00	1.00	**4.50**	0.00	1.00	4.50
0.50	0.50	0.00	**1.50**	0.00	0.00
0.00	0.33	1.00	0.00	**1.33**	1.00
0.50	1.00	4.50	0.00	1.00	**6.00**

Table 27: Variance-Covariance Matrix

Covariance Matrix					
1.00	0.41	0.00	0.71	0.00	0.35
0.41	**1.00**	0.67	0.58	0.41	0.58
0.00	0.67	**1.00**	0.00	0.41	0.87
0.71	0.58	0.00	**1.00**	0.00	0.00
0.00	0.41	0.41	0.00	**1.00**	0.35
0.35	0.58	0.87	0.00	0.35	**1.00**

Table 28: Covariance Matrix

The Covariance matrix will be used as an input for the factor analysis. For further insight into the covariance matrix and the relations between the variants, eigenvalues and eigenvectors have to be computed.

5.3.3 FAM – Step 2: Calculate eigenvalues and eigenvectors

The second Step in FAM is to compute these eigenvalues. As eigenvalues and eigenvectors are constantly used in this chapter, and are a vital part of the method, definitions on both will be provided to ensure a common understanding on this key element. The definitions will give the mathematical way to calculate eigenvalues.

Definition eigenvalue:

Let A be a square matrix of dimension $k \times k$ and let I be the $k \times k$ identity matrix. Then the scalars $\lambda_1, \lambda_2, ..., \lambda_k$ satisfying the polynomial equation

$$|A - \lambda I| = 0$$

are called eigenvalues [146], [186].

Definition eigenvector:

Let A be a square matrix of dimension $k \times k$ and let λ be an eigenvalue of A. If x is a non-zero vector of dimension $k \times 1$ such that

$$Ax = \lambda x$$

then x is said to be an eigenvector of the matrix A associated with the eigenvalue λ [146], [186].

However, in reality it is extremely expensive and sometimes impossible to calculate eigenvalues [187]. Therefore, numerical approximations are used to compute the eigenvalues. The quality of the approximation is proved in Golub's Matrix Computation handbook [186].

MS Excel was used in this example, however an algorithm for Excel has to be written that computes the eigenvalues of a real symmetric matrix. The function uses the Jacobi fast rotation, which is a fast and stable method to approximate eigenvalues [188], [189]. Information on the algorithm can be found in [186].

The function zeros out two off-diagonal elements of a given matrix Σ per calculation using rotations matrices Q. This rotation has to be applied iteratively to all off-diagonal entries as shown below:

$$\begin{aligned}
&\Sigma \mapsto Q_{kl}^T \Sigma Q_{kl} = \Sigma^{(1)} \\
&\Sigma^{(1)} \mapsto Q_{op}^T \Sigma^{(1)} Q_{op} = \Sigma^{(2)} \\
&\vdots \\
&\Sigma^{(n-1)} \mapsto Q_{qr}^T \Sigma^{(n-1)} Q_{qr} = \Sigma^{(n)} = \Phi
\end{aligned}$$

This function can be used to compute a diagonal matrix, zeroing out[14] the off-diagonal elements and therefore giving the eigenvalues on the diagonal. The result of this function is a diagonal matrix Φ with the eigenvalues as diagonal entries as shown in Table 29. This will not change the dimensions of the matrix but only the values. That is why it is of the same dimension as the covariance matrix in Table 28.

Eigenvalue Matrix					
0.00	0	0	0	0	0
0	0.09	0	0	0	0
0	0	0.79	0	0	0
0	0	0	1.85	0	0
0	0	0	0	1.18	0
0	0	0	0	0	10.27

Table 29: Eigenvalue Matrix

Next action in this step is the calculation of the eigenvectors. As there is no built-in-function in Excel, an algorithm has to be written implementing the Jacobi algorithm again. The calculation is quite similar as above and returns a matrix Ξ, shown in Table 30, with the eigenvectors as columns in the same dimension as the input VCV matrix in Table 27.

Eigenvectors					
0.87	-0.03	-0.34	0.32	-0.16	0.04
0.00	0.92	0.15	0.30	0.14	0.15
0.29	-0.21	0.66	-0.09	0.17	0.63
-0.29	-0.32	0.13	0.89	0.03	0.01
0.00	-0.10	-0.41	-0.01	0.89	0.16
-0.29	0.02	-0.49	-0.01	-0.35	0.74

Table 30: Eigenvector Matrix

5.3.4 FAM – Step 3: Sort eigenvalues and eigenvectors

The eigenvalues show the influence of the factor calculated on the total amount of variability observed in the function plan. That is why the eigenvalues have to be put in a decreasing order, so the first factor is the one indicating the biggest overlap, and the last factor is the one indicating the least overlap between variants. As the eigenvectors given in Table 30 be-

[14] This means reducing all off-diagonal entries to zero. This can be done by Gaussian Elimination [186].

long to specific eigenvalues given in Table 29, they have to be sorted according to the eigenvalues. This means, that the eigenvector belonging to the largest eigenvalue will be put in the first position, indicating that this eigenvector and the factor that is calculated from it, contains the most information. For example, as can be seen in Table 29, 10.27, which is the largest eigenvalue, is located in the last column. Therefore, the eigenvector in the last column of Table 30, with the entries 0.04 to 0.74, belongs to this eigenvalue. Table 31 shows the sorted eigenvalues and their eigenvectors. The eigenvalues can be found in the first row, the related eigenvectors in the columns below their eigenvector. That is, the largest eigenvalue, 10.27, is in the first cell in the first column of Table 31. The column below this eigenvalue gives the related eigenvector, 0.04 to 0.74. This sorting is done for all the eigenvalues with their corresponding eigenvectors.

These sorted matrices are now called Φ^{sort} containing the sorted eigenvalues and Ξ^{sort} containing the sorted eigenvectors.

By using these sorted matrices, as see in Table 31, as input for the factor calculation in Step 5 (Section 0), the first factor will be responsible for most of the variability found in the function plan from Table 26. This sorting allows better interpretation of the factors. Without the sorting, the factors would be the same, but the ordered output enables a better overview.

Eigen-values	10.27	1.85	1.18	0.79	0.09	0.00
Eigen-vectors	0.04	0.32	-0.16	-0.34	-0.03	0.87
	0.15	0.30	0.14	0.15	0.92	0.00
	0.63	-0.09	0.17	0.66	-0.21	0.29
	0.01	0.89	0.03	0.13	-0.32	-0.29
	0.16	-0.01	0.89	-0.41	-0.10	0.00
	0.74	-0.01	-0.35	-0.49	0.02	-0.29

Table 31: Sorted eigenvalues and eigenvectors

5.3.5 FAM – Step 4: Calculate factor loadings matrix

The fourth Step is the calculation of the loadings matrix Λ. This matrix is the main result of the factor analysis and displays the connection between variants. Therefore, this matrix can be used to decide which variants should be combined into one new variant or factor. This matrix can be calculated by multiplying the eigenvector matrix Ξ^{sort} with the eigenvalue matrix Φ^{sort} in the following way:

$$\Lambda = \Xi^{sort}\sqrt{\Phi^{sort}}$$

This completes the calculation of the factor analysis and leaves the discussion of the results. Now the loadings matrix, as can be seen in Table 32, has the desired factors as columns numbered 1 to 6. The loadings for each factor are explained as the six variants for the engine platform and emission standards given in rows. Each factors can be seen as a new potential variant. For clarity reasons these potentially new variants will be called NewVariant from now on (as they can be the result of a merger of two of the old variants). The intention in the beginning was to reduce the dimensions (six in the beginning, one for each variant) and keep as much information as possible. The information kept can be seen as the amount of functions covered by the now fewer NewVariants. In Table 32, the factor loadings are shown. The factor loading matrix should be read in the following way:

- The factors (potentially NewVariants) are given in columns.
- The old variants are given in rows.
- Each value λi,j gives the strength of each old variant in each factor.
- In Factor 1 the value for EP2 Japan, 2.38, is five times the value of EP1 Europe, 0.47. This indicates, that EP2 Japan is a very strong part of Factor 1. So, high positive numbers indicate strong affiliation of a variant to a factor.
- Negative numbers, like EP2 Japan in Factor 3, which is -0.38, indicate that a variant contains other functions than that factor.
- Numbers close to zero like EP2 Europe in Factor 3 indicate that this variant is not an essential part of the factor.
- In this example, Factor 1 consists of EP1 Europe, EP1 Japan, EP2 Europe and EP2 Japan. Factor 2 consists of EP1 NAFTA and EP2 NAFTA.
- The eigenvalues are given in the row below the factors and indicate that these eigenvalues give the amount of influence of that factor in total.
- Finally, the percentage of information, as well as the accumulated amount of information is given.

The information in each factor is measured by the value of the eigenvalue belonging to the eigenvector that was used to compute the factor loadings. Now, it can be computed how much information each factor contains (last two rows in Table 32). For example, Factor 1 explains 73% of the variability, Factor 2 explains 13%, and so on. Therefore, the first two factors explain 86% of the total information as can be seen in the last row of Table 32 where the accumulated amount of information is given. Using these two should be enough for the example project, thus only these are chosen for continued analysis.

		Factor (NewVariant)					
		1	2	3	4	5	6
EP1	NAFTA	0.13	0.44	-0.18	-0.30	-0.01	0.00
	Europe	0.47	0.40	0.16	0.14	0.27	0.00
	Japan	2.03	-0.12	0.18	0.59	-0.06	0.00
EP2	NAFTA	0.03	1.21	0.03	0.12	-0.09	0.00
	Europe	0.51	-0.01	0.97	-0.36	-0.03	0.00
	Japan	2.38	-0.02	-0.38	-0.43	0.01	0.00
Eigenvalues		10.27	1.85	1.18	0.79	0.09	0.00
Amount of information		73%	13%	8%	6%	1%	0%
Accumulated information		73%	86%	94%	99%	100%	100%

Table 32: Factor loadings and explained variance

To study possible NewVariants, the highest factor loadings in the first two factors have to be identified. The first potential NewVariant (Factor 1) could be constructed of variants EP1-Europe (0.47), EP1-Japan (2.03), EP2-Europe (0.51) and EP2-Japan (2.38). The second NewVariant (Factor 2) could be built out of variants EP1-NAFTA (0.44) and EP2-NAFTA (1.21). As can be seen in Factor 1, there is a discrepancy between the factor loadings ranging from 0.47 to 2.38. How to handle factors with a wide range of factor loadings will be explained in Section 5.3.7. Two important things can be seen in this example:

- The two potential NewVariants are distinct. That is, none of the old variants overlap the two NewVariants.
- All of the six old variants are contained in the two NewVariants.

In this project, these were major constraints. If an old variant appears in more than one factor, there would not be a significant reduction in work. Also, the new factors had to cover all the markets, and engine platforms, as did the original six variants.

These factors are calculated mathematically to discover the best possible arrangement of functions. Observe however, these NewVariants do not have to make any obvious sense

from a conceptual perspective. The functions are just grouped in a way they are needed in relation to overlap. For example, if a user needs the features “SUM” and “AVERAGE” from Excel along with the automated “CONTENT” feature in Word, FAM would group exactly these features into one factor (as their functional content overlaps), and therefore into one NewVariant. This NewVariant, however, does not need to make sense for a user or manager; rather the reasoning for assigning them to the same NewVariant is the overlap in underlying functionality (and not any human readable logic).

The factor analysis results obtained so far still needs some interpretation. The main question is how many factors must be kept in order to display as much variability as possible, while still keeping the smallest possible number of NewVariants. A potential user of FAM could also decide that for example 86% is not enough information. Then, a third factor would have to be set up and the factors would be constructed differently. Whether or not it makes sense to split e.g. Factor 1 into two NewVariants has to be evaluated, as the factor loadings clearly differ. This, however, depends on the data set. In our example, the two factors are kept. One reason for this is an attempt to keep the example simple. In addition, there is real value in calculating factors (as seen in Table 32), and not only a correlation matrix. The main difference between the two is that the factors not only identify highly correlated variants, as the correlation matrix does, but also make sure that the newly found NewVariants (or factors) are also not correlated themselves, i.e. variants used in one factor are also not correlated to variants used in another factor.

Still, there is a need for further analysis to offer additional decision support to decide how many factors to keep, as illustrated in the next step of FAM.

5.3.6 FAM – Step 5: How many factors to keep

To interpret the factors calculated in Step 4, it might help to imagine the data as n-tuple in a high dimensional space. That is, every line in the function development plan in Table 26 is a point in a n-dimensional space where n is the number of variants. With this in mind, the number of dimensions is reduced by FAM. Therefore, the points now are in an m-dimensional space with m<n. However, to interpret the data, the axis of this m-dimensional space might have to be orthogonalized, as described in Section 5.3.6.1. This rotation does not change the meaning of the data, it just simplifies the interpretation of the results. By orthogonalizing the axis, the influence between the different dimensions is reduced. And the factors will be much easier to select. After orthogonalizing, the number of factors has to be determined, which is shown in Section 5.3.6.2.

5.3.6.1 Rotation

There is also the possibility to rotate factors. This rotation can be done by Varimax rotation, which is the most common rotation used in combination with factor analysis. Information about the mathematics behind this method and also additional information on other rotation methods can be found in [146]. The rotation of the factors simplifies the identification of the factors as it orthogonalizes the factors and thereby eliminates the influence between factors. In the example in this chapter, the factors are easily visible. Therefore, there is no need to rotate them. In other cases, the factors might not be as easy to identify. Then the rotation is one method to illustrate the factors in a different way to identify NewVariants.

5.3.6.2 Number of Factors

Unfortunately, there is no formula to decide how many factors to choose. There have been many approaches helping users to decide on the number of factors. The two methods mostly used, the Kaiser Guttman rule, and the scree plot, will be described here [190], [191]. The Kaiser Guttman rule is probably the easiest and therefore widely used. As the main purpose of factor analysis is the reduction of data, only factors that have more impact than the underlying input are kept. Thus, factors with eigenvalues that are larger than 1 are kept, factors with eigenvalues smaller than 1 are dismissed. In this example, there would have been three factors as the third eigenvalue is 1.18, the fourth is 0.79 as can be seen in Table 31. So, three eigenvalues are larger than 1 and therefore three factors should be taken into consideration according to the Kaiser Guttman rule.

Another way to decide the number of factors is the scree plot[192]. To identify the number of NewVariants, the eigenvalues have to be plotted in an eigenvalue diagram or scree plot. The eigenvalues are plotted in decreasing order as shown in Figure 33.

If there is a convex bend (elbow) in the plot, the eigenvalue at that point of the bend should be the last to include as the information gained from additional factors is small. In this case, the convex bend appears after the second factor. As can be seen in Table 32, the first two factors explain about 86% of the variability. That fits the results from the scree plot as it confirms the first impression that the first two factors are the most important ones. That might help in finding the right number of factors.

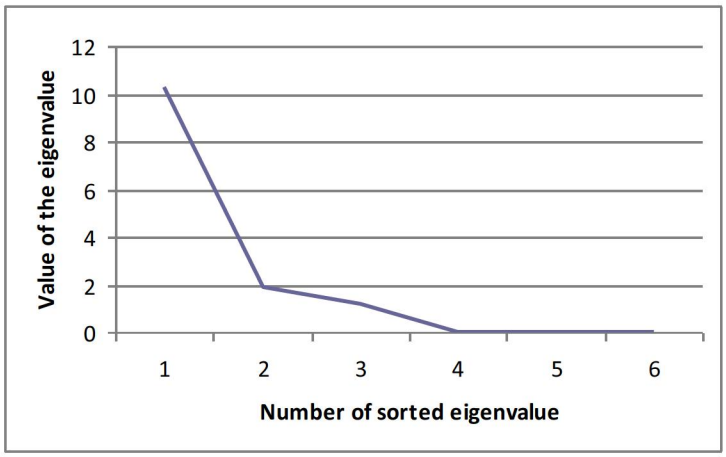

Figure 33: The convex bend in the Scree Plot

In this case, the scree plot was used to identify the number of two factors. However, using the Kaiser Guttman rule, three factors would have been taken. This shows that different rules can lead to a different number of factors chosen. The question that has to be asked is "how many factors really make sense in this project"? An example of a possible way to change the number of NewVariants after identifying the number of factors will be given below.

5.3.7 FAM – Step 6: Analysis of the factors

This test is an addition to the factor analysis described above, and it is usually performed before the final setup of the NewVariant structure. Remember, the factor analysis will output a number of factors that might be good choices for NewVariants. So, the situation might be described as follows. Let the input be six (possible) variants, and the number of factors be two, as described in the example. Then, the situation might look like Figure 34.

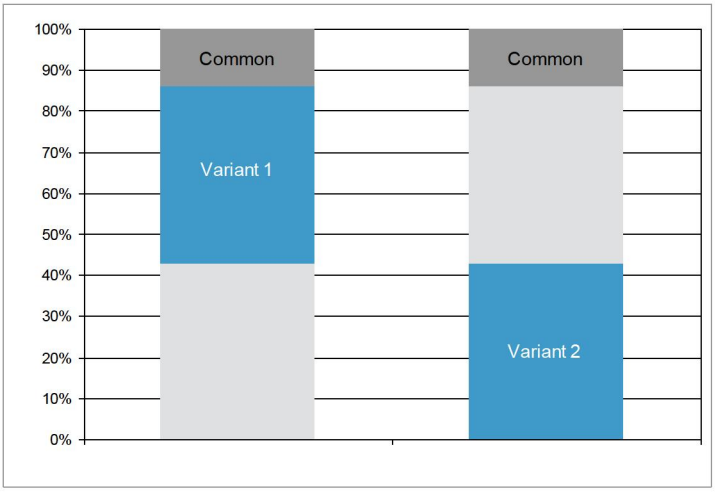

Figure 34: Location of NewVariants

The dark grey part is the part of functions that is not part of the two NewVariants. Remember, that for example the two factors combined do not cover the whole number of functions but, in this example, 86%. That leaves 14% as a Common part that needs to be implemented for each variant. In Figure 34, this is the Common part at the top of both bars that does not belong to either NewVariant.

Implementing the NewVariants like this, there are two possibilities for the location of desired functions. They can be located within the scope of one of the existing NewVariants, or they can be located across more than just one NewVariant as illustrated in Figure 35.

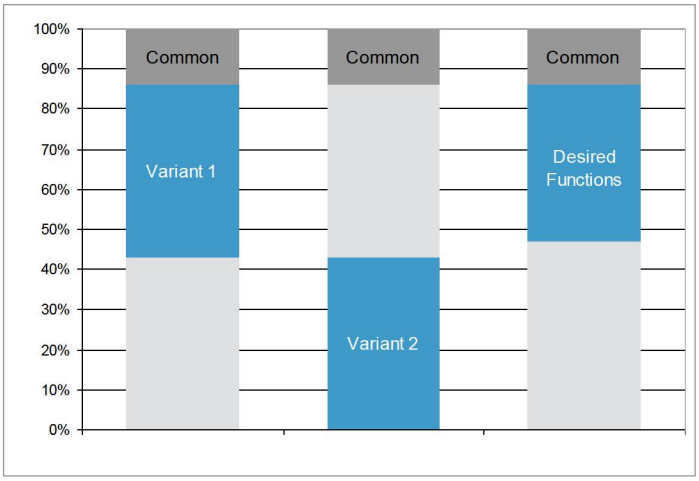

Figure 35: Location of a third NewVariant with respect to the first two NewVariants

If the scenario illustrated in Figure 35 appears, all efforts have paid off and NewVariant 1 can be used to implement the desired functions. The same applies if all desired functions are located in the scope of NewVariant 2.

In the scenario of Figure 36, the desired functions overlap the functions of NewVariants 1 and 2. That means an additional NewVariant has to be created. If this is the case it would most likely be visible in the factor loadings. If the relevant factor loadings overlap significantly in some parts, then an alternative “new” NewVariant may be most effective, where the desired functionality (not covered by NewVariant 1 or NewVariant 2 alone) is retained. As this example is not a real test, a little additional analysis has to be performed in order to identify exactly this situation.

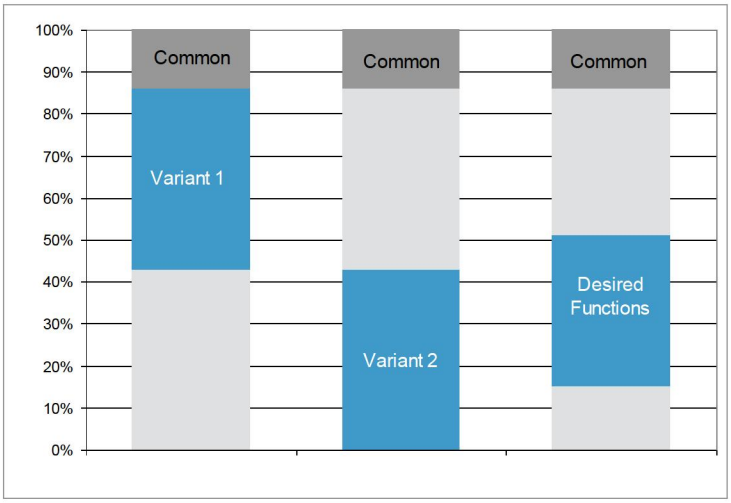

Figure 36: Different scenario of the location of the third NewVariant

Looking at Table 32, there has been a wide range in the factor loadings within Factor 1. As mentioned in Section 5.3.4, a closer look has to be taken to verify the factor. To do so, the overlapping between the variants in Factor 1, EP1-Europe (0.47), EP1-Japan (2.03), EP2-Europe (0.51) and number EP2-Japan (2.38) has to be computed as the differences between the factor loadings are quite high. The overlapping between the variants can easily be computed from the function plan. The output will be a percentage indicating how many functions are actually part of all the examined variants.

	Overlapping functions	Overlapping with respect to costs
EP1 Europe EP2 Europe	64%	79%
EP1 Japan EP2 Japan	67%	84%

Table 33: Overlapping between variants

It is better to split the first NewVariant into two NewVariants (see Table 33). The variants EP2-Europe and EP1-Europe overlap with 64% if the number of functions is counted. Also, as mentioned in Section 5.3, weighting of functions is not easy to interpret in the factor analysis. In this step, however, weightings can be taken into account. One way to do so is calculating the cost of the functions and weight the functions with their cost. Cost can be e.g. effort, as long as it is a number where high indicates big effort, and low indicates small effort, to develop the function. If the costs for each function are considered, the overlapping is even higher at e.g. 79%. That is why these two variants can be combined into one NewVariant.

The two other variants in Factor 1, EP2-Japan and EP1-Japan have a similarly high overlapping of 67% (and with cost e.g. 84%). These two variants can also be combined into yet another one NewVariant. This would change the result from Section 5.3.6, where there were two NewVariants after the factor analysis. However, the results shown in Figure 36 indicate that there is a good reason to split up Factor 1. Especially when looking at the overlappings after considering the cost of each function.

This completes the factor analysis part. Three NewVariants have been identified, EP1 Europe and EP2 Europe can be combined, EP1 Japan and EP2 Japan can be combined and EP1 NAFTA and EP2 NAFTA can be combined. That reduces the total amount of variants by 50%, and illustrates how variant reduction can be achieved by the use of decision support material acquired through the use of FAM. In addition, in the example the variants can be combined to create one NewVariant per market (Europe, Japan, NAFTA) over Engine Platform boundaries (heavy duty, medium duty).

5.3.8 FAM – Step 7: Continuous monitoring

After completing the FAM steps, a first leap towards the reduction of variants was taken. However, as projects in the automotive domain are quite long, projects could benefit from continuous monitoring, and variant reduction. The main advantage of this is to put measures into place, monitoring the creation of new variants, and giving continous indicators to management as to the possibilities of variant merger.

Three main things can be a part of continuous monitoring of development efforts (e.g. projects), Variant control, Overlapping control and Factor control.

5.3.8.1 Variant Control

The first and obvious way to show the success of the work is keeping track of the number of variants in the software project. If the number of variants decreases, the work was successful. However, this index is very slow. Especially in long term projects, this might not be the right measure for (short term) success. This number will be kept track of nonetheless, as this indicates the overall goal of variant control and reduction.

5.3.8.2 Overlapping control

A much better way to monitor the short term success is to compare the overlapping of (old) variants within (new) factors. Calculated for each factor, this number ω_i indicates the fit of the Factor i. As $0 \leq \omega_i \leq 1$, this number can also be calculated for the complete project :

$$\kappa = \prod_{i=1}^{n} \omega_i$$

$$0 \leq \kappa \leq 1.$$

For example, if the variants illustrated in Table 33 should be put under control, the overlapping at two different development stages could be compared. as can be seen in Table 34.

	First date (κ_1)	Second date (κ_2)
EP1 Europe EP2 Europe	64%	70%
EP1 Japan EP2 Japan	67%	78%

Table 34: Overlapping control over time

In this example, κ_1 indicates the control measurement at the first date, κ_2 indicates the second date. The ω_i, indicating the overlapping between variants, is given in Table 34. For this example, κ_1 and κ_2 are calculated:

$$\kappa_1 = 0.64 * 0.67 = 0.4288 < 0.546 = 0.7 * 0.78 = \kappa_2$$

As can be seen, κ_1 is smaller than κ_2. This indicates that there has been a positive development as the increased overlapping control indicates a higher overlapping in the project. This is the right direction towards combining the old variants.

κ indicates the fit of factors for the complete project. The closer the number gets to 1, the better the fit. Therefore, the indicator can be monitored and indicates success if increasing, and potential problems if decreasing.

5.3.8.3 Factor control

Factor loadings can be kept to achieve factor control by continuously performing multiple factor analysis along the way. Keeping track of the factor loadings, enables to identify changes in the fit of the factors. If there are fundamental changes in the analysis (different number of factors, other groups within the factor, etc.), this indicates a potential problem within the project, and a new FAM instance can be performed.

Table 35 gives an example of what a new factor loadings matrix may look like. Comparing this matrix to the last factor loadings matrix, in Table , gives some insight in to the latest development practices. It can be seen that Factor 1 and Factor 2 still cover the same markets. This is no real surprise as the software development now takes place with the knowledge of

the new factors (or NewVariants of the future). Therefore, new functions can be implemented to fit this new structure. In Factor 1 (Table 35) compared to Factor 1 in Table 32, all relevant factor loadings are bigger (e.g. EP2-Japan from 2.38 to 2.44), which is good as it means that this factor now contains even more information as it did before. In Factor 2, however, EP2-NAFTA decreased from 1.21 to 1.03. This is not a good sign and the reasons for the decrease have to be analyzed. The total weight of this factor increased from 13% to 17%, which indicates that there is more information contained in this factor as compared to Table 32.

In this example, the structure of the variants has not changed. The factors still include the same variants as in Table 32. This, however, does not always have to be true. If, for example, EP1-NAFTA switched into Factor 1 and EP1-Europe switched into Factor 2, this might indicate problems. There has to be a rather stable variant structure and not a constantly completely changing variant structure. This is the case in long-term development projects at Daimler Trucks.

		Factor					
		1	2	3	4	5	6
EP1	NAFTA	0.11	0.41	-0.16	-0.11	-0.06	0.00
	Europe	0.57	0.31	0.42	0.16	0.23	0.00
	Japan	2.02	-0.17	0.12	0.71	-0.26	0.00
EP2	NAFTA	0.07	**1.03**	0.28	0.02	-0.21	0.00
	Europe	0.63	0.01	0.92	-0.63	-0.01	0.00
	Japan	**2.44**	-0.01	-0.22	-0.12	0.02	0.00
Eigenvalues		11,29	2,61	0,82	0,58	0.05	0.00
Amount of information		73%	**17%**	6%	4%	1%	0%
Accumulated information		73%	90%	96%	99%	100%	100%

Table 35: New Factor scores

The overall point is that the use of FAM enables monitoring of the development continuously. This is especially relevant for long-term projects, typical for the automotive domain,

where distributed development makes monitoring, as well as the use of informal process for this purpose, very hard [145].

5.4 Conclusions

The use of FAM enables an explicit mapping and analysis of variants. The results can be used as decision support for managers and the development organization. The illustration of variant overlap, possibilities for variant split or merger helps to attain the ultimate goal of reducing the overall number of variants within the software offering.

FAM uses the present code base (present variants) as input, thus in combination with the automation of Steps 1-5, lowers the use threshold and increases the usability of the method [145], [193] . The level of detail in which the method is described in this chapter is and should be black boxed for users of the method in industry. However, tranparancy was opted for in this chapter for peers to judge the usability and usefulness of the method.

The possibilities of using FAM at Daimler AG came with reduced cost for maintenance, reduced costs in identifying and removing bugs, reduced complexity and therefore better handling of the software. This also led to increased quality and faster development times.

However, FAM is not a perfect solution. FAM is suited best for larger software development efforts with a significant amount of variants, as the method only identifies the overlapping between variants but does not create new, ideal NewVariants. That is why there has to be quite a large number of variants already, a situation not uncommon in many companies. In addition, the use of FAM produces decision support material, and not automation of variant reduction, which was never the goal.

It should be observed that the examples and analysis performed in this chapter are based on fictive (random) numbers, consequently the results are quite “black and white”. With random numbers, the correlations are expected to be much lower because real life function plans have much higher correlation between variants as these variants often arise from each other. The amount of identical or almost identical functions can be fairly high. That, however, will result in high correlations in the factor analysis, which means that the factors will even be stronger and a lot easier to identify, which should make the analysis of factors possible for merger easier than the ones exemplified in this chapter. For reasons of confidentiality, details regarding the analysis performed at Daimler AG cannot be shared beyond the example in the chapter.

Software development efforts at Daimler AG are generally run as long-time projects, which is why the project where FAM was introduced and used has not come to a completion as of

yet. However, reducing variants is a long-term goal. Therefore, measures to keep track of the success even in small parts are given along with the analysis itself in this chapter. The analysis in the example shows the ability to identify and continuously monitor variants and display possible ways to reduce the complexity in a software project.

5.5 Future Work

Today, FAM is supported by a MS Excel based tool (simple Excel sheet with add-in) for the automation of most of the calculations (see Step 1-5). In the future, improved tool support is planned, as well as the development of a standardized set of collection procedures for pre-project, initial, use of FAM, in earlier planning phases.

A central part of future research is also the explicit validation of FAM, and results will be available upon the completion of the present development project in about 2 years.

5.6 Connection to SODE

FAM is now completed and can be used. It has been designed to solve a specific problem at an industry partner and the validation has started. This is an input for SODE as it solves a real world problem that arises in distributed engineering.

In Figure 37 it can be seen that this solution has been grouped into *PA 3.2.1.1 Develop Alternative Solutions and Selection Criteria*. FAM is used as a solution within *PA 3 Engineering* to select and combine the right variants in a given software product.

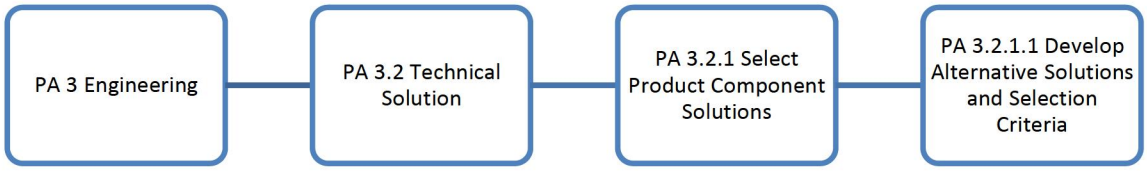

Figure 37: Sort FAM into SODE

Of course, the theoretical description of FAM in the last chapter can be used in SODE, but the Excel sheet with the macro calculating the factor analysis is of much greater value for practice as this can be used more or less out of the box. These files can then be added to the description file so that users can find both at the same time. This is a huge benefit for the usefulness of SODE.

6 Conclusions

6.1 Results

The goal of this work was to enable the industry partner, Daimler AG, to solve problems related to GSE. Especially in the Truck department, massively distributed work exists. Not only producing for but also developing in different markets brings huge complexity and challenges to daily work. The presence of several different brands within one company can also be quite challenging.

For these reasons, this work did not search for single solutions in problematic areas of distributed work as this would just be a drop in the ocean. As car/truck developing processes are extremely complex and time intensive, the idea was to take a more generalistic approach and create an add-on to these processes.

To ensure compatibility with Daimler AG and also supplier processes, widely-used and well-known process frameworks such as CMMI [120] and ISO 12207 [115] have been chosen as basis for SODE. Also, the solutions within SODE include but are not limited to Software Engineering.

The creation of a supporting framework has to be planned very well and conducted strictly according to this plan to ensure reproducible and meaningful outcome. The process that has been used to produce SODE can be seen in Figure 2 (see page 32). As this is a very high level process without details on the single steps, the detailed process for each step can be found in the chapters 2 to 4.

First, in Chapter 2, an initial version of SODE has to be set up using the input framework CMMI, ISO and others [115], [116], [120], [139]. In this initial setup, process areas that are related to GSE problems found in literature are combined in a framework. This new framework, SODE 0.2, is the first step towards a framework supporting industry with solutions relevant for their daily work in distributed teams.

To be able to provide relevant solutions, there has been a systematic literature review in Chapter 3. This review was used for the identification of solutions as well as of white spots within SODE. The solutions found in the review are relevant for industry as they support daily work by solving problems in distributed work. The white spots are also relevant for academia and research as they display potential areas of future work and further research. The combination of SODE 0.2 and the solutions identified in the literature review resulted in SODE 0.4.

Having combined structure (SODE 0.2) and contents (literature review) in SODE 0.4, it was important to find out whether SODE was usable and useful in practice. Therefore, a huge evaluation has been conducted and the results are presented in Chapter 4. This evaluation has two major steps, expert interviews and workshops. The expert interviews have been used to ensure the completeness and usefulness of SODE. The one-on-one interviews resulted in quite some improvement of SODE and also added additional solutions into SODE. The workshops have mainly been used to improve usability and usefulness of SODE. These also resulted in improved usability and added content to SODE. Overall, the evaluation has improved SODE and shown the need for a framework like SODE. After the implementation of the evaluation feedback, SODE 1.0 was released as main result of this work.

In Chapter 5, a solution for one of the white spots is presented. This solution was developed in close collaboration with the industry partner to solve the problem of superfluous variants in software products. This solution has then been sorted into SODE.

SODE 1.0 can be found in the appendix of this work as the chapters would have been incredibly long if every evolution of SODE would have been included. Therefore, the changes within each step are displayed and can then be followed and compared in the end result.

6.2 Future work

It is no surprise that this work was not able to solve all problems related to GSE and this was not the plan, either. There are unsolved problems in distributed teamwork and in GSE. But SODE is a good start as the evaluation in Chapter 4 showed. Industry as well as academia has been very pleased with the support of SODE as it tackles GSE problems in a way that has not been done before. Especially the systematic literature review gives SODE a head start providing real added value. The problem, however, is that there are still too many empty spots within SODE. Especially the industry partners in the evaluation stated the need for more solutions very clearly.

Therefore, the major task for future work is the completing of SODE with respect to the number and distribution of solutions. The structure itself has been rated very well fitting and appropriate. To develop missing solutions, industry and academia have to work together very closely as shown in Chapter 5. Though this is not a guaranty for the creation of solutions, it can help to ensure that the solutions are actually matching the problems in industry. Just as displayed in Figure 2 (see page 32), constant feedback loops with industry help to develop meaningful solutions.

For SODE, there are still two steps missing when comparing Figure 2 to the layout of this work. Due to time limitations, the dynamic validation could not be completed. However, the

dynamic validation is going on at an academical partner of Daimler AG, the University of Hanover. The first evaluation on an actual project will be done in a project at the University of Hanoverand SODE will be improved further during this project. After that, an evaluation in a real project at Daimler AG is planned. After that, there will be another refinement of SODE which should then result in a usable and useful SODE for practice. The main success factor for SODE is the constant input of solutions.

SODE can also be used to provide solutions not only for Software Engineering but also for other areas that work in distributed teams. This is already prepared in SODE as many solutions can be used on software engineering but are not limited to this area and can also be used independently of the area. The common denominator is distributed work, though, as the structure of SODE is based on problems that arise from distributed teams. Solutions for other engineering areas or basically any area that is working distributed across time-zones, geographical or cultural boundaries or in different organizations can be integrated in SODE. To do so, the solutions will probably have to be tagged to identify them for a specific area or as general solutions. This, however, can very easily be done as the solutions and the structure of SODE is already stored in a database. This would make SODE even more useful for practice as this would enable users to search for solutions beyond their own area and maybe get ideas to fill white spots within their own area.

Appendix A: Questionnaire A

Date: .....

Location: .....

Evaluation of SODE

(November 2010)

Stefan Schneider

Daimler AG

GR/PSP

Questionnaire

Level of knowledge (Please do NOT fill in): ☐ training ☐ used SODE

2.1. Do you have experience in GSE projects?

Expert					No experience
○	○	○	○	○	

Comments:

2.2. What is your position in the GSE project?

Strategic Management	Project Management	Daily Work	Some of my work	Research
○	○	○	○	○

Comments:

2.3. Do you have any previous knowledge of process models (e.g. CMMI, SPICE etc.)?

Expert					No experience
○	○	○	○	○	

Comments:

2.4. If you answered other than “no experience” in Question 2.3, do you think previous model experience is helpful to understand the model?

Absolutely ○ ○ ○ ○ ○ Not at all

Comments:

2.5 Do you think the training you received was necessary for you to be able to use SODE?

Absolutely ○ ○ ○ ○ ○ Not at all

Comments:

2.6 Do you think the training you received was necessary for you to be able to understand SODE?

Absolutely ○ ○ ○ ○ ○ Not at all

Comments:

2.7 Do you think the overall structure of SODE is logical and easy to understand?

Absolutely ○ ○ ○ ○ ○ Not at all

What can be improved?:

2.8 Do you think the structure of SODE is adequate to enable you to find solutions?

Absolutely ○ ○ ○ ○ ○ Not at all

Comments:

2.9 Do you think the fragmentation of SODE in three layers (Model, Solution item, Solution) is reasonable?

Absolutely ○ ○ ○ ○ ○ Not at all

Comments:

2.10 Did you recognize the visual separation of these layers?

Absolutely ○ ○ ○ ○ ○ Not at all

Comments:

2.11 Do you think the navigation within SODE is intuitive?						
Absolutely	○	○	○	○	○	Not at all
Comments:						

2.12 Do you think the navigation within SODE helps you to find the solutions that solve your problems?						
Absolutely	○	○	○	○	○	Not at all
Comments:						

Thank you!

Appendix B: Questionnaire B

Date: .....

Location: .....

Evaluation of SODE

(November 2010)

Stefan Schneider

Daimler AG

GR/PSP

Questionnaire

Preparation for this questionnaire:

- Think of a problem that you discovered in your daily GSE (Global Software Engineering) work. Please try to find a solution for this problem within SODE.
  Please give me a short description of your problem:

- Alternatively, you can look up a solution for the following problem:
  Imagine that you are a project manager for a project that develops software at two sites, Germany and India. Requirements are elicitated in Germany, only. The development teams are located in Germany and India. The problem now is that you have to decide where to test which part of the developed software.
  Try to find a solution for this problem within SODE.

Level of knowledge (please do NOT fill out): ☑ training ☑ used SODE

1.1. Do you have experience in GSE projects?

Expert					No experience
○	○	○	○	○	

Comments:

1.2. What is your position in the GSE project?

Strategic Management	Project Management	Daily Work	Some of my work	Research
○	○	○	○	○

Comments:

1.3. Do you have any previous knowledge of process models (e.g. CMMI, SPICE etc.)?

Expert					No experience
○	○	○	○	○	

Comments:

1.4. If you answered other than “no experience” in Question 1.3, do you think previous model experience is helpful to understand SODE?

Absolutely					Not at all
○	○	○	○	○	

Comments:

1.5 Was it easy to get an overview of the contents in SODE?

Absolutely ○ ○ ○ ○ ○ Not at all

What was the problem(s) (if any)?:

1.6 Did you have any problems navigating through SODE?

Absolutely ○ ○ ○ ○ ○ Not at all

What was the problem(s) (if any)?:

1.7 Can the solutions provided by SODE be used your problem-setting?

Absolutely ○ ○ ○ ○ ○ Not at all

What is missing?:

1.8 Do the provided solutions in SODE address problems in your projects?

Absolutely ○ ○ ○ ○ ○ Not at all

What is missing?:

1.9 Do the solutions in SODE cover all problems/challenges that you have encountered in relation to globally distributed work?

Absolutely ○ ○ ○ ○ ○ Not at all

How can it be improved?:

1.10 Are the provided solutions helpful for your projects?

Absolutely ○ ○ ○ ○ ○ Not at all

What was the problem(s) (if any)?:

1.11 How long did it take to identify/find a solution relevant to your problem in SODE (on average) (in minutes)?

1-5	6-10	11-15	16-20	>21
○	○	○	○	○

Comments:

1.12 To what degree do you consider this time reasonable in terms of time?

No problem ○ ○ ○ ○ ○ Much too long

Comments:

Thank you!

Appendix C: SODE 1.0

PA 1 Process Management

PA 1.1 Organizational Training

PA 1.1.1 Establish an Organizational Training Capability

- PA 1.1.1.1 Establish the Strategic Training Needs
- PA 1.1.1.2 Determine Which Training Needs Are the Responsibility of the Organization
- PA 1.1.1.3 Establish an Organizational Training Tactical Plan
- PA 1.1.1.4 Establish Training Capability

PA 1.1.2 Provide Necessary Training

PA 1.1.2.1 Deliver Training

Training of Testers

Hackett M IEEE	Hackett M	Building effective global software test teams through training

- PA 1.1.2.2 Establish Training Records
- PA 1.1.2.3 Assess Training Effectiveness

PA 1.2 Organizational Process Definition

PA 1.2.1 Establish Organizational Process Assets

PA 1.2.1.1 Establish Standard Processes

Process automation

Sengupta et al. 1998	Sengupta, K.; Leon Zhao, J.	Designing workflow management systems for virtual organizations: an empirically grounded approach
Baker et al. 2005	Baker, Paul; Loh, Shiou; Weil, Frank	Model-Driven Engineering in a Large Industrial Context — Motorola Case Study

Process reengineering

Sengupta et al. 1998	Sengupta, K.; Leon Zhao, J.	Designing workflow management systems for virtual organizations: an empirically grounded approach

Software Process Modelling and Improvement

Roehling ST et al. 5, no	Roehling ST; Collofello JS; Hermann BG; Smith Daniels DE	System dynamics modeling applied to software outsourcing decision support
Muhsin et al. 2008	Muhsin, Z. J.; El-Mousa, A. H.; Al-Taee, M. A.	A distributed workflow management system utilizing computational intelligence for decision support
Martins et al. 2006	Martins, Paula; da	SPI Methodology for

		Silva, Alberto	Virtual Organizations
	Casati et al. 2007	Casati, F.; Castellanos, M.; Salazar, N.; Dayal, U.	Abstract Process Data Warehousing
	Alho et al. 1998	Alho, Kari; Sulonen, Reijo	Supporting virtual software projects on the Web
	Clerc 2008	Clerc, Viktor	Towards architectural knowledge management practices for global software development
	Brechner 2005	Brechner, Eric	Journey of enlightenment: the evolution of development at Microsoft
	Fernandez A et al. 9, no	Fernandez A; Garzaldeen B; Grutzner I; Munch J	Guided support for collaborative modeling, enactment and simulation of software development processes
	Coleman et al. 2006	Coleman, Gerry; O'Connor, Rory	Software Process in Practice: A Grounded Theory of the Irish Software Industry
Business Process Outsourcing			
	Weihua Gan 2006	Weihua Gan	Analysis on the Costs of IT-Outsourcing
	Aksin OZ et al. 26, no	Aksin OZ; Masini A	Effective strategies for internal outsourcing and offshoring of business services: an empirical investigation
	Siva Namasivayam 2004	Siva Namasivayam	Profiting from business process outsourcing
	Rao et al. 2007	Rao, B.; Mulloth, B.	Building a Global Brand: The Case of Wipro
	Casati F et al. No	Casati F; Castellanos M; Salazar N; Dayal U	Abstract process data warehousing
	Agarwal R et al. 14, no	Agarwal R; Bajaj N	Managing outsourcing process: applying six sigma
	Casati et al. 2007	Casati, F.; Castellanos, M.; Salazar, N.; Dayal, U.	Abstract Process Data Warehousing
Model driven Engineering			
	Baker et al. 2005	Baker, Paul; Loh, Shiou; Weil, Frank	Model-Driven Engineering in a Large Industrial Context —

			Motorola Case Study
Establish agile Processes			
	Karlsson et al. 2000	Karlsson, Even-André; Andersson, Lars-Göran; Leion, Per	Daily build and feature development in large distributed projects
	Taylor et al. 2006	Taylor, Philip; Des Greer; Sage, Paul; Coleman, Gerry; McDaid, Kevin; Lawthers, Ian; Corr, Ronan	Applying an Agility/Discipline Assessment for a Small Software Organisation
	Muhsin et al. 2008	Muhsin, Z. J.; El-Mousa, A. H.; Al-Taee, M. A.	A distributed workflow management system utilizing computational intelligence for decision support
	Martins et al. 2006	Martins, Paula; da Silva, Alberto	SPI Methodology for Virtual Organizations
Define Processes			
	Macke et al. 2008	Macke, Dirk; Galinac, Tihana	Optimized Software Process for Fault Handling in Global Software Development
Knowledge transfer Process setup			
	Maurer et al. 2002	Maurer, Frank; Holz, Harald	Integrating Process Support and Knowledge Management for Virtual Software Development Teams
	Casey et al. 2006	Casey, Valentine; Richardson, Ita	Uncovering the reality within virtual software teams
	Anderl et al. 2008	Anderl, Reiner; Völz, Diana; Rollmann, Thomas	Knowledge Integration in Global Engineering
	Beyah et al. 2001	Beyah, G.; Gallivan, M.	Knowledge management as a framework for understanding public sector outsourcing
PA 1.2.1.2 Establish Lifecycle Model Descriptions			
PA 1.2.1.3 Establish Tailoring Criteria and Guidelines			
PA 1.2.1.4 Establish the Organization's Measurement Repository			
PA 1.2.1.5 Establish the Organization's Process Asset Library			
PA 1.2.1.6 Establish Work Environment Standards			
PA 1.3 Organizational Innovation and Deployment			
PA 1.3.1 Select Improvements			
PA 1.3.1.1 Collect and Analyze Improvement Proposals			
Communicate the Improvement to all remote parts of the			

team			
	Haag et al. 1997	Haag, Z.; Foley, R.; Newman, J.	Software process improvement in geographically distributed software engineering: an initial evaluation
Software Process Improvement			
	Coleman et al. 2006	Coleman, Gerry; O'Connor, Rory	Software Process in Practice: A Grounded Theory of the Irish Software Industry
PA 1.3.1.2 Identify and Analyze Innovations			
PA 1.3.1.3 Pilot Improvements			
PA 1.3.1.4 Select Improvements for Deployment			
PA 1.3.2 Deploy Improvements			
PA 1.3.2.1 Plan the Deployment			
PA 1.3.2.2 Manage the Deployment			
PA 1.3.2.3 Measure Improvement Effects			
PA 2 Project Management			
PA 2.1 Project Planning			
PA 2.1.1 Establish Estimates			
PA 2.1.1.1 Estimate the Scope of the Project			
Rules of thumb			
	2007		Empirical Laws and Rules of Thumb
	2007		Empirical Laws and Rules of Thumb
PA 2.1.1.2 Establish Estimates of Work Product and Task Attributes			
PA 2.1.1.3 Define Project Lifecycle			
PA 2.1.1.4 Determine Estimates of Effort and Cost			
Planning Models			
	Xu et al. 2007	Xu, Bin; Hu, Hua; Ling, Yun; Yang, Xiaohu; He, Zhijun; Ma, Albert	Achieving Better Collaboration in Global Software Design with Micro Estimation
	Singh et al. 2009	Singh, Yogesh; Bhatia, Pradeep Kumar; Kaur, Arvinder; Sangwan, Omprakash	Application of Neural Networks in Software Engineering: A Review
	Helander et al. 1998	Helander, M. E.; Ming Zhao; Ohlsson, N.	Planning models for software reliability and cost
	Xu et al. 2007	Xu, Bin; Hu, Hua; Ling, Yun; Yang, Xiaohu; He, Zhijun; Ma, Albert	Achieving Better Collaboration in Global Software Design with Micro Estimation
Rules of thumb			
PA 2.1.2 Develop a Project Plan			
PA 2.1.2.1 Establish the Budget and Schedule			

Distributed Project Planning Model		
Setamanit et al. 2006	Setamanit, Siri-on; Wakeland, Wayne; Raffo, David	Planning and improving global software development process using simulation
Zhenyuan Liu et al. 2008	Zhenyuan Liu; Yanqi Zheng	Resource-constrained multiple projects scheduling with the objective of minimizing activities cost
Helander et al. 1998	Helander, M. E.; Ming Zhao; Ohlsson, N.	Planning models for software reliability and cost
PA 2.1.2.2 Identify Project Risks		
Planning Models Externally developed software is a risk		
Sakthivel 2007	Sakthivel, S.	Managing risk in offshore systems development
Carver et al. 2007	Carver, Jeffrey C.; Kendall, Richard P.; Squires, Susan E.; Post, Douglass E.	Software Development Environments for Scientific and Engineering Software: A Series of Case Studies
PA 2.1.2.3 Plan for Data Management PA 2.1.2.4 Plan for Project Resources		
Tools for distributed project management		
Rice et al. 2007	Rice, Daniel; Davidson, Barry; Dannenhoffer, John; Gay, Geri	Improving the Effectiveness of Virtual Teams by Adapting Team Processes
Gorton et al. 1997	Gorton, I.; Hawryszkiewycz, I.; Ragoonaden, K.	Collaborative tools and processes to support software engineering shift work
Lewkowicz et al. 2008	Lewkowicz, Myriam; Wijnhoven, Fons; Draghici, Anca	Misunderstandings in Global Virtual Engineering Teams: Definitions, Causes, and Guidelines for Knowledge Sharing and Interaction
Donker et al. 2008	Donker, Hilko; Blumberg, Malte	Collaborative process management and virtual teams
Tremaine et al. 2007	Tremaine, Marilyn; Milewski, Allen; Egan, Richard; Zhang, Suling	A Tale of Two Teams: Success and Failure in Virtual Team Meetings
Prasad 1999	Prasad, Biren	System Integration Techniques of Sharing and Collaboration

		among Work-groups, Computers and Processes
Lee et al. 2006	Lee, Lai-Chung; Wei, Whei-Jane	Case Study of Breakdown Analysis on Identification of Remote Team Communication Problems
Cichocki et al. 2008	Cichocki, Przemyslaw; Maccari, Alessandro	Empirical Analysis of a Distributed Software Development Project
Smith et al. 2005	Smith, J. L.; Bohner, S. A.; McCrickard, D. S.	Toward introducing notification technology into distributed project teams
Ralyte et al. 2008	Ralyte, J.; Lamielle, X.; Arni-Bloch, N.; Leonard, M.	A framework for supporting management in distributed information systems development
Au et al. 2004	Au, P. K.; Ma, S.	Lessons learned from development and implementation of a strategic life cycle model
Gotel et al. 2009	Gotel, Olly; Kulkarni, Vidya; Des Phal; Say, Moniphal; Scharff, Christelle; Sunetnanta, Thanwadee	Evolving an infrastructure for student global software development projects: lessons for industry
Beise 2004	Beise, Catherine M.	IT project management and virtual teams
Thomsen J et al. 10, no	Thomsen J; Levitt RE; Nass CI	The virtual team alliance (VTA): extending Galbraith's information-processing model to account for goal incongruency
Fernandez A et al. 9, no	Fernandez A; Garzaldeen B; Grutzner I; Munch J	Guided support for collaborative modeling, enactment and simulation of software development processes
McQuay 2005	McQuay, W. K.	Distributed collaborative environments for systems engineering
Maurer et al. 2002	Maurer, Frank; Holz, Harald	Integrating Process Support and Knowledge Management for Virtual Software Development Teams

Suzuki et al. 1999	Suzuki, J.; Yamamoto, Y.	Leveraging distributed software development
Poehler et al. 2007	Poehler, L.; Schumacher, T.	The Virtual Team Challenge: Is It Time for Training?
Ellis et al. 2007	Ellis, Jason B.; Wahid, Shahtab; Danis, Catalina; Kellogg, Wendy A.	Task and social visualization in software development: evaluation of a prototype
PA 2.1.2.5 Plan for Needed Knowledge and Skills		
Importance of the primary language		
Carver et al. 2007	Carver, Jeffrey C.; Kendall, Richard P.; Squires, Susan E.; Post, Douglass E.	Software Development Environments for Scientific and Engineering Software: A Series of Case Studies
Higher level languages is unimportant		
Carver et al. 2007	Carver, Jeffrey C.; Kendall, Richard P.; Squires, Susan E.; Post, Douglass E.	Software Development Environments for Scientific and Engineering Software: A Series of Case Studies
Agile methodologies are better accepted by scientific and engineering code developers than more traditional methodologies		
PA 2.1.2.6 Plan Stakeholder Involvement		
PA 2.1.2.7 Establish the Project Plan		
... for corporate open source development		
Gurbani et al. 2006	Gurbani, Vijay K.; Garvert, Anita; Herbsleb, James D.	A case study of a corporate open source development model
PA 2.1.3 Obtain Commitment to the Plan		
PA 2.1.3.1 Review Plans that Affect the Project		
PA 2.1.3.2 Reconcile Work and Resource Levels		
PA 2.1.3.3 Obtain Plan Commitment		
PA 2.2 Project Monitoring and Control		
PA 2.2.1 Monitor Project Against Plan		
PA 2.2.1.1 Monitor Project Planning Parameters		
Controling for IT and Software		
2007		Controlling for IT and Software
Measures for Distributed Development		
Gurbani et al. 2006	Gurbani, Vijay K.; Garvert, Anita; Herbsleb, James D.	A case study of a corporate open source development model
PA 2.2.1.2 Monitor Commitments		
PA 2.2.1.3 Monitor Project Risks		
identify risks		
Smith et al. 2005	Smith, Jamie L.; Bohner, Shawn A.; McCrickard, D. Scott	Project management for the 21st century: supporting collaborative design through risk analysis

Sakthivel 2007	Sakthivel, S.	Managing risk in offshore systems development
PA 2.2.1.4 Monitor Data Management		
PA 2.2.1.5 Monitor Stakeholder Involvement		
PA 2.2.1.6 Conduct Progress Reviews		
Using Concurrent Version System for Project Planning		
Fitzpatrick et al. 2006	Fitzpatrick, Geraldine; Marshall, Paul; Phillips, Anthony	CVS integration with notification and chat: lightweight software team collaboration
PA 2.2.1.7 Conduct Milestone Reviews		
PA 2.2.2 Manage Corrective Action to Closure		
PA 2.2.2.1 Analyze Issues		
Process of Performance Degradation Detection		
Yilmaz et al. 2005	Yilmaz, Cemal; Krishna, Arvind S.; Memon, Atif; Porter, Adam; Schmidt, Douglas C.; Gokhale, Aniruddha; Natarajan, Balachandran	Main effects screening: a distributed continuous quality assurance process for monitoring performance degradation in evolving software systems
PA 2.2.2.2 Take Corrective Action		
PA 2.2.2.3 Manage Corrective Action		
PA 2.3 Management Process		
PA 2.3.1 Use the Project's Defined Process		
PA 2.3.1.1 Establish the Project's Defined Process		
Process to overcome geographical barriers		
Reijers et al.	Reijers, Hajo; Song, Minseok; Jeong, Byungduk	Analysis of a collaborative workflow process with distributed actors
PA 2.3.1.2 Use Organizational Process Assets for Planning Project Activities		
PA 2.3.1.3 Establish the Project's Work Environment		
PA 2.3.1.4 Integrate Plans		
PA 2.3.1.5 Manage the Project Using the Integrated Plans		
PA 2.3.1.6 Contribute to the Organizational Process Assets		
PA 2.3.2 Coordinate and Collaborate with Relevant Stakeholders		
PA 2.3.2.1 Manage Stakeholder Involvement		
Collaborative Product Commerce		
Bardhan 2007	Bardhan, Indranil	Toward a theory to study the use of collaborative product commerce for product development
Communication styles and tools with stakeholders		
Pesendorfer et al. 2007	Pesendorfer, Eva-Maria; Graf, Andrea; Koeszegi, Sabine	Relationship in electronic negotiations: Tracking behavior over time

Fricker et al. 2008	Fricker, Samuel; Grünbacher, Paul	Negotiation Constellations – Method Selection Framework for Requirements Negotiation
Damian D 24, no	Damian D	Stakeholders in global requirements engineering: Lessons learned from practice
Aranda et al. 2007	Aranda, Gabriela; Vizcaíno, Aurora; Cechich, Alejandra; Piattini, Mario	How to Choose Groupware Tools Considering Stakeholders' Preferences During Requirements Elicitation?
Al-Ani et al. 2008	Al-Ani, Ban; Trainer, Erik; Ripley, Roger; Sarma, Anita; van der Hoek, André; Redmiles, David	Continuous coordination within the context of cooperative and human aspects of software engineering
PA 2.3.2.2 Manage Dependencies		
PA 2.3.2.3 Resolve Coordination Issues		
PA 2.3.3 Quantitatively Manage the Project		
PA 2.3.3.1 Establish the Project's Objectives		
PA 2.3.3.2 Compose the Defined Processes		
PA 2.3.3.3 Select the Subprocesses that Will Be Statistically Managed		
PA 2.3.3.4 Manage Project Performance		
Monitor performance		
Yilmaz et al. 2005	Yilmaz, Cemal; Krishna, Arvind S.; Memon, Atif; Porter, Adam; Schmidt, Douglas C.; Gokhale, Aniruddha; Natarajan, Balachandran	Main effects screening: a distributed continuous quality assurance process for monitoring performance degradation in evolving software systems
Visualize quantitative performance data		
Ellis et al. 2007	Ellis, Jason B.; Wahid, Shahtab; Danis, Catalina; Kellogg, Wendy A.	Task and social visualization in software development: evaluation of a prototype
PA 2.3.4 Statistically Manage Subprocess Performance		
PA 2.3.4.1 Select Measures and Analytic Techniques		
PA 2.3.4.2 Apply Statistical Methods to Understand Variation		
PA 2.3.4.3 Monitor Performance of the Selected Subprocesses		
PA 2.3.4.4 Record Statistical Management Data		
PA 2.3.5 Project initiation		
Open Source in Distributed Developmenet		
Gurbani et al. 2006	Gurbani, Vijay K.; Garvert, Anita; Herbsleb, James D.	A case study of a corporate open source development model

Distributed Effort and Cost estimation		
Kommeren et al. 2007	Kommeren, Rob; Parviainen, Päivi	Philips experiences in global distributed software development
Seifert et al. 2005	Seifert, Marcus; Eschenbaecher, Jens	Predictive Performance Measurement in Virtual Organisations
Pillis et al. 2007	Pillis, Emmeline de; Furumo, Kimberly	Counting the cost of virtual teams
Keil et al. 2006	Keil, Patrick; Paulish, Daniel J.; Sangwan, Raghvinder S.	Cost estimation for global software development
Weihua Gan 2006	Weihua Gan	Analysis on the Costs of IT-Outsourcing
Early Estimation of Project Size		
Abrahão et al. 2006	Abrahão, Silvia; Poels, Geert; Pastor, Oscar	A functional size measurement method for object-oriented conceptual schemas: design and evaluation issues
2007		Empirical Laws and Rules of Thumb
Helander et al. 1998	Helander, M. E.; Ming Zhao; Ohlsson, N.	Planning models for software reliability and cost
PA 2.3.6 Portfolio evaluation PA 2.3.7 Project closure PA 2.3.8 Safety Management		
PA 2.3.8.1 Develop Safety Plans		
Safety Aspects in Automotive Software Engineering		
Tindell et al. 2003	Tindell, Ken; Kopetz, Hermann; Wolf, Fabian; Ernst, Rolf	Safe Automotive Software Development
Patterns for safety related Projects		
C:\Daten\arbeitsverzeichnis\Promotion\Modell in Excel\papers sys-rev\Safety\06 Patterns in Safety-Related Projects.pdf		
Integration of safety analysis		
Domis et al. 2008	Domis, Dominik; Trapp, Mario	Integrating Safety Analyses and Component-Based Design
PA 2.3.8.2 Monitor Safety Incidents PA 2.3.8.3 Manage Safety related Suppliers		
Supplier Agreement		
confidential		
Audits		
confidential		

confidential		
Assessments		
confidential		
confidential		
Safety Case		
confidential		
PA 2.4 Distributed Team Management		
PA 2.4.1 Team Building /Management		
Multi-disciplinary teams improve creativity		
Bachmann 2006	Bachmann, Anne	Melting pot or tossed salad? Implications for designing effective multicultural workgroups
Maloney et al. 2006	Maloney, Mary; Zellmer-Bruhn, Mary	Building bridges, windows and cultures: Mediating mechanisms between team heterogeneity and performance in global teams
Carte et al. 2006	Carte, Traci; Chidambaram, Laku; Garfield, Monica; Hicks, Lindsey; Cole, Cassie	Group Creativity and Collaborative Technologies: Understanding the Role of Visual Anonymity
van der Duim et al. 2007	van der Duim, Louwarnoud; Andersson, Jesper; Sinnema, Marco	Good Practices for Educational Software Engineering Projects
Draghici 2008	Draghici, Anca	Influence of Multi-Culturality in Virtual Teams
Carver et al. 2007	Carver, Jeffrey C.; Kendall, Richard P.; Squires, Susan E.; Post, Douglass E.	Software Development Environments for Scientific and Engineering Software: A Series of Case Studies
Team Unity and Cohesion		
Nicolopoulou et al. 2006	Nicolopoulou, Katerina; Koštomaj, Mitja; Campos, Andre	How to address group dynamics in virtual worlds
Oshri et al. 2008	Oshri, Ilan; Kotlarsky, Julia; Willcocks, Leslie	Missing links: building critical social ties for global collaborative teamwork
Kirkman BL et al. 47, no	Kirkman BL; Rosen B; Tesluk PE; Gibson CB	The impact of team empowerment on virtual team performance: the moderating role of face-to-face interaction

	Huang et al. 2006	Huang, Haiyan; Ocker, Rosalie	Preliminary insights into the in-group/out-group effect in partially distributed teams: an analysis of participant reflections
	Ellis et al. 2008	Ellis, Jason B.; Luther, Kurt; Bessiere, Katherine; Kellogg, Wendy A.	Games for virtual team building
	Lings et al. 2006	Lings, Brian; Lundell, Björn; Ågerfalk, Pär; Fitzgerald, Brian	Ten Strategies for Successful Distributed Development
	Gotel et al. 2007	Gotel, Olly; Kulkarni, Vidya; Neak, Long; Scharff, Christelle; Seng, Sopheap	Introducing Global Supply Chains into Software Engineering Education
	Ocker 2005	Ocker, R. J.	Influences on creativity in asynchronous virtual teams: a qualitative analysis of experimental teams
	van der Duim et al. 2007	van der Duim, Louwarnoud; Andersson, Jesper; Sinnema, Marco	Good Practices for Educational Software Engineering Projects
	Salisbury et al. 2006	Salisbury, Wm David; Carte, Traci A.; Chidambaram, Laku	Cohesion in virtual teams: validating the perceived cohesion scale in a distributed setting
	Bos et al. 2004	Bos, Nathan; Shami, N. Sadat; Olson, Judith S.; Cheshin, Arik; Nan, Ning	In-group/out-group effects in distributed teams: an experimental simulation
accessablilty of technical and academical abilities of team members			
	Gutwin et al. 2004	Gutwin, Carl; Penner, Reagan; Schneider, Kevin	Group awareness in distributed software development
	Karduck et al. 2006	Karduck, Achim; Sienou, Amadou	Teambroker: Constraint Based Brokerage of Virtual Teams
	Cristal et al. 2006	Cristal, M.; Reis, J.	Leveraging lessons learned for distributed projects through communities of practice
	Draghici et al. 2008	Draghici, Anca; Draghici, George	Building a Knowledge Share Culture in a Virtual Organization. Case Study for VRL-KCiP NoE

	Shen 2007	Shen, Yide	Transactive memory system development in virtual teams: the potential role of shared identity and shared context
	Casey et al. 2006	Casey, Valentine; Richardson, Ita	Uncovering the reality within virtual software teams
	Adya et al. 2007	Adya, Monica; Nath, Dhruv; Malik, Amit; Sridhar, Varadharajan	Bringing global sourcing into the classroom: experiential learning via software development project
Prepare for Tensions in corporate and national culture			
	Wei 2007	Wei, Kangning	Sharing Knowledge in Global Virtual Teams
	Mikawa et al. 2009	Mikawa, Suzanne P.; Cunnington, Sharon K.; Gaskins, Scott A.	Removing barriers to trust in distributed teams: understanding cultural differences and strengthening social ties
	Prikladnicki R et al. IEEE	Prikladnicki R; Evaristo R; Damian D; Audy JLN	Conducting qualitative research in an international and distributed research team: challenges and lessons learned
	Kussmaul C et al. 4th Conference on Extreme Programming and Agile Methods	Kussmaul C; Jack R; Sponsler B	Outsourcing and offshoring with agility: a case study
	Cater-Steel et al. 2008	Cater-Steel, A.; Toleman, M.	The impact of national culture on software engineering practices
	Ungerleider 2008	Ungerleider, John	Conflict
	Last et al. 2002	Last, Mary Z.; Daniels, Mats; Hause, Martha L.; Woodroffe, Mark R.	Learning from students: continuous improvement in international collaboration
	Cater Steel A et al. 8, no	Cater Steel A; Toleman M	The impact of national culture on software engineering practices
	Setamanit et al. 2008	Setamanit, Siri-on; Raffo, David	Identifying Key Success Factors for Globally Distributed Software Development Project Using Simulation: A Case Study
	Reijers et al.	Reijers, Hajo; Song, Minseok; Jeong,	Analysis of a collaborative workflow process

		Byungduk	with distributed actors
	Cichocki et al. 2008	Cichocki, Przemyslaw; Maccari, Alessandro	Empirical Analysis of a Distributed Software Development Project
	Bachmann 2006	Bachmann, Anne	Melting pot or tossed salad? Implications for designing effective multicultural workgroups
	Pillis et al. 2007	Pillis, Emmeline de; Furumo, Kimberly	Counting the cost of virtual teams
	Lindqvist et al. 2006	Lindqvist, Edvin; Lundell, Björn; Lings, Brian	Distributed development in an intra-national, intra-organisational context: an experience report
	Koh et al. 2009	Koh, Christine; Joseph, Damien; Ang, Soon	Cultural intelligence and collaborative work: intercultural competencies in global technology work teams
	Gibbs 2009	Gibbs, Jennifer L.	Culture as kaleidoscope: navigating cultural tensions in global collaboration
	Adya et al. 2007	Adya, Monica; Nath, Dhruv; Malik, Amit; Sridhar, Varadharajan	Bringing global sourcing into the classroom: experiential learning via software development project
Leadership in Distributed Teams			
	Heckman et al. 2007	Heckman, Robert; Crowston, Kevin; Misiolek, Nora	A Structurational Perspective on Leadership in Virtual Teams
Set up a Team structure			
	Pichler 2007	Pichler, H.	Be successful, take a hostage or "outsourcing the outsourcing Manager"
	Karayaz et al. 2007	Karayaz, G.; Keating, C. B.	Virtual Team Effectiveness Using Dyadic Teams
	Potter RE et al. 56, no	Potter RE; Balthazard PA	Virtual team interaction styles: assessment and effects
	Mortensen et al. 2007	Mortensen, Mark; Woolley, Anita; O'Leary, Michael	Conditions Enabling Effective Multiple Team Membership
	D'Eredita et al. 2007	D'Eredita, Michael; Nilan, Michael	Conceptualizing Virtual Collaborative Work

Braithwaite et al. 2005	Braithwaite, Keith; Joyce, Tim	XP Expanded: Distributed Extreme Programming
Balthazard P et al. 35, no	Balthazard P; Potter RE; Warren J	Expertise, extraversion and group interaction styles as performance indicators in virtual teams
Sharp et al. 2008	Sharp, Jason H.; Ryan, Sherry D.	A Preliminary Conceptual Model for Exploring Global Agile Teams
He et al. 2008	He, Mei; Li, Mingshu; Wang, Qing; Yang, Ye; Ye, Kai	An Investigation of Software Development Productivity in China
2008		Customer Challenges and Demands
Bal J et al. 38, no	Bal J; Foster P	Managing the virtual team and controlling effectiveness
Casey et al. 2006	Casey, V.; Richardson, I.	Project management within virtual software teams
Karduck et al. 2006	Karduck, Achim; Sienou, Amadou	Teambroker: Constraint Based Brokerage of Virtual Teams
Adya et al. 2007	Adya, Monica; Nath, Dhruv; Malik, Amit; Sridhar, Varadharajan	Bringing global sourcing into the classroom: experiential learning via software development project
Analyze and Solve conflicts		
Ungerleider 2008	Ungerleider, John	Conflict
Oportunity for self reflection and self Correction		
Adya et al. 2007	Adya, Monica; Nath, Dhruv; Malik, Amit; Sridhar, Varadharajan	Bringing global sourcing into the classroom: experiential learning via software development project
Create social ties		
Kussmaul C et al. 4th Conference on Extreme Programming and Agile Methods	Kussmaul C; Jack R; Sponsler B	Outsourcing and offshoring with agility: a case study
Setamanit et al. 2008	Setamanit, Siri-on; Raffo, David	Identifying Key Success Factors for Globally Distributed Software Development Project Using Simulation: A Case Study
Braithwaite et al. 2005	Braithwaite, Keith; Joyce, Tim	XP Expanded: Distributed Extreme Programming

		Oshri et al. 2008	Oshri, Ilan; Kotlarsky, Julia; Willcocks, Leslie	Missing links: building critical social ties for global collaborative teamwork
		Huang et al. 2006	Huang, Haiyan; Ocker, Rosalie	Preliminary insights into the in-group/out-group effect in partially distributed teams: an analysis of participant reflections
		Adya et al. 2007	Adya, Monica; Nath, Dhruv; Malik, Amit; Sridhar, Varadharajan	Bringing global sourcing into the classroom: experiential learning via software development project
PA 2.4.2 Coordination / Communication				
	Maintain close communication with the supplier			
		Fricker et al. 2007	Fricker, Samuel; Gorschek, Tony; Myllyperkiö, Petri	Handshaking Between Software Projects and Stakeholders Using Implementation Proposals
		Edgar 2006	Edgar, D. R.	Lessons Learned: Outsourcing Support Equipment Development
		Mathrani et al. 2007	Mathrani, Anuradha; Parsons, David	Management of Knowledge Transfer in Distributed Software Organizations: The Outsourcers' Perspective
		Kommeren et al. 2007	Kommeren, Rob; Parviainen, Päivi	Philips experiences in global distributed software development
	Share knowledge			
		Wongthongtham et al. 2007	Wongthongtham, Pornpit; Chang, Elizabeth; Aseeri, Ahmed	Towards Social Network Based Approach for Software Engineering Ontology Sharing and Evolution
		Lim 2007	Lim, Shung	GSSD-Enterprise for Multinational Corporations
		Wei 2007	Wei, Kangning	Sharing Knowledge in Global Virtual Teams
		Mathrani et al. 2007	Mathrani, Anuradha; Parsons, David	Management of Knowledge Transfer in Distributed Software Organizations: The Outsourcers' Perspective

2008		Customer Challenges and Demands
Distributed Decision Making		
Fang He et al. 2007	Fang He; Paul, S.	Time Pressure and Reward Inspiration as Outcome Controls for Information Sharing in Problem-Solving Virtual Teams
Rice et al. 2007	Rice, Daniel; Davidson, Barry; Dannenhoffer, John; Gay, Geri	Improving the Effectiveness of Virtual Teams by Adapting Team Processes
Ferscha et al. 1999	Ferscha, A.; Scheiner, C.	Collective choice in virtual teams
Clear Distribution Rational		
Lings et al. 2006	Lings, Brian; Lundell, Björn; Ågerfalk, Pär; Fitzgerald, Brian	Ten Strategies for Successful Distributed Development
Barrier-free communication		
Weber DW 2001	Weber DW	Data topology and process patterns for distributed development
García et al. 2005	García, Ariel; Hardt, Marcus; Kornmayer, Harald	Tools for Distributed Development and Deployment on the Grid
Rama et al. 2006	Rama, Jiten; Bishop, Judith	A survey and comparison of CSCW groupware applications
Herbsleb 2007	Herbsleb, James D.	Global Software Engineering: The Future of Socio-technical Coordination
Lings et al. 2006	Lings, Brian; Lundell, Björn; Ågerfalk, Pär; Fitzgerald, Brian	Ten Strategies for Successful Distributed Development
2008		Customer Challenges and Demands
Purvis et al. 2004	Purvis, Maryam; Purvis, Martin; Cranefield, Stephen	Educational experiences from a Global Software Engineering (GSE) project
Gurbani et al. 2006	Gurbani, Vijay K.; Garvert, Anita; Herbsleb, James D.	A case study of a corporate open source development model
rules and constraints for distributed communication		
Šmite et al. 2008	Šmite, Darja; Moe, Nils; Torkar, Richard	Pitfalls in Remote Team Coordination: Lessons Learned from a Case Study

	Ratcheva et al. 2001	Ratcheva, Violina; Vyakarnam, Shailendra	The challenges of virtual partnerships: Critical success factors in the formation of inter-organisational teams
	Mathrani et al. 2007	Mathrani, Anuradha; Parsons, David	Management of Knowledge Transfer in Distributed Software Organizations: The Outsourcers' Perspective
Clarity about who does what where			
	Haag et al. 1998	Haag, Zsolt; Foley, Richard; Newman, Julian	Deontic formalism for co-ordinating software development in virtual software corporations
	Ocker et al. 2007	Ocker, Rosalie; Huang, Haiyan; Trauth, Eileen; Purao, Sandeep	The Tension Between Expectations of Availability and the Reality of Availability in Hybrid Teams
	Seybold et al. 2008	Seybold, C.; Keller, R. K.	Aligning Software Maintenance to the Offshore Reality
	Ignat et al. 2008	Ignat, Claudia-Lavinia; Papadopoulou, Stavroula; Oster, Gérald; Norrie, Moira C.	Providing awareness in multi-synchronous collaboration without compromising privacy
	Grinter et al. 1999	Grinter, Rebecca E.; Herbsleb, James D.; Perry, Dewayne E.	The geography of co-ordination: dealing with distance in R&D work
	Kussmaul C et al. 4th Conference on Extreme Programming and Agile Methods	Kussmaul C; Jack R; Sponsler B	Outsourcing and off-shoring with agility: a case study
	Wahyudin et al. 2008	Wahyudin, Dindin; Heindl, Matthias; Eckhard, Benedikt; Schatten, Alexander; Biffl, Stefan	In-Time Role-Specific Notification as Formal Means to Balance Agile Practices in Global Software Development Settings
	Setamanit et al. 2008	Setamanit, Siri-on; Raffo, David	Identifying Key Success Factors for Globally Distributed Software Development Project Using Simulation: A Case Study
	Lewkowicz et al. 2008	Lewkowicz, Myriam; Wijnhoven, Fons; Draghici, Anca	Misunderstandings in Global Virtual Engineering Teams: Defini-

			tions, Causes, and Guidelines for Knowledge Sharing and Interaction
	Lee et al. 2006	Lee, Lai-Chung; Wei, Whei-Jane	Case Study of Breakdown Analysis on Identification of Remote Team Communication Problems
	Kommeren et al. 2007	Kommeren, Rob; Parviainen, Päivi	Philips experiences in global distributed software development
	Gutwin et al. 2005	Gutwin, Carl; Schneider, Kevin; Paquette, David; Penner, Reagan	Supporting Group Awareness in Distributed Software Development
	Rama et al. 2006	Rama, Jiten; Bishop, Judith	A survey and comparison of CSCW groupware applications
	Purvis et al. 2004	Purvis, Maryam; Purvis, Martin; Cranefield, Stephen	Educational experiences from a Global Software Engineering (GSE) project
	Herbsleb et al. 2005	Herbsleb, James D.; Paulish, Daniel J.; Bass, Matthew	Global software development at siemens: experience from nine projects
	Gutwin et al. 2004	Gutwin, Carl; Penner, Reagan; Schneider, Kevin	Group awareness in distributed software development
	Boyer et al. 1998	Boyer, David G.; Handel, Mark J.; Herbsleb, James	Virtual community prescence awareness
	Adya et al. 2007	Adya, Monica; Nath, Dhruv; Malik, Amit; Sridhar, Varadharajan	Bringing global sourcing into the classroom: experiential learning via software development project
Negotiation Styles			
	Burden K 19, no	Burden K	"We're in it to win it"- negotiating successful outsourcing transactions
	Pesendorfer et al. 2007	Pesendorfer, Eva-Maria; Graf, Andrea; Koeszegi, Sabine	Relationship in electronic negotiations: Tracking behavior over time
Ethical behaviour within the project (e.g. regarding time and resource consumption)			
	Hawthorne et al. 2006	Hawthorne, Matthew; Perry, Dewayne	Software Engineering Education in the Era of Outsourcing, Distribut-

			ed Development, and Open Source Software: Challenges and Opportunities
	Schadewitz et al. 2009	Schadewitz, Nicole; Zakaria, Norhayati	Cross-cultural collaboration Wiki: evolving knowledge about international teamwork
	Boehm 2006	Boehm, Barry	Value-Based Software Engineering: Seven Key Elements and Ethical Considerations
	Huang et al. 2006	Huang, Haiyan; Ocker, Rosalie	Preliminary insights into the in-group/out-group effect in partially distributed teams: an analysis of participant reflections
Communication styles and techhniques			
	Damian et al. 2006	Damian, Daniela; Hadwin, Allyson; Al-Ani, Ban	Instructional design and assessment strategies for teaching global software development: a framework
	Urdangarin R et al. IEEE 2008	Urdangarin R; Fernandes P; Avritzer A; Paulish D	Experiences with agile practices in the global studio project Information & Communication Technologies
	Chituc et al. 2005	Chituc, Claudia-Melania; Azevedo, Americo	Multi-Perspective Challenges on Collaborative Networks Business Environments
	Chang et al. 2007	Chang, Klarissa T.; Ehrlich, Kate	Out of sight but not out of mind?: Informal networks, communication and media use in global software teams
	Noll et al. 1999	Noll, J.; Scacchi, W.	Supporting software development in virtual enterprises
	Leinonen et al. 2005	Leinonen, Piritta; Järvelä, Sanna; Häkkinen, Päivi	Conceptualizing the Awareness of Collaboration: A Qualitative Study of a Global Virtual Team
	Huysman et al. 2003	Huysman, Marleen; Steinfield, Charles; Jang, Chyng-Yang; David, Kenneth; Veld, Mirjam Huis in	Virtual Teams and the Appropriation of Communication Technology: Exploring the Concept of Media

	't; Poot, Jan; Mulder, Ingrid	Stickiness
Griffin 2008	Griffin, Teressa Moore	Effective Intercultural Communication
Gilbert et al. 2008	Gilbert, Eric; Karahalios, Karrie	CodeSaw: A Social Visualization of Distributed Software Development
Dustdar 2004	Dustdar, Schahram	Caramba—A Process-Aware Collaboration System Supporting Ad hoc and Collaborative Processes in Virtual Teams
Bernuy et al. 2008	Bernuy, Augusto; Joyanes, Luis	Collaborative e-Business and Software Agents
Rees 2002	Rees, M. J.	A feasible user story tool for agile software development?
Boden et al. 2007	Boden, A.; Nett, B.; Wulf, V.	Coordination Practices in Distributed Software Development of Small Enterprises
Ocker et al. 2008	Ocker, Rosalie J.; Fjermestad, Jerry	Communication differences in virtual design teams: findings from a multi-method analysis of high and low performing experimental teams
Cataldo et al. 2008	Cataldo, Marcelo; Herbsleb, James D.	Communication networks in geographically distributed software development
Boden et al. 2008	Boden, Alexander; Nett, Bernhard; Wulf, Volker	Articulation work in small-scale offshore software development projects
Weber DW 2001	Weber DW	Data topology and process patterns for distributed development
Prikladnicki R et al. IEEE	Prikladnicki R; Evaristo R; Damian D; Audy JLN	Conducting qualitative research in an international and distributed research team: challenges and lessons learned

	Potter RE et al. 56, no	Potter RE; Balthazard PA	Virtual team interaction styles: assessment and effects
	Kirkman BL et al. 47, no	Kirkman BL; Rosen B; Tesluk PE; Gibson CB	The impact of team empowerment on virtual team performance: the moderating role of face-to-face interaction
	Haag Z et al. 23rd Euromicro Conference: New Frontiers of Information Technology (Cat	Haag Z; Foley R; Newman J	Software process improvement in geographically distributed software engineering: an initial evaluation
	Cater Steel A et al. 8, no	Cater Steel A; Toleman M	The impact of national culture on software engineering practices
	Balthazard P et al. 35, no	Balthazard P; Potter RE; Warren J	Expertise, extraversion and group interaction styles as performance indicators in virtual teams
	Wongthongtham et al. 2007	Wongthongtham, Pornpit; Chang, Elizabeth; Aseeri, Ahmed	Towards Social Network Based Approach for Software Engineering Ontology Sharing and Evolution
	Wongthongtham et al. 2007	Wongthongtham, Pornpit; Chang, Elizabeth; Dillon, Tharam	Multi-site Distributed Software Development: Issues, Solutions, and Challenges
	Wahyudin et al. 2008	Wahyudin, Dindin; Heindl, Matthias; Eckhard, Benedikt; Schatten, Alexander; Biffl, Stefan	In-Time Role-Specific Notification as Formal Means to Balance Agile Practices in Global Software Development Settings
	Sharp et al. 2008	Sharp, Jason H.; Ryan, Sherry D.	A Preliminary Conceptual Model for Exploring Global Agile Teams
	Setamanit et al. 2008	Setamanit, Siri-on; Raffo, David	Identifying Key Success Factors for Globally Distributed Software Development Project Using Simulation: A Case Study
	Ocker et al. 2007	Ocker, Rosalie; Huang, Haiyan; Trauth, Eileen; Purao, Sandeep	The Tension Between Expectations of Availability and the Reality of Availability in Hybrid Teams

	Ning et al. 2006	Ning, Anliang; Hou, Hong; Hua, Qingyi; Yu, Bin; Hao, Kegang	Requirements Engineering Processes Improvement: A Systematic View
	Mullally et al. 2009	Mullally, Brenda; Stapleton, Larry	Methodology Usage by Virtual Information Systems Development Teams
	Maurer et al. 2002	Maurer, Frank; Holz, Harald	Integrating Process Support and Knowledge Management for Virtual Software Development Teams
	Lings et al. 2006	Lings, Brian; Lundell, Björn; Ågerfalk, Pär; Fitzgerald, Brian	Ten Strategies for Successful Distributed Development
	Lewkowicz et al. 2008	Lewkowicz, Myriam; Wijnhoven, Fons; Draghici, Anca	Misunderstandings in Global Virtual Engineering Teams: Definitions, Causes, and Guidelines for Knowledge Sharing and Interaction
	Lee et al. 2006	Lee, Lai-Chung; Wei, Whei-Jane	Case Study of Breakdown Analysis on Identification of Remote Team Communication Problems
	Hole et al. 2008	Hole, Steinar; Moe, Nils Brede	A Case Study of Coordination in Distributed Agile Software Development
	Gotel et al. 2007	Gotel, Olly; Kulkarni, Vidya; Neak, Long; Scharff, Christelle; Seng, Sopheap	Introducing Global Supply Chains into Software Engineering Education
	Godart et al. 2004	Godart, Claude; Molli, Pascal; Oster, Gérald; Perrin, Olivier; Skaf-Molli, Hala; Ray, Pradeep; Rabhi, Fethi	The ToxicFarm Integrated Cooperation Framework for Virtual Teams
	Draghici et al. 2008	Draghici, Anca; Draghici, George	Building a Knowledge Share Culture in a Virtual Organization. Case Study for VRL-KCiP NoE
	Braithwaite et al. 2005	Braithwaite, Keith; Joyce, Tim	XP Expanded: Distributed Extreme Programming

	Bongio et al. 2006	Bongio, Aldo; van Bruggen, Jan; Ceri, Stefano; Cristea, Valentin; Dolog, Peter; Hoffmann, Andreas; Matera, Maristella; Mura, Marzia; Taddeo, Antonio; Zhou, Xuan; Zoni, Larissa	COOPER: Towards a Collaborative Open Environment of Project-Centred Learning
	Bansler et al. 2006	Bansler, JÃ¸rgen P.; Havn, Erling	Sensemaking in Technology-Use Mediation: Adapting Groupware Technology in Organizations
	Seybold et al. 2008	Seybold, C.; Keller, R. K.	Aligning Software Maintenance to the Offshore Reality
	Ocker 2005	Ocker, R. J.	Influences on creativity in asynchronous virtual teams: a qualitative analysis of experimental teams
	Marttiin et al. 2002	Marttiin, P.; Lehto, J. A.; Nyman, G.	Understanding and evaluating collaborative work in multi-site software projects - a framework proposal and preliminary results
	Hogan 2006	Hogan, B.	Lessons learned from an extremely distributed project
	Haag et al. 1997	Haag, Z.; Foley, R.; Newman, J.	Software process improvement in geographically distributed software engineering: an initial evaluation
	van der Duim et al. 2007	van der Duim, Louwarnoud; Andersson, Jesper; Sinnema, Marco	Good Practices for Educational Software Engineering Projects
	Valetto et al. 2007	Valetto, Giuseppe; Helander, Mary; Ehrlich, Kate; Chulani, Sunita; Wegman, Mark; Williams, Clay	Using Software Repositories to Investigate Socio-technical Congruence in Development Projects
	Thomas et al. 2007	Thomas, Dominic M.; Bostrom, Robert P.; Gouge, Marianne	Making knowledge work in virtual teams
	Schadewitz et al. 2009	Schadewitz, Nicole; Zakaria, Norhayati	Cross-cultural collaboration Wiki: evolving knowledge about in-

			ternational teamwork
	Rama et al. 2006	Rama, Jiten; Bishop, Judith	A survey and comparison of CSCW groupware applications
	Ignat et al. 2008	Ignat, Claudia-Lavinia; Papadopoulou, Stavroula; Oster, Gérald; Norrie, Moira C.	Providing awareness in multi-synchronous collaboration without compromising privacy
	Herbsleb et al. 2005	Herbsleb, James D.; Paulish, Daniel J.; Bass, Matthew	Global software development at siemens: experience from nine projects
	Gotel et al. 2009	Gotel, Olly; Kulkarni, Vidya; Des Phal; Say, Moniphal; Scharff, Christelle; Sunetnanta, Thanwadee	Evolving an infrastructure for student global software development projects: lessons for industry
	Godart et al. 2001	Godart, C.; Bouthier, C.; Canalda, P.; Charoy, F.; Molli, P.; Perrin, O.; Saliou, H.; Bignon, J. C.; Halin, G.; Malcurat, O.	Asynchronous coordination of virtual teams in creative applications (co-design or co-engineering): requirements and design criteria
	Casey et al. 2006	Casey, Valentine; Richardson, Ita	Uncovering the reality within virtual software teams
	Boyer et al. 1998	Boyer, David G.; Handel, Mark J.; Herbsleb, James	Virtual community prescence awareness
	Belov et al. 2005	Belov, Nadya; Koeck, Colin; Krandick, Werner; Shaffer, Joshua; Hewett, Thomas	Intellectual teamwork on smartphones
	Begel 2008	Begel, Andrew	Effecting change: coordination in large-scale software development
	Alexander 2002	Alexander, Patricia M.	Teamwork, time, trust and information
	Al-Ani et al. 2008	Al-Ani, Ban; Trainer, Erik; Ripley, Roger; Sarma, Anita; van der Hoek, André; Redmiles, David	Continuous coordination within the context of cooperative and human aspects of software engineering
Media for communication			
	DeLuca D et al. 19, no	DeLuca D; Valacich JS	Virtual teams in and out of synchronicity

	Aranda et al. 2005	Aranda, Gabriela N.; Vizcaíno, Aurora; Cechich, Alejandra; Piattini, Mario	A Cognitive Perspective for Choosing Groupware Tools and Elicitation Techniques in Virtual Teams
	Vitale V IEEE Computer Society	Vitale V	IBM software development leveraging geographically distributed teams - the Interactive Solution Marketplace 2.0 (ISM) case study
	Midwinter et al. 2000	Midwinter, T.; Sheppard, P. J.	eCollaboration — 'The Drive for Simplicity'
	Aiello et al. 2007	Aiello, Giovanni; Alessi, Marco; Cossentino, Massimo; Urso, Alfonso; Vella, Giuseppe	RTDWD: Real-Time Distributed Wideband-Delphi for User Stories Estimation
	Whitehead 2007	Whitehead, Jim	Collaboration in Software Engineering: A Roadmap
	Thissen et al. 2007	Thissen, M. Rita; Page, Jean M.; Bharathi, Madhavi C.; Austin, Toyia L.	Communication tools for distributed software development teams
	Vizcaino A et al. First International Atlantic Web Intelligence Conference	Vizcaino A; Favela J; Piattini M; Garcia F	Supporting software maintenance in Web repositories through a multi-agent system
	Haag Z et al. 23rd Euromicro Conference: New Frontiers of Information Technology (Cat	Haag Z; Foley R; Newman J	Software process improvement in geographically distributed software engineering: an initial evaluation
	Dustdar et al. 2003	Dustdar, S.; Gall, H.	Architectural concerns in distributed and mobile collaborative systems
	Tanger et al. 2006	Tanger, Ralf; Kauff, Peter; Schreer, Oliver; Pavy, Dominique; Louis Dit Picard, Stephane; Saugis, Grégory	Team Collaboration Mixing Immersive Video Conferencing With Shared Virtual 3D Objects
	Maloney et al. 2006	Maloney, Mary; Zellmer-Bruhn, Mary	Building bridges, windows and cultures: Mediating mechanisms between team heterogeneity and performance in global teams

	Huysman et al. 2003	Huysman, Marleen; Steinfield, Charles; Jang, Chyng-Yang; David, Kenneth; Veld, Mirjam Huis in 't; Poot, Jan; Mulder, Ingrid	Virtual Teams and the Appropriation of Communication Technology: Exploring the Concept of Media Stickiness
	Gutwin et al. 2005	Gutwin, Carl; Schneider, Kevin; Paquette, David; Penner, Reagan	Supporting Group Awareness in Distributed Software Development
	Godart et al. 2004	Godart, Claude; Molli, Pascal; Oster, Gérald; Perrin, Olivier; Skaf-Molli, Hala; Ray, Pradeep; Rabhi, Fethi	The ToxicFarm Integrated Cooperation Framework for Virtual Teams
	Dustdar 2004	Dustdar, Schahram	Caramba—A Process-Aware Collaboration System Supporting Ad hoc and Collaborative Processes in Virtual Teams
	Aranda et al. 2007	Aranda, Gabriela; Vizcaíno, Aurora; Cechich, Alejandra; Piattini, Mario	How to Choose Groupware Tools Considering Stakeholders' Preferences During Requirements Elicitation?
	Anderl et al. 2008	Anderl, Reiner; Völz, Diana; Rollmann, Thomas	Knowledge Integration in Global Engineering
	Smith et al. 2005	Smith, J. L.; Bohner, S. A.; McCrickard, D. S.	Toward introducing notification technology into distributed project teams
	Seybold et al. 2008	Seybold, C.; Keller, R. K.	Aligning Software Maintenance to the Offshore Reality
	Hogan 2006	Hogan, B.	Lessons learned from an extremely distributed project
	Wong et al. 2007	Wong, Felix; Fernandez, George; McGovern, Jim	CONFER: towards groupware for building consensus in collaborative software engineering
	Steinfield et al. 1999	Steinfield, Charles; Jang, Chyng-Yang; Pfaff, Ben	Supporting virtual team collaboration: the TeamSCOPE system

	Schadewitz et al. 2009	Schadewitz, Nicole; Zakaria, Norhayati	Cross-cultural collaboration Wiki: evolving knowledge about international teamwork
	Rama et al. 2006	Rama, Jiten; Bishop, Judith	A survey and comparison of CSCW groupware applications
	Herbsleb 2007	Herbsleb, James D.	Global Software Engineering: The Future of Socio-technical Coordination
	Egea 2003	Egea, Kathy	Managing the managers: collaborative virtual teams with large staff and student numbers
	Boyer et al. 1998	Boyer, David G.; Handel, Mark J.; Herbsleb, James	Virtual community prescence awareness
	Belov et al. 2005	Belov, Nadya; Koeck, Colin; Krandick, Werner; Shaffer, Joshua; Hewett, Thomas	Intellectual teamwork on smartphones
	Ali-Babar 2008	Ali-Babar, Muhammad	The application of knowledge-sharing workspace paradigm for software architecture processes
	Al-Ani et al. 2008	Al-Ani, Ban; Trainer, Erik; Ripley, Roger; Sarma, Anita; van der Hoek, André; Redmiles, David	Continuous coordination within the context of cooperative and human aspects of software engineering
PA 2.4.3 Motivation			
	Pattern for motivation		
	Clear et al. 2005	Clear, Tony; Kassabova, Diana	Motivational patterns in virtual team collaboration
	define clear and transparent project priorities and goals		
	Thomsen J et al. 10, no	Thomsen J; Levitt RE; Nass CI	The virtual team alliance (VTA): extending Galbraith's information-processing model to account for goal incongruency
	Kussmaul C et al. 4th Conference on Extreme Programming and Agile Methods	Kussmaul C; Jack R; Sponsler B	Outsourcing and offshoring with agility: a case study
	Lings et al. 2006	Lings, Brian; Lundell, Björn; Ågerfalk, Pär;	Ten Strategies for Successful Distributed

		Fitzgerald, Brian	Development
	van der Duim et al. 2007	van der Duim, Louwarnoud; Andersson, Jesper; Sinnema, Marco	Good Practices for Educational Software Engineering Projects
	Lindqvist et al. 2006	Lindqvist, Edvin; Lundell, Björn; Lings, Brian	Distributed development in an intra-national, intra-organisational context: an experience report
	Gurbani et al. 2006	Gurbani, Vijay K.; Garvert, Anita; Herbsleb, James D.	A case study of a corporate open source development model
shared vision			
	Sangwan et al. 2008	Sangwan, Raghvinder S.; Ros, Johannes	Architecture leadership and management in globally distributed software development
	Adya et al. 2007	Adya, Monica; Nath, Dhruv; Malik, Amit; Sridhar, Varadharajan	Bringing global sourcing into the classroom: experiential learning via software development project
Clarity about future of the team			
	Casey et al. 2006	Casey, Valentine; Richardson, Ita	Uncovering the reality within virtual software teams
PA 2.4.4 Awareness			
Establish and Maintain Trust			
	Jain N IEEE Comput	Jain N	Offshore agile maintenance
	Babar MA et al. 80, no	Babar MA; Verner JM; Nguyen Phong Thanh	Establishing and maintaining trust in software outsourcing relationships: an empirical investigation
	Oza NV et al. 48, no	Oza NV; Hall T; Rainer A; Grey S	Trust in software outsourcing relationships: An empirical investigation of Indian software companies
	Nakayama et al. 2006	Nakayama, Marina; Binotto, Erlaine; Pilla, Bianca	Trust in Virtual Teams: A Performance Indicator
	Moe et al. 2007	Moe, Nils; Šmite, Darja	Understanding Lacking Trust in Global Software Teams: A Multicase Study
	Mezgar 2006	Mezgar, István	Trust Building for Enhancing Collaboration in Virtual Organiza-

			tions
	Maloney et al. 2006	Maloney, Mary; Zellmer-Bruhn, Mary	Building bridges, windows and cultures: Mediating mechanisms between team heterogeneity and performance in global teams
	Hole et al. 2008	Hole, Steinar; Moe, Nils Brede	A Case Study of Coordination in Distributed Agile Software Development
	Braithwaite et al. 2005	Braithwaite, Keith; Joyce, Tim	XP Expanded: Distributed Extreme Programming
	Sangwan et al. 2008	Sangwan, Raghvinder S.; Ros, Johannes	Architecture leadership and management in globally distributed software development
	Mikawa et al. 2009	Mikawa, Suzanne P.; Cunnington, Sharon K.; Gaskins, Scott A.	Removing barriers to trust in distributed teams: understanding cultural differences and strengthening social ties
	Huang et al. 2006	Huang, Haiyan; Ocker, Rosalie	Preliminary insights into the in-group/outgroup effect in partially distributed teams: an analysis of participant reflections
	Herbsleb et al. 2005	Herbsleb, James D.; Paulish, Daniel J.; Bass, Matthew	Global software development at siemens: experience from nine projects
	Casey et al. 2006	Casey, Valentine; Richardson, Ita	Uncovering the reality within virtual software teams
	Alexander 2002	Alexander, Patricia M.	Teamwork, time, trust and information
awareness has to be connected to the process models			
	Nakayama et al. 2006	Nakayama, Marina; Binotto, Erlaine; Pilla, Bianca	Trust in Virtual Teams: A Performance Indicator
	Leinonen et al. 2005	Leinonen, Piritta; Järvelä, Sanna; Häkkinen, Päivi	Conceptualizing the Awareness of Collaboration: A Qualitative Study of a Global Virtual Team
	Estublier et al. 2005	Estublier, Jacky; Garcia, Sergio	Process model and awareness in SCM
Identify the persons to talk to			

	Gutwin et al. 2004	Gutwin, Carl; Penner, Reagan; Schneider, Kevin	Group awareness in distributed software development
	Cataldo et al. 2008	Cataldo, Marcelo; Herbsleb, James D.	Communication networks in geographically distributed software development
Virtual co-location			
	Mullally et al. 2009	Mullally, Brenda; Stapleton, Larry	Methodology Usage by Virtual Information Systems Development Teams
	Lings et al. 2006	Lings, Brian; Lundell, Björn; Ågerfalk, Pär; Fitzgerald, Brian	Ten Strategies for Successful Distributed Development
	Hwang et al. 2006	Hwang, Jane; Kim, Namgyu; Kim, Gerard	Immersing Tele-operators in Collaborative Augmented Reality
	Hole et al. 2008	Hole, Steinar; Moe, Nils Brede	A Case Study of Coordination in Distributed Agile Software Development
	Braithwaite et al. 2005	Braithwaite, Keith; Joyce, Tim	XP Expanded: Distributed Extreme Programming
	Herbsleb 2007	Herbsleb, James D.	Global Software Engineering: The Future of Socio-technical Coordination
	Boyer et al. 1998	Boyer, David G.; Handel, Mark J.; Herbsleb, James	Virtual community prescence awareness
	Begel 2008	Begel, Andrew	Effecting change: coordination in large-scale software development
PA 2.4.5 Distributed Project Management			
Benefits of going global in Software Development			
	Ågerfalk et al. 2008	Ågerfalk, Pär; Fitzgerald, Brian; Holmström Olsson, Helena; Ó Conchúir, Eoin	Benefits of Global Software Development: The Known and Unknown
Think global, act local			
	2008		ABB: Management of Technology: Think Global, Act Local
Decision Support and Management Techniques for Distributed projects			
	Mann 2009	Mann, A.	Information Technology and the Related Services Industry: Evaluating India's Suc-

		cess Factors
Mei He et al. 2008	Mei He; Mingshu Li; Qing Wang; Ye Yang; Kai Ye	An investigation of software development productivity in China
Leonardo Pilatti et al. 2006	Leonardo Pilatti; Jorge Luis Nicolas Audy	Global Software Development Offshore Insourcing Organizations Characteristics: Lessons Learned from a Case Study
Jiyang Xu et al. 2007	Jiyang Xu; Lippert, D.	Lesson Learned in Managing IT Departments
Roehling ST et al. 5, no	Roehling ST; Collofello JS; Hermann BG; Smith Daniels DE	System dynamics modeling applied to software outsourcing decision support
Prikladnicki R et al. IEEE	Prikladnicki R; Evaristo R; Damian D; Audy JLN	Conducting qualitative research in an international and distributed research team: challenges and lessons learned
Prikladnicki R et al. 8, no	Prikladnicki R; Audy JLN; Evaristo R	Global software development in practice lessons learned
Nair KGK et al. 21, no	Nair KGK; Prasad PN	Offshore outsourcing: a swot analysis of a state in India
Jain N IEEE Comput	Jain N	Offshore agile maintenance
Biro M et al. 2005	Biro M; Feher P	Forces affecting offshore software development
Muhsin et al. 2008	Muhsin, Z. J.; El-Mousa, A. H.; Al-Taee, M. A.	A distributed workflow management system utilizing computational intelligence for decision support
Setamanit et al. 2006	Setamanit, Siri-on; Wakeland, Wayne; Raffo, David	Exploring the Impact of Task Allocation Strategies for Global Software Development Using Simulation
Siva Namasivayam 2004	Siva Namasivayam	Profiting from business process outsourcing
Ralyte et al. 2008	Ralyte, J.; Lamielle, X.; Arni-Bloch, N.; Leonard, M.	A framework for supporting management in distributed information systems development

	Matsubara 2001	Matsubara, T.	Japan: a huge IT consumption market
	Langford 2008	Langford, L. K.	Surf"s Up: The Project Management Process
	Cusumano 2008	Cusumano, Michael A.	Managing software development in globally distributed teams
Distributed Agile			
	Välimäki et al. 2008	Välimäki, A.; Kääriäinen, J.	Patterns for Distributed Scrum — A Case Study
	Sureshchandra et al. 2008	Sureshchandra, K.; Shrinivasavadhani, J.	Adopting Agile in Distributed Development
	Nisar et al. 2004	Nisar, M. F.; Hameed, T.	Agile methods handling offshore software development issues
	Urdangarin R et al. IEEE	Urdangarin R; Fernandes P; Avritzer A; Paulish D	Experiences with agile practices in the global studio project
	Kussmaul C et al. 4th Conference on Extreme Programming and Agile Methods	Kussmaul C; Jack R; Sponsler B	Outsourcing and offshoring with agility: a case study
	Taylor et al. 2006	Taylor, Philip; Des Greer; Sage, Paul; Coleman, Gerry; McDaid, Kevin; Lawthers, Ian; Corr, Ronan	Applying an Agility/Discipline Assessment for a Small Software Organisation
	Sharp et al. 2008	Sharp, Jason H.; Ryan, Sherry D.	A Preliminary Conceptual Model for Exploring Global Agile Teams
	Morgan et al. 2007	Morgan, Robert; Walny, Jagoda; Kolenda, Henning; Ginez, Estaban; Maurer, Frank	Using Horizontal Displays for Distributed and Collocated Agile Planning
	Hole et al. 2008	Hole, Steinar; Moe, Nils Brede	A Case Study of Coordination in Distributed Agile Software Development
	Hazzan et al. 2008	Hazzan, Orit; Dubinsky, Yael	Globalization
	Rees 2002	Rees, M. J.	A feasible user story tool for agile software development?
	Karlsson et al. 2000	Karlsson, Even-André; Andersson, Lars-Göran; Leion, Per	Daily build and feature development in large distributed projects
PA 2.5 Documentation Process			
PA 2.5.1 Design and development			

summaries are important for distributed teams		
Ocker et al. 2008	Ocker, Rosalie J.; Fjermestad, Jerry	Communication differences in virtual design teams: findings from a multi-method analysis of high and low performing experimental teams
Distribiuted Software Design Meetings		
Dekel 2005	Dekel, Uri	Supporting distributed software design meetings: what can we learn from co-located meetings?
Chira et al. 2006	Chira, O.; Chira, C.; Roche, T.; Tormey, D.; Brennan, A.	An agent-based approach to knowledge management in distributed design
Seperation of Design and Development		
Cordes et al. 2007	Cordes, Richard E.; Spine, Thomas M.	Transcending organizational boundaries: virtual team approach in UI guideline development
Detecting Software Conflicts		
Dewan 2008	Dewan, Prasun	Dimensions of tools for detecting software conflicts
Tools for distributed design		
Farenhorst et al. 2007	Farenhorst, Rik; Lago, Patricia; van Vliet, Hans	Effective Tool Support for Architectural Knowledge Sharing
Chira et al. 2006	Chira, O.; Chira, C.; Roche, T.; Tormey, D.; Brennan, A.	An agent-based approach to knowledge management in distributed design
Smith et al. 2005	Smith, J. L.; Bohner, S. A.; McCrickard, D. S.	Toward introducing notification technology into distributed project teams

PA 2.5.2 Production
PA 2.5.3 Maintenance

PA 3 Engineering

PA 3.1 Requirements Management/Development
PA 3.1.1 Develop Customer Requirements
PA 3.1.1.1 Elicit Needs

Success or failure of the project depends on keeping customers satisfied		
Carver et al. 2007	Carver, Jeffrey C.; Kendall, Richard P.; Squires, Susan E.; Post, Douglass E.	Software Development Environments for Scientific and Engineering Software: A Series of

		Case Studies
Techniques for Requirements Elicitation		
Cheng et al. 2009	Cheng, Betty H. C.; Atlee, Joanne M.	Current and Future Research Directions in Requirements Engineering
Aranda et al. 2006	Aranda, Gabriela; Vizcaíno, Aurora; Cechich, Alejandra; Piattini, Mario	CHOOSING GROUPWARE TOOLS AND ELICITATION TECHNIQUES ACCORDING TO STAKEHOLDERS' FEATURES
Prikladnicki R et al. 8, no	Prikladnicki R; Audy JLN; Evaristo R	Global software development in practice lessons learned
Geisser et al. 2006	Geisser, Michael; Hildenbrand, Tobias	A Method for Collaborative Requirements Elicitation and Decision-Supported Requirements Analysis
Aranda et al. 2007	Aranda, Gabriela; Vizcaíno, Aurora; Cechich, Alejandra; Piattini, Mario	How to Choose Groupware Tools Considering Stakeholders' Preferences During Requirements Elicitation?
Damian et al. 2006	Damian, Daniela; Lanubile, Filippo; Mallardo, Teresa	The role of asynchronous discussions in increasing the effectiveness of remote synchronous requirements negotiations
Performance competes with other important goals		
Carver et al. 2007	Carver, Jeffrey C.; Kendall, Richard P.; Squires, Susan E.; Post, Douglass E.	Software Development Environments for Scientific and Engineering Software: A Series of Case Studies
Consumer-Centric Service-Oriented Architecture		
Tsai et al. 2006	Tsai, W.; Xiao, Bingnan; Huang, Qian; Chen, Yinong	Collaborative software design in an SOA environment
How to successfully organize (virtual) requirements meetings		
Fricker et al. 2008	Fricker, Samuel; Grünbacher, Paul	Negotiation Constellations – Method Selection Framework for Requirements Negotiation

	Fricker et al. 2007	Fricker, Samuel; Gorschek, Tony; Myllyperkiö, Petri	Handshaking Between Software Projects and Stakeholders Using Implementation Proposals
	Aranda et al. 2005	Aranda, Gabriela N.; Vizcaíno, Aurora; Cechich, Alejandra; Piattini, Mario	A Cognitive Perspective for Choosing Groupware Tools and Elicitation Techniques in Virtual Teams
	Damian 2001	Damian, D.	An empirical study of requirements engineering in distributed software projects: is distance negotiation more effective?
PA 3.1.1.2 Develop Requirements			
Negotiation constellations			
	Damian D 24, no	Damian D	Stakeholders in global requirements engineering: Lessons learned from practice
	Fricker et al. 2008	Fricker, Samuel; Grünbacher, Paul	Negotiation Constellations – Method Selection Framework for Requirements Negotiation
	Fricker et al. 2007	Fricker, Samuel; Gorschek, Tony; Myllyperkiö, Petri	Handshaking Between Software Projects and Stakeholders Using Implementation Proposals
Evolution of Customer Demands			
	2008		Customer Challenges and Demands
Media for requirements communication			
	Geisser et al. 2006	Geisser, Michael; Hildenbrand, Tobias	A Method for Collaborative Requirements Elicitation and Decision-Supported Requirements Analysis
	Aranda et al. 2007	Aranda, Gabriela; Vizcaíno, Aurora; Cechich, Alejandra; Piattini, Mario	How to Choose Groupware Tools Considering Stakeholders' Preferences During Requirements Elicitation?
	Herbsleb 2007	Herbsleb, James D.	Global Software Engineering: The Future of Socio-technical Coordination
PA 3.1.2 Develop Product Requirements			

PA 3.1.2.1 Establish Product and Product Component Requirements		
PA 3.1.2.2 Allocate Product Component Requirements		
PA 3.1.2.3 Identify Interface Requirements		
PA 3.1.3 Analyze and Validate Requirements		
PA 3.1.3.1 Establish Operational Concepts and Scenarios		
Process Improvement		
Nikula U et al. 2005	Nikula U; Sajaniemi J	Tackling the complexity of requirements engineering process improvement by partitioning the improvement task
Ning et al. 2006	Ning, Anliang; Hou, Hong; Hua, Qingyi; Yu, Bin; Hao, Kegang	Requirements Engineering Processes Improvement: A Systematic View
Siemens Requirements State Model		
Mikulovic et al. 2006	Mikulovic, Vesna; Heiss, Michael	How do I know what I have to do?": the role of the inquiry culture in requirements communication for distributed software development projects
PA 3.1.3.2 Establish a Definition of Required Functionality		
PA 3.1.3.3 Analyze Requirements		
Decision supported Requirements Analysis		
Geisser et al. 2006	Geisser, Michael; Hildenbrand, Tobias	A Method for Collaborative Requirements Elicitation and Decision-Supported Requirements Analysis
PA 3.1.3.5 Validate Requirements		
PA 3.1.4 Manage Requirements		
PA 3.1.4.1 Obtain an Understanding of Requirements		
Techniques and Tools		
Kedian Mu et al. 2007	Kedian Mu; Weiru Liu; Zhi Jin; Ruqian Lu; Anbu Yue; Bell D	A merging-based approach to handling inconsistency in locally prioritized software requirements
Hawlitzky N et al. 2000	Hawlitzky N; Munnich A; Nothhelfer Kolb B; Rappl M	Managing distributed software development-a portfolio based methodology
Välimäki et al. 2007	Välimäki, Antti; Kääriäinen, Jukka	Requirements Management Practices as Patterns for Distributed Product Management
Fricker et al. 2007	Fricker, Samuel; Gorschek, Tony;	Handshaking Between Software Projects and

		Myllyperkiö, Petri	Stakeholders Using Implementation Proposals
	Bégin et al. 2007	Bégin, Marc-Elian; Sancho, Guillermo; Di Meglio, Alberto; Ferro, Enrico; Ronchieri, Elisabetta; Selmi, Matteo; Żurek, Marian	Build, Configuration, Integration and Testing Tools for Large Software Projects: ETICS
	Seybold et al. 2008	Seybold, C.; Keller, R. K.	Aligning Software Maintenance to the Offshore Reality
	Heindl et al. 2006	Heindl, Matthias; Biffl, Stefan	Risk management with enhanced tracing of requirements rationale in highly distributed projects
	Clerc 2008	Clerc, Viktor	Towards architectural knowledge management practices for global software development
	Conceptual Framework for Requirements evolution in Outsourcing Context		
	Lormans et al. 2004	Lormans, M.; van Dijk, H.; van Deursen, A.; Nocker, E.; Zeeuw, A. de	Managing evolving requirements in an outsourcing context: an industrial experience report
	Siemens Requirements State Model		
	Mikulovic et al. 2006	Mikulovic, Vesna; Heiss, Michael	How do I know what I have to do?": the role of the inquiry culture in requirements communication for distributed software development projects
PA 3.1.4.2 Obtain Commitment to Requirements PA 3.1.4.3 Manage Requirements Changes			
	Techniques and Tools		
	Välimäki et al. 2007	Välimäki, Antti; Kääriäinen, Jukka	Requirements Management Practices as Patterns for Distributed Product Management
	Fricker et al. 2007	Fricker, Samuel; Gorschek, Tony; Myllyperkiö, Petri	Handshaking Between Software Projects and Stakeholders Using Implementation Proposals
	Seybold et al. 2008	Seybold, C.; Keller, R. K.	Aligning Software Maintenance to the

			Offshore Reality
		Heindl et al. 2006	Heindl, Matthias; Biffl, Stefan	Risk management with enhanced tracing of requirements rationale in highly distributed projects
		Clerc 2008	Clerc, Viktor	Towards architectural knowledge management practices for global software development
	Ensure the correctness and tracability of the requirements documents			
		Kommeren et al. 2007	Kommeren, Rob; Parviainen, Päivi	Philips experiences in global distributed software development
		Edgar 2006	Edgar, D. R.	Lessons Learned: Outsourcing Support Equipment Development
	PA 3.1.4.4 Maintain Bidirectional Traceability of Requirements			
	Techniques and Tools			
		Fricker et al. 2007	Fricker, Samuel; Gorschek, Tony; Myllyperkiö, Petri	Handshaking Between Software Projects and Stakeholders Using Implementation Proposals
		Heindl et al. 2006	Heindl, Matthias; Biffl, Stefan	Risk management with enhanced tracing of requirements rationale in highly distributed projects
	PA 3.1.4.5 Identify Inconsistencies Between Project Work and Requirements			
	Techniques and Tools Siemens Requirements State Model			
		Mikulovic et al. 2006	Mikulovic, Vesna; Heiss, Michael	How do I know what I have to do?": the role of the inquiry culture in requirements communication for distributed software development projects
PA 3.2 Technical Solution				
PA 3.2.1 Select Product Component Solutions				
PA 3.2.1.1 Develop Alternative Solutions and Selection Criteria PA 3.2.1.2 Select Product Component Solutions				
PA 3.2.2 Develop the Design				
PA 3.2.2.1 Design the Product or Product Component				
Tactics for improving architecture/organization fit				
		Herbsleb 2007	Herbsleb, James D.	Global Software Engineering: The Future of

		Socio-technical Coordination
PA 3.2.2.2 Establish a Technical Data Package		
PA 3.2.2.3 Design Interfaces Using Criteria		
PA 3.2.2.4 Perform Make, Buy, or Reuse Analyses		
PA 3.2.3 Implement the Product Design		
PA 3.2.3.1 Implement the Design		
PA 3.2.3.2 Develop Product Support Documentation		
PA 3.3 Product Integration		
PA 3.3.1 Prepare for Product Integration		
PA 3.3.1.1 Determine Integration Sequence		
PA 3.3.1.2 Establish the Product Integration Environment		
PA 3.3.1.3 Establish Product Integration Procedures and Criteria		
PA 3.3.2 Ensure Interface Compatibility		
PA 3.3.2.1 Review Interface Descriptions for Completeness		
PA 3.3.2.2 Manage Interfaces		
Updates in Component based Software		
Nguyen 2008	Nguyen, T. N.	
PA 3.3.3 Assemble Product Components and Deliver the Product		
PA 3.3.3.1 Confirm Readiness of Product Components for Integration		
PA 3.3.3.2 Assemble Product Components		
Involve the supplier prior to the assemble		
Edgar 2006	Edgar, D. R.	Lessons Learned: Outsourcing Support Equipment Development
PA 3.3.3.3 Evaluate Assembled Product Components		
PA 3.3.3.4 Package and Deliver the Product or Product Component		
PA 3.4 Validation		
PA 3.4.1 Prepare for Validation		
PA 3.4.1.1 Select Products for Validation		
PA 3.4.1.2 Establish the Validation Environment		
PA 3.4.1.3 Establish Validation Procedures and Criteria		
PA 3.4.2 Validate Product or Product Components		
PA 3.4.2.1 Perform Validation		
PA 3.4.2.2 Test		
Rapid integration and Testing		
Carter 1997	Carter, C. J.	Rapid integration and testing of business solutions
Test Case Generation		
Arantes et al. 2008	Arantes, Alessandro Oliveira; Vijaykumar, Nandamudi Lankalapalli; Santiago Junior, Valdivino Alexandre de; Guimaraes, Danielle	WEB-PerformCharts: a collaborative web-based tool for test case generation from statecharts

Tools and techniques			
	Bégin et al. 2007	Bégin, Marc-Elian; Sancho, Guillermo; Di Meglio, Alberto; Ferro, Enrico; Ronchieri, Elisabetta; Selmi, Matteo; Żurek, Marian	Build, Configuration, Integration and Testing Tools for Large Software Projects: ETICS
	Varma 2005	Varma, P.	Stub Libraries for Software Migration and Development
Community of Practice Experience from Brasil			
	Andrade Paiva E IEEE Computer Society	Andrade Paiva E	The test community of practice experience in Brazil
Test Strategy			
	Varma 2005	Varma, P.	Stub Libraries for Software Migration and Development
	Seybold et al. 2008	Seybold, C.; Keller, R. K.	Aligning Software Maintenance to the Offshore Reality
	Martin et al. 2008	Martin, David; Rooksby, John; Rouncefield, Mark; Sommerville, Ian	Cooperative work in software testing
Test Automation			
	Berner et al. 2005	Berner, Stefan; Weber, Roland; Keller, Rudolf K.	Observations and lessons learned from automated testing
PA 3.4.2.3 Analyze Validation Results			
PA 3.5 Safety Engineering			
PA 3.5.1 Identify Hazards, Accidents and Sources of Hazards			
PA 3.5.2 Analyse Hazards and Perform Risk Assessment			
PA 3.5.3 Define and Maintain Safety Requirements			
Design for Safety			
Modularisierung for Safety			
	confidential		
PA 3.5.4 Verification			
PA 3.5.5 Safety Analysis			
PA 3.5.6 HW/SW Integration			
PA 3.5.7 System Integration / Verification			
PA 4 Support			
PA 4.1 Causal Analysis and Resolution			
PA 4.1.1 Determine Causes of Defects			
PA 4.1.1.1 Select Defect Data for Analysis			
Software Reliability Growth Models			
	Kapur et al. 2008	Kapur, P. K.; Jha, P. C.; Singh, V. B.	On the Development of Discrete Software Reliability Growth

		Models
PA 4.1.1.2 Analyze Causes		
PA 4.1.2 Address Causes of Defects		
PA 4.1.2.1 Implement the Action Proposals		
PA 4.1.2.2 Evaluate the Effect of Changes		
PA 4.1.2.3 Record Data		
PA 4.2 Decision Analysis and Resolution		
PA 4.2.1 Evaluate Alternatives		
PA 4.2.1.1 Establish Guidelines for Decision Analysis		
PA 4.2.1.2 Establish Evaluation Criteria		
PA 4.2.1.3 Identify Alternative Solutions		
PA 4.2.1.4 Select Evaluation Methods		
PA 4.2.1.5 Evaluate Alternatives		
PA 4.2.1.6 Select Solutions		
PA 4.3 Process and Product Quality Assurance		
PA 4.3.1 Objectively Evaluate Processes and Work Products		
PA 4.3.1.1 Objectively Evaluate Processe		
Evaluate Process Models for Distributed Software Development		
Yalaho et al. 2007	Yalaho, A.; Nahar, N.	The ICT-Supported Unified Process Model of Offshore Outsourcing of Software Production: Exploratory Examination and Validation
Success indicators for SPI		
Macke et al. 2008	Macke, Dirk; Galinac, Tihana	Optimized Software Process for Fault Handling in Global Software Development
Conflict Management for Model based SE		
Cicchetti et al. 2008	Cicchetti, Antonio; Di Ruscio, Davide; Pierantonio, Alfonso	Managing Model Conflicts in Distributed Development
Cost of Processes		
Coleman et al. 2006	Coleman, Gerry; O'Connor, Rory	Software Process in Practice: A Grounded Theory of the Irish Software Industry
Evaluate Communication Patterns in virtual teams		
Cicchetti et al. 2008	Cicchetti, Antonio; Di Ruscio, Davide; Pierantonio, Alfonso	Managing Model Conflicts in Distributed Development
Sengupta et al. 1998	Sengupta, K.; Leon Zhao, J.	Designing workflow management systems for virtual organizations: an empirically grounded approach
PA 4.3.1.2 Objectively Evaluate Work Products and Services		

PA 4.3.2 Provide Objective Insight		
PA 4.3.2.1 Communicate and Ensure Resolution of Noncompliance Issues		
PA 4.3.2.2 Establish Records		
PA 4.4 Acquisition Process		
PA 4.4.1 Acquisition preparation		
Auditing Model		
Evaristo et al. 2005	Evaristo, R.; Nicolas, J. L.; Prikladnicki, R.; Avritchir, J.	Wholly Owned Offshore Subsidiaries for IT Development: A Program of Research
Decide where to offshore to		
Evaristo et al. 2005	Evaristo, R.; Nicolas, J. L.; Prikladnicki, R.; Avritchir, J.	Wholly Owned Offshore Subsidiaries for IT Development: A Program of Research
PA 4.4.2 Supplier selection		
Supplier Agreement Management		
Kumar S et al. Knowledge	Kumar S; Aquino EC; Anderson E	Application of a process methodology and a strategic decision model for business process outsourcing
2007		Controlling for IT and Software
Junchao Xiao et al. 2005	Junchao Xiao; Qing Wang	Contract-driven cooperation development process for software outsourcing
Ebert 2007	Ebert, C.	Optimizing Supplier Management in Global Software Engineering
Make or Buy		
Coman A et al. 38, no	Coman A; Ronen B	Production outsourcing: a linear programming model for the theory-of-constraints
Establish trust		
Siakas KV et al. 2006	Siakas KV; Maoutsidis D; Siakas E	Trust facilitating good software outsourcing relationships
Oza NV et al. 48, no	Oza NV; Hall T; Rainer A; Grey S	Trust in software outsourcing relationships: An empirical investigation of Indian software companies
Babar MA et al. IEEE	Babar MA; Niazi M	Implementing software process improvement initiatives: an analysis of Vietnamese practitioners' views
Outsourcing Decision		

	Huang SH et al. 105, no	Huang SH; Keskar H	Comprehensive and configurable metrics for supplier selection
	Kumar S et al. Knowledge	Kumar S; Aquino EC; Anderson E	Application of a process methodology and a strategic decision model for business process outsourcing
	Aksin OZ et al. 26, no	Aksin OZ; Masini A	Effective strategies for internal outsourcing and offshoring of business services: an empirical investigation
	Beyah et al. 2001	Beyah, G.; Gallivan, M.	Knowledge management as a framework for understanding public sector outsourcing
Audit			
	Kumar S et al. Knowledge	Kumar S; Aquino EC; Anderson E	Application of a process methodology and a strategic decision model for business process outsourcing
	Huang SH et al. 105, no	Huang SH; Keskar H	Comprehensive and configurable metrics for supplier selection
	Kommeren et al. 2007	Kommeren, Rob; Parviainen, Päivi	Philips experiences in global distributed software development
	Iwami 2008	Iwami, Y.	How to Measure Quality of Software Developed by Subcontractors (Short Paper)
	Evaristo et al. 2005	Evaristo, R.; Nicolas, J. L.; Prikladnicki, R.; Avritchir, J.	Wholly Owned Offshore Subsidiaries for IT Development: A Program of Research
Select primary and secondary supplier			
	Hawlitzky N et al. 2000	Hawlitzky N; Munnich A; Nothhelfer Kolb B; Rappl M	Managing distributed software development-a portfolio based methodology
	Edgar 2006	Edgar, D. R.	Lessons Learned: Outsourcing Support Equipment Development
	Ebert 2007	Ebert, C.	Optimizing Supplier Management in Global Software Engineering
	Cao Ying et al. 2008	Cao Ying; Sun Fuquan; Wang Lina; Chen Tingbin	Promotion of Offshore Software Outsourcing Enterprise Value Chain

		Based on Knowledge Transfer Effects
PA 4.4.3 Supplier monitoring		
Supplier Relationship Management		
Choy KL et al. 17, no	Choy KL; Lee WB; Lau HCW; Dawei Lu; Lo V	Design of an intelligent supplier relationship management system for new product development
Supplier Risk Management		
Smite D et al. 2006	Smite D; Borzovs J	A framework for overcoming supplier related threats in global projects
Maintain Trust		
Siakas KV et al. 2006	Siakas KV; Maoutsidis D; Siakas E	Trust facilitating good software outsourcing relationships
Oza NV et al. 48, no	Oza NV; Hall T; Rainer A; Grey S	Trust in software outsourcing relationships: An empirical investigation of Indian software companies
Babar MA et al. IEEE	Babar MA; Niazi M	Implementing software process improvement initiatives: an analysis of Vietnamese practitioners' views
importance of regular status reports		
Edgar 2006	Edgar, D. R.	Lessons Learned: Outsourcing Support Equipment Development
PA 4.4.4 Customer acceptance		
PA 4.5 Maintenance Process		
PA 4.5.1 Problem and modification analysis		
Knowledge of the Effort Drivers has a significant effect on effort estimates		
Rao et al. 2003	Rao, B. S.; Sarda, N. L.	Effort drivers in maintenance outsourcing-an experiment using Taguchi's methodology
Sabetzadeh et al. 2008	Sabetzadeh, Mehrdad; Nejati, Shiva; Easterbrook, Steve; Chechik, Marsha	Global consistency checking of distributed models with TReMer+
PA 4.5.2 Modification implementation		
PA 4.5.3 Maintenance review/acceptance		
PA 4.5.4 Migration		
PA 4.6 Joint Review Process		

PA 4.6.1 Project Management Reviews
PA 4.6.2 Technical Reviews

Appendix D: SLR Ratings

Reference	Rating of		
	Context	Design	Validation
SLR2	1	2	1
SLR3	2	2	1
SLR4	1	1	1
SLR5	1	2	1
SLR6	2	2	1
SLR7	0	2	2
SLR8	2	2	1
SLR9	0	1	1
SLR10	1	1	0
SLR11	2	1	1
SLR127	1	1	1
SLR17	1	1	1
SLR16	2	1	2
SLR18	1	1	1
SLR20	1	1	2
SLR23	1	1	1
SLR22	2	2	2
SLR24	2	2	1
SLR25	2	1	2
SLR26	1	1	1
SLR27	2	2	1
SLR28	0	0	0
SLR29	2	3	1
SLR32	2	3	2
SLR33	1	1	2
SLR34	0	0	0
SLR36	2	1	3
SLR37	1	1	1
SLR38	2	2	1
SLR39	0	0	1
SLR41	3	2	3
SLR42	1	1	1
SLR44	2	1	1
SLR45	2	2	1
SLR48	2	2	2
SLR47	1	2	2
SLR49	1	1	1
SLR50	1	1	1
SLR51	1	1	1
SLR52	2	2	1

SLR53	2	2	2
SLR54	2	2	2
SLR55	1	1	1
SLR58	1	1	1
SLR60	1	2	2
SLR61	2	1	2
SLR62	3	3	3
SLR63	1	1	2
SLR66	2	2	1
SLR65	2	2	2
SLR67	2	2	2
SLR68	1	1	2
SLR70	2	3	2
SLR73	1	1	1
SLR74	1	1	1
SLR76	0	0	0
SLR75	2	2	2
SLT78	1	1	0
SLR79	2	2	3
SLR80	2	1	1
SLR81	2	2	2
SLR82	2	2	2
SLR83	1	1	1
SLR85	0	1	3
SLR89	0	0	3
SLR90	1	1	1
SLR95	1	1	2
SLR93	1	1	2
SLR91	2	2	1
SLR92	1	1	2
SLR97	1	1	2
SLR98	1	1	1
SLR99	3	3	2
SLR100	2	2	1
SLR101	1	1	1
SLR102	2	2	1
SLR103	2	2	1
SLR104	1	1	1
SLR106	2	2	1
SLR107	1	1	1
SLR108	2	2	2
SLR110	1	1	1
SLR112	2	2	1
SLR113	2	2	2
SLR114	2	3	2

SLR116	3	3	2
SLR117	1	1	1
SLR118	1	1	1
SLR119	1	0	0
SLR120	1	1	1
SLR121	1	1	1
SLR122	2	2	1
SLR123	3	3	3
SLR124	3	2	3
SLR126	2	1	1
SLR127	2	2	1
SLR129	2	2	1
SLR130	1	1	1
SLR131	2	2	1
SLR132	1	1	1
SLR133	0	0	0
SLR134	1	1	2
SLR135	3	2	2
SLR136	1	1	1
SLR137	2	1	1
SLR138	3	2	2
SLR139	2	2	2
SLR140	2	2	2
SLR141	2	1	1
SLR142	2	1	1
SLR143	1	1	2
SLR144	1	1	1
SLR146	0	0	0
SLR148	1	1	1
SLR150	0	0	0
SLR151	2	2	2
SLR152	0	0	0
SLR155	1	1	1
SLR156	3	3	3
SLR157	1	1	0
SLR158	2	1	1
SLR159	1	2	1
SLR160	2	2	1
SLR161	3	3	2
SLR164	2	1	3
SLR166	2	2	3
SLR168	2	2	2
SLR167	2	2	1
SLR169	1	1	1
SLR171	2	1	1

SLR174	1	1	2
SLR173	1	1	1
SLR175	2	2	3
SLR177	1	1	3
SLr178	1	1	1
SLR180	1	2	1
SLR181	2	2	2
SLR183	2	1	1
SLR184	2	2	2
SLR185	1	0	1
SLR186	1	1	2
SLR187	1	1	2
SLR188	1	1	1
SLR190	2	1	3
SLR192	2	2	2
SLR193	2	2	2
SLR196	2	1	1
SLR197	1	1	1
SLR198	2	1	1
SLR199	0	1	1
SLR201	1	1	1
SLR203	0	1	0
SLR204	2	2	2
SLR206	1	2	3
SLR207	2	2	2
SLR208	2	2	1
SLR210	2	2	2
SLR211	2	1	1
SLR213	1	1	2
SLR214	1	1	1
SLR215	2	1	2
SLR216	2	2	2
SLR219	2	2	1
SLR221	1	1	1
SLR223	0	0	0
SLR224	1	1	2
SLR225	1	1	1
SLR226	1	1	1
SLR227	1	1	2
SLR228	1	1	1
SLR229	2	1	2
SLR230	1	1	1
SLR231	1	2	2
SLR232	1	1	1
SLR233	1	1	1

SLR234	2	2	1
SLR235	1	1	1
SLR236	1	1	1
SLR238	2	2	1
SLR239	1	2	2
SLR241	1	1	1
SLR242	1	1	2
SLR243	1	1	1
SLR246	1	1	1
SLR245	2	3	1
SLR244	0	0	1
SLR247	1	1	1
SLR248	1	2	2
SLR249	1	0	1
SLR253	2	2	2
SLR255	1	1	2
SLR258	1	1	1
SLR259	1	1	1
SLR260	2	2	2
SLR261	1	1	1
SLR262	1	1	2
SLR263	2	2	2
SLR265	1	1	1
SLR 266	2	1	2
SLR267	2	2	1
SLR270	1	2	2
SLR269	2	2	2
SLR272	1	1	1
SLR273	2	2	2
SLR274	2	2	2
Slr275	2	3	3
SLR276	1	0	0
SLR277	0	0	0
SLR278	1	1	1
SLR279	1	1	1
SLR280	2	2	1
SLR281	1	1	2
SLR282	1	1	1
SLR284	2	2	1
SLR285	0	0	0
SLR286	2	2	1
SLR287	2	2	2
SLR288	2	2	1
SLR289	2	1	2
SLR290	2	1	3

SLR291	2	1	1
SLR292	1	1	1
SLR293	1	1	0
SLR295	1	1	1
SLR296	1	1	1
SLR298	1	1	1
SLR297	2	1	2
SLR300	1	1	1
SLR299	1	1	1
SLR302	2	1	1
SLR303	1	1	3
SLR304	1	1	1
SLR305	1	1	1
SLR306	2	1	1
SLR307	2	2	2
SLR308	1	1	2
SLR309	1	1	1
SLR310	1	1	1
SLR312	1	1	2
SLR313	2	2	1
SLR314	2	2	1
SLR315	2	2	2
SLR316	1	1	1
SLR317	2	2	2
SLR318	2	2	1
SLR320	2	2	1
SLR323	0	0	2
SLR324	1	1	1
SLR325	1	2	1
SLR326	2	1	2
SLR327	2	2	2
SLR328	1	1	2
SLR330	1	1	1
SLR333	1	2	1
SLR335	1	1	1
SLR334	1	1	1
SLR336	1	1	1
SLR337	2	2	2
SLR338	2	2	1
SLR339	1	1	1
SLR1	1	1	1

Appendix E: List of SLR Papers

[SLR1] "ABB: Management of Technology: Think Global, Act Local," *Managing Global Innovation*, pp. 559–571, 2008.

[SLR2] Abrahão, S., Poels, G., and Pastor, O., "A functional size measurement method for object-oriented conceptual schemas: design and evaluation issues," *Software and Systems Modeling*, vol. 5, no. 1, pp. 48–71, Apr. 2006.

[SLR3] Adya, M., Nath, D., Malik, A., and Sridhar, V., "Bringing global sourcing into the classroom: experiential learning via software development project," 2007. In *SIGMIS-CPR '07: Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel research*, 20-27. New York, NY, USA: ACM.

[SLR4] Ågerfalk, P., Fitzgerald, B., Holmström Olsson, H., and Ó Conchúir, E., " Benefits of Global Software Development: The Known and Unknown," *Making Globally Distributed Software Development a Success Story*, pp. 1–9, 2008.

[SLR5] Aiello, G., Alessi, M., Cossentino, M., Urso, A., and Vella, G., "RTDWD: Real-Time Distributed Wideband-Delphi for User Stories Estimation," *Rapid Integration of Software Engineering Techniques*, pp. 35–50, 2007.

[SLR6] Aksin OZ and Masini A, "Effective strategies for internal outsourcing and offshoring of business services: an empirical investigation," *Journal of Operations Management, vol*, no. Elsevier Science B.V., pp. vol, 26, no.

[SLR7] Al-Ani, B., Trainer, E., Ripley, R., Sarma, A., van der Hoek, A., and Redmiles, D., "Continuous coordination within the context of cooperative and human aspects of software engineering," 2008. In *CHASE '08: Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering*, 1-4. New York, NY, USA: ACM.

[SLR8] Alexander, P. M., "Teamwork, time, trust and information," 2002. In *SAICSIT '02: Proceedings of the 2002 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology*, 65-74. Republic of South Africa: South African Institute for Computer Scientists and Information Technologists.

[SLR9] Alfaro, I., "The impact of multiple team memberships in leader: member exchange relationship (LMX)," 2009. In *IWIC '09: Proceeding of the 2009 international workshop on Intercultural collaboration*, 195-198. New York, NY, USA: ACM.

[SLR10] Alho, K. and Sulonen, R., "Supporting virtual software projects on the Web," *Proc Workshop Enabling Technol Infrastruct Collab Enterp WET*, pp. 10–14, 1998.

[SLR11] Ali-Babar, M., "The application of knowledge-sharing workspace paradigm for software architecture processes," 2008. In *SHARK '08: Proceedings of the 3rd international workshop on Sharing and reusing architectural knowledge*, 45-48. New York, NY, USA: ACM.

[SLR12] Anderl, R., Völz, D., and Rollmann, T., "Knowledge Integration in Global Engineering," *Enterprise Interoperability III*, pp. 471–482, 2008.

[SLR13] Andrade Paiva E, "The test community of practice experience in Brazil," *2006 1st IEEE International Conference on Global Software Engineering*, pp. pp, IEEE Computer Society.

[SLR14] "Applying Toyota production system to process improvement activities for outsourcing services," *Fujitsu, vol*, no. Fujitsu Ltd., pp. vol, 59, no.

[SLR15] Aranda, G., Vizcaíno, A., Cechich, A., and Piattini, M., "CHOOSING GROUPWARE TOOLS AND ELICITATION TECHNIQUES ACCORDING TO STAKEHOLDERS' FEATURES," *Enterprise Information Systems VII*, pp. 69–76, 2006.

[SLR16] Aranda, G., Vizcaíno, A., Cechich, A., and Piattini, M., "How to Choose Groupware Tools Considering Stakeholders' Preferences During Requirements Elicitation?" *Groupware: Design, Implementation, and Use*, pp. 319–327, 2007.

[SLR17] Aranda, G. N., Vizcaíno, A., Cechich, A., and Piattini, M., "A Cognitive Perspective for Choosing Groupware Tools and Elicitation Techniques in Virtual Teams," *Computational Science and Its Applications – ICCSA 2005*, pp. 1064–1074, 2005.

[SLR18] Arantes, A. O., Vijaykumar, N. L., Santiago Junior, V. A. de, and Guimaraes, D., "WEB-PerformCharts: a collaborative web-based tool for test case generation from statecharts," 2008. In *iiWAS '08: Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services*, 374-381. New York, NY, USA: ACM.

[SLR19] Ashmore C, "The outsourcing solution to inter-networking," *Engineering Management Journal, vol*, no. IEE, pp. vol, 5, no.

[SLR20] Au, P. K. and Ma, S., "Lessons learned from development and implementation of a strategic life cycle model: Reliability and Maintainability, 2004 Annual Symposium - RAMS," *Reliability and Maintainability, 2004 Annual Symposium - RAMS*, pp. 647–653, 2004.

[SLR21] Babar, M. A. and Niazi, M., "Implementing Software Process Improvement Initiatives: An Analysis of Vietnamese Practitioners' Views: Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on," *Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on*, pp. 67–76, 2008.

[SLR22] Babar MA and Niazi M, "Implementing software process improvement initiatives: an analysis of Vietnamese practitioners' views," *2008 IEEE International Conference on Global Software Engineering (ICGSE)*, pp. NJ USA, IEEE.

[SLR23] Babar MA, Verner JM, and Nguyen Phong Thanh, "Establishing and maintaining trust in software outsourcing relationships: an empirical investigation," *The Journal of Systems & Software, vol*, no. Elsevier Science Publishing Co. Inc., pp. vol, 80, no.

[SLR24] Bachmann, A., "Melting pot or tossed salad? Implications for designing effective multicultural workgroups," *Management International Review*, vol. 46, no. 6, pp. 721–748, Dec. 2006.

[SLR25] Baker, P., Loh, S., and Weil, F., "Model-Driven Engineering in a Large Industrial Context — Motorola Case Study," *Model Driven Engineering Languages and Systems*, pp. 476–491, 2005.

[SLR26] Bal J and Foster P, "Managing the virtual team and controlling effectiveness," *International Journal of Production Research, vol*, no. Taylor & Francis, pp. vol, 38, no.

[SLR27] Balthazard, P., Potter, R. E., and Warren, J., "Expertise, extraversion and group interaction styles as performance indicators in virtual teams: how do perceptions of IT's performance get formed?" *SIGMIS Database*, vol. 35, no. 1, pp. 41-64, 2004.

[SLR28] Balthazard, P. A., Potter, R. E., and Warren, J., "The effects of extraversion and expertise on virtual team interaction and performance: System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Con-

ference on," *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, pp. 10 pp., 2002.

[SLR29] Balthazard P, Potter RE, and Warren J, "Expertise, extraversion and group interaction styles as performance indicators in virtual teams," *Data Base for Advances in Information Systems, vol*, no. ACM, pp. vol, 35, no.

[SLR30] Balthazard PA, Potter RE, and Warren J, "The effects of extraversion and expertise on virtual team interaction and performance," *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, pp. pp, IEEE Comput.

[SLR31] Bandinelli R and Orsoni A, "Simulation-based DSS for the systematic assessment of outsourcing strategies," *International Journal of Simulation: Systems, Science & Technology, vol*, no. Systems, pp. Science & Technology vol, 6, no.

[SLR32] Bansler, J. P. and Havn, E., "Sensemaking in Technology-Use Mediation: Adapting Groupware Technology in Organizations," *Computer Supported Cooperative Work (CSCW)*, vol. 15, no. 1, pp. 55–91, Feb. 2006.

[SLR33] Bardhan, I., "Toward a theory to study the use of collaborative product commerce for product development," *Information Technology and Management*, vol. 8, no. 2, pp. 167–184, Jun. 2007.

[SLR34] Bartholomew, R. A. and Pearson, A. W., "Uncertainty and distance as determinant factors in portfolio management and strategic alliances: Management of Engineering and Technology, 1999. Technology and Innovation Management. PICMET '99. Portland International Conference on," *Management of Engineering and Technology, 1999. Technology and Innovation Management. PICMET '99. Portland International Conference on*, vol. 1, pp. 158 vol.1, 1999.

[SLR35] Beech B, "From important to vital: the evolution of a metrics program from internal to outsourced applications," *Cutter IT Journal, vol*, no. Cutter Inf. Corp, pp. vol, 16, no.

[SLR36] Begel, A., "Effecting change: coordination in large-scale software development," 2008. In *CHASE '08: Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering*, 17-20. New York, NY, USA: ACM.

[SLR37] Bégin, M.-E., Sancho, G., Di Meglio, A., Ferro, E., Ronchieri, E., Selmi, M., and Żurek, M., "Build, Configuration, Integration and Testing Tools for Large

Software Projects: ETICS," *Rapid Integration of Software Engineering Techniques*, pp. 81–97, 2007.

[SLR38] Beise, C. M., "IT project management and virtual teams," 2004. In *SIGMIS CPR '04: Proceedings of the 2004 SIGMIS conference on Computer personnel research*, 129-133. New York, NY, USA: ACM.

[SLR39] Belov, N., Koeck, C., Krandick, W., Shaffer, J., and Hewett, T., "Intellectual teamwork on smartphones," 2005. In *MobileHCI '05: Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*, 379-380. New York, NY, USA: ACM.

[SLR40] Bennett M, "Tips for better outsourcing," *IT Week, vol*, no. VNU Business Publications, pp. vol, 7, no.

[SLR41] Berner, S., Weber, R., and Keller, R. K., "Observations and lessons learned from automated testing," 2005. In *ICSE '05: Proceedings of the 27th international conference on Software engineering*, 571-579. New York, NY, USA: ACM.

[SLR42] Bernuy, A. and Joyanes, L., "Collaborative e-Business and Software Agents," *Innovations and Advanced Techniques in Systems, Computing Sciences and Software Engineering*, pp. 121–126, 2008.

[SLR43] Bertch W, "Why offshore outsourcing failed us," *Network Computing, vol*, no. CMP Media Inc, pp. vol, 14, no.

[SLR44] Beyah, G. and Gallivan, M., "Knowledge management as a framework for understanding public sector outsourcing: System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on," *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, pp. 9 pp., 2001.

[SLR45] Biro M and Feher P, "Forces affecting offshore software development," *Software Process Improvement*, pp. EuroSPI 2005, 12th European Conference, EuroSPI 2005.

[SLR46] Blecher M, "Outsourcing IT governance to deliver business value," *Information Systems Control Journal, vol*, no. Information Systems Audit and Control Association, pp. vol, 4, 2007, pp.

[SLR47] Boden, A., Nett, B., and Wulf, V., "Articulation work in small-scale offshore software development projects," 2008. In *CHASE '08: Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering*, 21-24. New York, NY, USA: ACM.

[SLR48] Boden, A., Nett, B., and Wulf, V., "Coordination Practices in Distributed Software Development of Small Enterprises: Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on," *Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on*, pp. 235–246, 2007.

[SLR49] Boehm, B., "Value-Based Software Engineering: Seven Key Elements and Ethical Considerations," *Value-Based Software Engineering*, pp. 109–132, 2006.

[SLR50] Bongio, A., van Bruggen, J., Ceri, S., Cristea, V., Dolog, P., Hoffmann, A., Matera, M. et al., "COOPER: Towards a Collaborative Open Environment of Project-Centred Learning," *Innovative Approaches for Learning and Knowledge Sharing*, pp. 561–566, 2006.

[SLR51] Borgman, C., "What can Studies of e-Learning Teach us about Collaboration in e-Research? Some Findings from Digital Library Studies," *Computer Supported Cooperative Work (CSCW)*, vol. 15, no. 4, pp. 359–383, Aug. 2006.

[SLR52] Bos, N., Shami, N. S., Olson, J. S., Cheshin, A., and Nan, N., "In-group/outgroup effects in distributed teams: an experimental simulation," 2004. In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, 429-436. New York, NY, USA: ACM.

[SLR53] Boyer, D. G., Handel, M. J., and Herbsleb, J., "Virtual community prescence awareness," *SIGGROUP Bull.*, vol. 19, no. 3, pp. 11-14, 1998.

[SLR54] Braithwaite, K. and Joyce, T., "XP Expanded: Distributed Extreme Programming," *Extreme Programming and Agile Processes in Software Engineering*, pp. 180–188, 2005.

[SLR55] Brechner, E., "Journey of enlightenment: the evolution of development at Microsoft," 2005. In *ICSE '05: Proceedings of the 27th international conference on Software engineering*, 39-42. New York, NY, USA: ACM.

[SLR56] Buck Lew M, "To outsource or not?" *International Journal of Information Management, vol*, pp. vol, 12, no.

[SLR57] Burden K, "'We're in it to win it"- negotiating successful outsourcing transactions," *Computer Law and Security Report, vol*, no. Elsevier, pp. vol, 19, no.

[SLR58] Cao Ying, Sun Fuquan, Wang Lina, and Chen Tingbin, "Promotion of Offshore Software Outsourcing Enterprise Value Chain Based on Knowledge Transfer Effects: Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on," *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, pp. 1–4, 2008.

[SLR59] Carmel E and Nicholson B, "Small firms and offshore software outsourcing: high transaction costs and their mitigation," *Journal of Global Information Management, vol*, no. Idea Group Publishing, pp. vol, 13, no.

[SLR60] Carte, T., Chidambaram, L., Garfield, M., Hicks, L., and Cole, C., "Group Creativity and Collaborative Technologies: Understanding the Role of Visual Anonymity," *Groupware: Design, Implementation, and Use*, pp. 12–21, 2006.

[SLR61] Carter, C. J., "Rapid integration and testing of business solutions," *BT Technology Journal*, vol. 15, no. 3, pp. 37–47, Jul. 1997.

[SLR62] Carver, J. C., Kendall, R. P., Squires, S. E., and Post, D. E., "Software Development Environments for Scientific and Engineering Software: A Series of Case Studies," 2007. In *ICSE '07: Proceedings of the 29th international conference on Software Engineering*, 550-559. Washington, DC, USA: IEEE Computer Society.

[SLR63] Casati, F., Castellanos, M., Salazar, N., and Dayal, U., "Abstract Process Data Warehousing: Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on," *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pp. 1387–1389, 2007.

[SLR64] Casati F, Castellanos M, Salazar N, and Dayal U, "Abstract process data warehousing," *2007 IEEE 23rd International Conference on Data Enginering (IEEE Cat*, pp. NJ USA, No.

[SLR65] Casey, V. and Richardson, I., "Uncovering the reality within virtual software teams," 2006. In *GSD '06: Proceedings of the 2006 international workshop on Global software development for the practitioner*, 66-72. New York, NY, USA: ACM.

[SLR66] Casey, V. and Richardson, I., "Project management within virtual software teams," *Proc. IEEE Int. Conf. Global Softw. Eng., ICGSE*, pp. 33–42, 2006.

[SLR67] Cataldo, M. and Herbsleb, J. D., "Communication networks in geographically distributed software development," 2008. In *CSCW '08: Proceedings of the ACM 2008 conference on Computer supported cooperative work*, 579-588. New York, NY, USA: ACM.

[SLR68] Cater Steel A and Toleman M, "The impact of national culture on software engineering practices," *International Journal of Technology, Policy and Management, vol*, no. Inderscience Enterprises Ltd., pp. Policy and Management vol, 8, no.

[SLR69] Cater-Steel, A. and Toleman, M., "The impact of national culture on software engineering practices," *Int. J. Technol. Policy Manage.*, vol. 8, no. 1, pp. 76–90, 2008.

[SLR70] Chang, K. T. and Ehrlich, K., "Out of sight but not out of mind? Informal networks, communication and media use in global software teams," 2007. In *CASCON '07: Proceedings of the 2007 conference of the center for advanced studies on Collaborative research*, 86-97. New York, NY, USA: ACM.

[SLR71] Cheng, B. H. C. and Atlee, J. M., "Current and Future Research Directions in Requirements Engineering," *Design Requirements Engineering: A Ten-Year Perspective*, pp. 11–43, 2009.

[SLR72] Chestnutt A, "The ins and outs of outsourcing," *Application Development Advisor, vol*, no. SIGS, pp. vol, 1, no.

[SLR73] Chira, O., Chira, C., Roche, T., Tormey, D., and Brennan, A., "An agent-based approach to knowledge management in distributed design," *Journal of Intelligent Manufacturing*, vol. 17, no. 6, pp. 737–750, Dec. 2006.

[SLR74] Chituc, C.-M. and Azevedo, A., "Multi-Perspective Challenges on Collaborative Networks Business Environments," *Collaborative Networks and Their Breeding Environments*, pp. 25–32, 2005.

[SLR75] Choi, S.-M., Yoo, C.-J., and Chang, O.-B., "Development of Integrated DAO Pattern Applying Iterator Pattern," *Computational Science and Its Applications - ICCSA 2006*, pp. 955–965, 2006.

[SLR76] Choy KL, Lee WB, Lau HCW, Dawei Lu, and Lo V, "Design of an intelligent supplier relationship management system for new product development,"

International Journal of Computer Integrated Manufacturing, vol, no. Taylor & Francis, pp. vol, 17, no.

[SLR77] Choy KL, Lee WB, and Lo V, "Design of an intelligent supplier relationship management system: a hybrid case based neural network approach," *Expert Systems with Applications, vol*, no. Elsevier, pp. vol, 24, no.

[SLR78] Cicchetti, A., Di Ruscio, D., and Pierantonio, A., "Managing Model Conflicts in Distributed Development," *Model Driven Engineering Languages and Systems*, pp. 311–325, 2008.

[SLR79] Cichocki, P. and Maccari, A., "Empirical Analysis of a Distributed Software Development Project," *Balancing Agility and Formalism in Software Engineering*, pp. 169–181, 2008.

[SLR80] Clear, T. and Kassabova, D., "Motivational patterns in virtual team collaboration," 2005. In *ACE '05: Proceedings of the 7th Australasian conference on Computing education*, 51-58. Darlinghurst, Australia, Australia: Australian Computer Society, Inc.

[SLR81] Clerc, V., "Towards architectural knowledge management practices for global software development," 2008. In *SHARK '08: Proceedings of the 3rd international workshop on Sharing and reusing architectural knowledge*, 23-28. New York, NY, USA: ACM.

[SLR82] Coleman, G. and O'Connor, R., "Software Process in Practice: A Grounded Theory of the Irish Software Industry," *Software Process Improvement*, pp. 28–39, 2006.

[SLR83] Coman A and Ronen B, "Production outsourcing: a linear programming model for the theory-of-constraints," *International Journal of Production Research, vol*, no. Taylor & Francis, pp. vol, 38, no.

[SLR84] "Controlling for IT and Software," *Software Measurement*, pp. 435–470, 2007.

[SLR85] Cordes, R. E. and Spine, T. M., "Transcending organizational boundaries: virtual team approach in UI guideline development," 2007. In *CHIMIT '07: Proceedings of the 2007 symposium on Computer human interaction for the management of information technology*, 7. New York, NY, USA: ACM.

[SLR86] Cristal, M. and Reis, J., "Leveraging lessons learned for distributed projects through communities of practice," *Proc. IEEE Int. Conf. Global Softw. Eng., ICGSE*, pp. 239–240, 2006.

[SLR87] Cristal M and Reis J, "Leveraging lessons learned for distributed projects through communities of practice," *2006 1st IEEE International Conference on Global Software Engineering*, pp. pp, IEEE Computer Society.

[SLR88] "Customer Challenges and Demands," *Active Knowledge Modeling of Enterprises*, pp. 27–64, 2008.

[SLR89] Cusumano, M. A., "Managing software development in globally distributed teams," *Commun. ACM*, vol. 51, no. 2, pp. 15-17, 2008.

[SLR90] D'Eredita, M. and Nilan, M., "Conceptualizing Virtual Collaborative Work," *Virtuality and Virtualization*, pp. 21–34, 2007.

[SLR91] Damian, D., Hadwin, A., and Al-Ani, B., "Instructional design and assessment strategies for teaching global software development: a framework," 2006. In *ICSE '06: Proceedings of the 28th international conference on Software engineering*, 685-690. New York, NY, USA: ACM.

[SLR92] Damian, D., Lanubile, F., and Mallardo, T., "The role of asynchronous discussions in increasing the effectiveness of remote synchronous requirements negotiations," 2006. In *ICSE '06: Proceedings of the 28th international conference on Software engineering*, 917-920. New York, NY, USA: ACM.

[SLR93] Damian, D., "An empirical study of requirements engineering in distributed software projects: is distance negotiation more effective? Software Engineering Conference, 2001. APSEC 2001. Eighth Asia-Pacific," *Software Engineering Conference, 2001. APSEC 2001. Eighth Asia-Pacific*, pp. 149–152, 2001.

[SLR94] Damian D, "An empirical study of requirements engineering in distributed software projects: is distance negotiation more effective?" *Proceedings Eighth Asia Pacific Software Engineering Conference*, pp. pp, IEEE Comput.

[SLR95] Damian D, "Stakeholders in global requirements engineering: Lessons learned from practice," *IEEE Software, vol*, no. IEEE, pp. vol, 24, no.

[SLR96] de Jager W, "The management paradox [outsourcing]," *Informatie, vol*, no. Kluwer Bedrijfswetenschappen, pp. vol, 40, spec.

[SLR97] Dekel, U., "Supporting distributed software design meetings: what can we learn from co-located meetings?" 2005. In *HSSE '05: Proceedings of the 2005 workshop on Human and social factors of software engineering*, 1-7. New York, NY, USA: ACM.

[SLR98] Dekel, U., "Towards distributed software design meetings: what can we learn from co-located meetings?" 2005. In *OOPSLA '05: Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, 226-227. New York, NY, USA: ACM.

[SLR99] DeLuca D and Valacich JS, "Virtual teams in and out of synchronicity," *Information Technology & People, vol*, no. Emerald, pp. vol, 19, no.

[SLR100] Dewan, P., "Dimensions of tools for detecting software conflicts," 2008. In *RSSE '08: Proceedings of the 2008 international workshop on Recommendation systems for software engineering*, 21-25. New York, NY, USA: ACM.

[SLR101] Domis, D. and Trapp, M., "Integrating Safety Analyses and Component-Based Design," *Computer Safety, Reliability, and Security*, pp. 58–71, 2008.

[SLR102] Donker, H. and Blumberg, M., "Collaborative process management and virtual teams," 2008. In *CHASE '08: Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering*, 41-43. New York, NY, USA: ACM.

[SLR103] Draghici, A., "Influence of Multi-Culturality in Virtual Teams," *Methods and Tools for Effective Knowledge Life-Cycle-Management*, pp. 61–89, 2008.

[SLR104] Draghici, A. and Draghici, G., "Building a Knowledge Share Culture in a Virtual Organization. Case Study for VRL-KCiP NoE," *Methods and Tools for Effective Knowledge Life-Cycle-Management*, pp. 45–60, 2008.

[SLR105] Drummer M, "Lower-cost collaborations [document conferencing]," *Communications News, vol*, no. Nelson Publishing, pp. vol, 34, no.

[SLR106] Dustdar, S. and Gall, H., "Architectural concerns in distributed and mobile collaborative systems," *J Syst Archit*, vol. 49, no. 10-11, pp. 457–473, 2003.

[SLR107] Dustdar, S., "Caramba—A Process-Aware Collaboration System Supporting Ad hoc and Collaborative Processes in Virtual Teams," *Distributed and Parallel Databases*, vol. 15, no. 1, pp. 45–66, Jan. 2004.

[SLR108] Ebert, C., "Optimizing Supplier Management in Global Software Engineering: Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on," *Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on*, pp. 177–185, 2007.

[SLR109] Ebert C, "Optimizing supplier management in global software engineering," *2007 2nd IEEE International Conference on Global Software Engineering*, pp. NJ USA, IEEE.

[SLR110] Edgar, D. R., "Lessons Learned: Outsourcing Support Equipment Development: Autotestcon, 2006 IEEE," *Autotestcon, 2006 IEEE*, pp. 80–85, 2006.

[SLR111] Edwards L, "When outsourcing is appropriate," *Wall Street & Technology, vol*, no. Miller Freeman, pp. vol, 16, no.

[SLR112] Egea, K., "Managing the managers: collaborative virtual teams with large staff and student numbers," 2003. In *ACE '03: Proceedings of the fifth Australasian conference on Computing education*, 87-94. Darlinghurst, Australia, Australia: Australian Computer Society, Inc.

[SLR113] Ellis, J. B., Luther, K., Bessiere, K., and Kellogg, W. A., "Games for virtual team building," 2008. In *DIS '08: Proceedings of the 7th ACM conference on Designing interactive systems*, 295-304. New York, NY, USA: ACM.

[SLR114] Ellis, J. B., Wahid, S., Danis, C., and Kellogg, W. A., "Task and social visualization in software development: evaluation of a prototype," 2007. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, 577-586. New York, NY, USA: ACM.

[SLR115] "Empirical Laws and Rules of Thumb," *Software Measurement*, pp. 487–497, 2007.

[SLR116] Estublier, J. and Garcia, S., "Process model and awareness in SCM," 2005. In *SCM '05: Proceedings of the 12th international workshop on Software configuration management*, 59-74. New York, NY, USA: ACM.

[SLR117] Evaristo, R., Nicolas, J. L., Prikladnicki, R., and Avritchir, J., "Wholly Owned Offshore Subsidiaries for IT Development: A Program of Research: System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on," *System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on*, pp. 258b-258b, 2005.

[SLR118] Fang He and Paul, S., "Time Pressure and Reward Inspiration as Outcome Controls for Information Sharing in Problem-Solving Virtual Teams: System

Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on," *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, pp. 39-39, 2007.

[SLR119] Farenhorst, R., Lago, P., and van Vliet, H., "Effective Tool Support for Architectural Knowledge Sharing," *Software Architecture*, pp. 123–138, 2007.

[SLR120] Fernandez A, Garzaldeen B, Grutzner I, and Munch J, "Guided support for collaborative modeling, enactment and simulation of software development processes," *Software Process Improvement and Practice, vol*, no. Wiley, pp. vol, 9, no.

[SLR121] Ferscha, A. and Scheiner, C., "Collective choice in virtual teams: Enabling Technologies: Infrastructure for Collaborative Enterprises, 1999. (WET ICE '99) Proceedings. IEEE 8th International Workshops on," *Enabling Technologies: Infrastructure for Collaborative Enterprises, 1999. (WET ICE '99) Proceedings. IEEE 8th International Workshops on*, pp. 96–101, 1999.

[SLR122] Fitzpatrick, G., Marshall, P., and Phillips, A., "CVS integration with notification and chat: lightweight software team collaboration," 2006. In *CSCW '06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, 49-58. New York, NY, USA: ACM.

[SLR123] Fricker, S., Gorschek, T., and Myllyperkiö, P., "Handshaking Between Software Projects and Stakeholders Using Implementation Proposals," *Requirements Engineering: Foundation for Software Quality*, pp. 144–159, 2007.

[SLR124] Fricker, S. and Grünbacher, P., "Negotiation Constellations – Method Selection Framework for Requirements Negotiation," *Requirements Engineering: Foundation for Software Quality*, pp. 37–51, 2008.

[SLR125] Fricker S, Gorschek T, and Myllyperkio P, "Handshaking between software projects and stakeholders using implementation proposals," *Requirements Engineering: Foundation for Software Quality*, no. Foundation for Software Quality. Proceedings 13th International Working Conference, pp. REFSQ 2007, Proceedings 13th International Working Conference, REFSQ 2007.

[SLR126] García, A., Hardt, M., and Kornmayer, H., "Tools for Distributed Development and Deployment on the Grid," *Advances in Grid Computing - EGC 2005*, pp. 831–839, 2005.

[SLR127] Geisser, M. and Hildenbrand, T., "A Method for Collaborative Requirements Elicitation and Decision-Supported Requirements Analysis," *Advanced Software Engineering: Expanding the Frontiers of Software Technology*, pp. 108–122, 2006.

[SLR128] George B and Mansour M, "A multidisciplinary virtual team," *6th World Multiconference on Systemics, Cybernetics and Informatics*, pp. Cybernetics and Informatics, Proceedings.

[SLR129] Gibbs, J. L., "Culture as kaleidoscope: navigating cultural tensions in global collaboration," 2009. In *IWIC '09: Proceeding of the 2009 international workshop on Intercultural collaboration*, 89-98. New York, NY, USA: ACM.

[SLR130] Gilbert, E. and Karahalios, K., "CodeSaw: A Social Visualization of Distributed Software Development," *Human-Computer Interaction – INTERACT 2007*, pp. 303–316, 2008.

[SLR131] Godart, C., Bouthier, C., Canalda, P., Charoy, F., Molli, P., Perrin, O., Saliou, H., Bignon, J. C., Halin, G., and Malcurat, O., "Asynchronous coordination of virtual teams in creative applications (co-design or co-engineering): requirements and design criteria," 2001. In *ITVE '01: Proceedings of the workshop on Information technology for virtual enterprises*, 135-142. Washington, DC, USA: IEEE Computer Society.

[SLR132] Godart, C., Molli, P., Oster, G., Perrin, O., Skaf-Molli, H., Ray, P., and Rabhi, F., "The ToxicFarm Integrated Cooperation Framework for Virtual Teams," *Distributed and Parallel Databases*, vol. 15, no. 1, pp. 67–88, Jan. 2004.

[SLR133] Gorschek, T., Fricker, S., Felt, R., Wohlin, C., and Mattsson, M., "1st International Global Requirements Engineering Workshop -- GREW'07," *SIGSOFT Softw. Eng. Notes*, vol. 33, no. 2, pp. 29-32, 2008.

[SLR134] Gorton, I., Hawryszkiewycz, I., and Ragoonaden, K., "Collaborative tools and processes to support software engineering shift work," *BT Technology Journal*, vol. 15, no. 3, pp. 189–198, Jul. 1997.

[SLR135] Gotel, O., Kulkarni, V., Des Phal, Say, M., Scharff, C., and Sunetnanta, T., " Evolving an infrastructure for student global software development projects: lessons for industry," 2009. In *ISEC '09: Proceeding of the 2nd annual conference on India software engineering conference*, 117-126. New York, NY, USA: ACM.

[SLR136] Gotel, O., Kulkarni, V., Neak, L., Scharff, C., and Seng, S., "Introducing Global Supply Chains into Software Engineering Education," *Software Engineering Approaches for Offshore and Outsourced Development*, pp. 44–58, 2007.

[SLR137] Griffin, T. M., "Effective Intercultural Communication," *Effective Multicultural Teams: Theory and Practice*, pp. 173–210, 2008.

[SLR138] Grinter, R. E., Herbsleb, J. D., and Perry, D. E., "The geography of coordination: dealing with distance in R&D work," 1999. In *GROUP '99: Proceedings of the international ACM SIGGROUP conference on Supporting group work*, 306-315. New York, NY, USA: ACM.

[SLR139] Guowei Zu, Taira, H., Makino, K., Kano, T., and Matsumoto, S., "The Supporting Technology of Business Document Proofreading based on Intercultural Differences: E-Commerce Technology and the 4th IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services, 2007. CEC/EEE 2007. The 9th IEEE International Conference on," *E-Commerce Technology and the 4th IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services, 2007. CEC/EEE 2007. The 9th IEEE International Conference on*, pp. 91–98, 2007.

[SLR140] Gurbani, V. K., Garvert, A., and Herbsleb, J. D., "A case study of a corporate open source development model," 2006. In *ICSE '06: Proceedings of the 28th international conference on Software engineering*, 472-481. New York, NY, USA: ACM.

[SLR141] Gutwin, C., Penner, R., and Schneider, K., "Group awareness in distributed software development," 2004. In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, 72-81. New York, NY, USA: ACM.

[SLR142] Gutwin, C., Schneider, K., Paquette, D., and Penner, R., "Supporting Group Awareness in Distributed Software Development," *Engineering Human Computer Interaction and Interactive Systems*, pp. 383–397, 2005.

[SLR143] Haag, Z., Foley, R., and Newman, J., "Software process improvement in geographically distributed software engineering: an initial evaluation: EUROMICRO 97. 'New Frontiers of Information Technology'., Proceedings of the 23rd EUROMICRO Conference," *EUROMICRO 97. 'New Frontiers of Information Technology'., Proceedings of the 23rd EUROMICRO Conference*, pp. 134–141, 1997.

[SLR144] Haag, Z., Foley, R., and Newman, J., "Deontic formalism for co-ordinating software development in virtual software corporations," *Proc Workshop Enabling Technol Infrastruct Collab Enterp WET*, pp. 15–20, 1998.

[SLR145] Haag Z, Foley R, and Newman J, "Software process improvement in geographically distributed software engineering: an initial evaluation," *Proceedings*, no. New Frontiers of Information Technology (Cat. No.97TB100167). IEEE Comput. Soc. 1997, pp. pp, 23rd Euromicro Conference: New Frontiers of Information Technology (Cat.

[SLR146] Hackett M, "Building effective global software test teams through training," *2007 2nd IEEE International Conference on Global Software Engineering*, pp. NJ USA, IEEE.

[SLR147] Haddad M, La Salle AJ, and Harrald JR, "Software subcontracting and the cost to the mature organization," *Proceedings of the Fifth Americas Conference on Information Systems (AMCIS 1999)*, pp. pp, Assoc.

[SLR148] Hammes, D., Schumann, J. H., Steinbach, A., and Wangenheim, F., " Interkulturelle Problemfelder beim Offshoring von IT-Dienstleistungen nach China und Indien sowie Lösungsansätze aus der Unternehmenspraxis," *Remote Services*, pp. 67–88, 2007.

[SLR149] Hammock N, "Using the CMM for contract requirements and managing outsourcing," *Cutter IT Journal, vol*, no. Cutter Inf. Corp, pp. vol, 12, no.

[SLR150] Hargreaves, E., Damian, D., Lanubile, F., and Chisan, J., "Global software development: building a research community," *SIGSOFT Softw. Eng. Notes*, vol. 29, no. 5, pp. 1-5, 2004.

[SLR151] Hawlitzky N, Munnich A, Nothhelfer Kolb B, and Rappl M, "Managing distributed software development-a portfolio based methodology," *Product Focused Software Process Improvement*, pp. PROFES 2000, Second International Conference, PROFES 2000.

[SLR152] Hawthorne, M. J. and Perry, D. E., "Software engineering education in the era of outsourcing, distributed development, and open source software: challenges and opportunities," 2005. In *ICSE '05: Proceedings of the 27th international conference on Software engineering*, 643-644. New York, NY, USA: ACM.

[SLR153] Hawthorne, M. and Perry, D., "Software Engineering Education in the Era of Outsourcing, Distributed Development, and Open Source Software: Chal-

lenges and Opportunities," *Software Engineering Education in the Modern Age*, pp. 166–185, 2006.

[SLR154] Hayes IS, "Ready or not: global sourcing is in your IT future," *Cutter IT Journal, vol*, no. Cutter Inf. Corp, pp. vol, 15, no.

[SLR155] Hazzan, O. and Dubinsky, Y., "Globalization," *Agile Software Engineering*, pp. 1–15, 2008.

[SLR156] He, M., Li, M., Wang, Q., Yang, Y., and Ye, K., "An Investigation of Software Development Productivity in China," *Making Globally Distributed Software Development a Success Story*, pp. 381–394, 2008.

[SLR157] Heckman, R., Crowston, K., and Misiolek, N., "A Structurational Perspective on Leadership in Virtual Teams," *Virtuality and Virtualization*, pp. 151–168, 2007.

[SLR158] Heindl, M. and Biffl, S., "Risk management with enhanced tracing of requirements rationale in highly distributed projects," 2006. In *GSD '06: Proceedings of the 2006 international workshop on Global software development for the practitioner*, 20-26. New York, NY, USA: ACM.

[SLR159] Helander, M. E., Ming Zhao, and Ohlsson, N., "Planning models for software reliability and cost: Software Engineering, IEEE Transactions on," *Software Engineering, IEEE Transactions on*, vol. 24, no. 6, pp. 420–434, 1998.

[SLR160] Herbsleb, J. D., "Global Software Engineering: The Future of Socio-technical Coordination," 2007. In *FOSE '07: 2007 Future of Software Engineering*, 188-198. Washington, DC, USA: IEEE Computer Society.

[SLR161] Herbsleb, J. D., Paulish, D. J., and Bass, M., "Global software development at siemens: experience from nine projects," 2005. In *ICSE '05: Proceedings of the 27th international conference on Software engineering*, 524-533. New York, NY, USA: ACM.

[SLR162] Herbsleb, J. D., Paulish, D. J., and Bass, M., "Global software development at Siemens: experience from nine projects: Software Engineering, 2005. ICSE 2005. Proceedings. 27th International Conference on," *Software Engineering, 2005. ICSE 2005. Proceedings. 27th International Conference on*, pp. 524–533, 2005.

[SLR163] Herbsleb JD, Paulish DJ, and Bass M, "Global software development at Siemens: experience from nine projects," *27th International Conference on Software Engineering (IEEE Cat*, pp. pp, No.

[SLR164] Hogan, B., "Lessons learned from an extremely distributed project: Agile Conference, 2006," *Agile Conference, 2006*, pp. 6 pp.-326, 2006.

[SLR165] Hogan B, "Lessons learned from an extremely distributed project," *AGILE 2006*, pp. pp, IEEE Comput.

[SLR166] Hole, S. and Moe, N. B., "A Case Study of Coordination in Distributed Agile Software Development," *Software Process Improvement*, pp. 189–200, 2008.

[SLR167] Huang, H. and Ocker, R., "Preliminary insights into the in-group/out-group effect in partially distributed teams: an analysis of participant reflections," 2006. In *SIGMIS CPR '06: Proceedings of the 2006 ACM SIGMIS CPR conference on computer personnel research*, 264-272. New York, NY, USA: ACM.

[SLR168] Huang SH and Keskar H, "Comprehensive and configurable metrics for supplier selection," *International Journal of Production Economics, vol*, no. Elsevier, pp. vol, 105, no.

[SLR169] Huysman, M., Steinfield, C., Jang, C.-Y., David, K., Veld, M. H. in 't, Poot, J., and Mulder, I., "Virtual Teams and the Appropriation of Communication Technology: Exploring the Concept of Media Stickiness," *Computer Supported Cooperative Work (CSCW)*, vol. 12, no. 4, pp. 411–436, Dec. 2003.

[SLR170] Hwang, J., Kim, N., and Kim, G., "Immersing Tele-operators in Collaborative Augmented Reality," *Advances in Visual Computing*, pp. 692–701, 2006.

[SLR171] Ignat, C.-L., Papadopoulou, S., Oster, G., and Norrie, M. C., "Providing awareness in multi-synchronous collaboration without compromising privacy," 2008. In *CSCW '08: Proceedings of the ACM 2008 conference on Computer supported cooperative work*, 659-668. New York, NY, USA: ACM.

[SLR172] "Information & Communication Technologies," *Managing Global Innovation*, pp. 233–251, 2008.

[SLR173] Iwami, Y., "How to Measure Quality of Software Developed by Subcontractors (Short Paper): Quality Software, 2008. QSIC '08. The Eighth Interna-

tional Conference on," *Quality Software, 2008. QSIC '08. The Eighth International Conference on*, pp. 300–304, 2008.

[SLR174] Iwami Y, "How to measure quality of software developed by subcontractors," *Eighth International Conference on Quality Software*, pp. NJ USA, QSIC 2008.

[SLR175] Jain N, "Offshore agile maintenance," *AGILE 2006*, pp. pp, IEEE Comput.

[SLR176] Jamieson D, Vinsen K, and Callender PG, "Measuring software costs: a new perspective on a recurring problem," *"Sixth International Workshop on Economics Driven Software Engineering Research (EDSER 6)" W9L Workshop 26th International Conference on Software Engineering*, pp. pp, IEE.

[SLR177] Jiyang Xu and Lippert, D., "Lesson Learned in Managing IT Departments: Management of Engineering and Technology, Portland International Center for," *Management of Engineering and Technology, Portland International Center for*, pp. 2107–2115, 2007.

[SLR178] Junchao Xiao and Qing Wang, "Contract-driven cooperation development process for software outsourcing: Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference on," *Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference on*, pp. 1104–1108, 2005.

[SLR179] Kalibera, T., Bulej, L., and Tuma, P., "Quality Assurance in Performance: Evaluating Mono Benchmark Results," *Quality of Software Architectures and Software Quality*, pp. 271–288, 2005.

[SLR180] Kapur, P. K., Jha, P. C., and Singh, V. B., "On the Development of Discrete Software Reliability Growth Models," *Handbook of Performability Engineering*, pp. 1239–1255, 2008.

[SLR181] Karayaz, G. and Keating, C. B., "Virtual Team Effectiveness Using Dyadic Teams: Management of Engineering and Technology, Portland International Center for," *Management of Engineering and Technology, Portland International Center for*, pp. 2593–2603, 2007.

[SLR182] Karayaz G and Keating CB, "Virtual team effectiveness using dyadic teams," *PICMET '07*, pp. NJ USA, 2007 Portland International Conference on Management of Engineering and Technology.

[SLR183] Karduck, A. and Sienou, A., "Teambroker: Constraint Based Brokerage of Virtual Teams," *Enterprise Information Systems VI*, pp. 229–236, 2006.

[SLR184] Karlsson, E.-A., Andersson, L.-G., and Leion, P., "Daily build and feature development in large distributed projects," 2000. In *ICSE '00: Proceedings of the 22nd international conference on Software engineering*, 649-658. New York, NY, USA: ACM.

[SLR185] Kedian Mu, Weiru Liu, Zhi Jin, Ruqian Lu, Anbu Yue, and Bell D, "A merging-based approach to handling inconsistency in locally prioritized software requirements," *Knowledge Science, Engineering and Management*, pp. Engineering and Management, Proceedings Second International Conference, KSEM 2007.

[SLR186] Keil, P., Paulish, D. J., and Sangwan, R. S., "Cost estimation for global software development," 2006. In *EDSER '06: Proceedings of the 2006 international workshop on Economics driven software engineering research*, 7-10. New York, NY, USA: ACM.

[SLR187] Kirkman BL, Rosen B, Tesluk PE, and Gibson CB, "The impact of team empowerment on virtual team performance: the moderating role of face-to-face interaction," *Academy of Management Journal, vol*, no. Acad. of Management, pp. vol, 47, no.

[SLR188] Koh, C., Joseph, D., and Ang, S., "Cultural intelligence and collaborative work: intercultural competencies in global technology work teams," 2009. In *IWIC '09: Proceeding of the 2009 international workshop on Intercultural collaboration*, 261-264. New York, NY, USA: ACM.

[SLR189] Kommeren, R. and Parviainen, P., "Philips experiences in global distributed software development," *Empirical Software Engineering*, vol. 12, no. 6, pp. 647–660, Dec. 2007.

[SLR190] Kommeren R and Parviainen P, "Philips experiences in global distributed software development," *Empirical Software Engineering, vol*, no. Kluwer Academic Publishers Group, pp. vol, 12, no.

[SLR191] Kumar N and Palvia P, "A framework for global IT outsourcing management: key influence factors and strategies," *Journal of Information Technology Cases and Applications (JITCA), vol*, no. Ivy League Publishing, pp. vol, 4, no.

[SLR192] Kumar S, Aquino EC, and Anderson E, "Application of a process methodology and a strategic decision model for business process outsourcing," *Information*, no. IOS Press, pp. vol, Knowledge.

[SLR193] Kussmaul C, Jack R, and Sponsler B, "Outsourcing and offshoring with agility: a case study," *Extreme Programming and Agile Methods XP/Agile Universe 2004*, pp. pp, 4th Conference on Extreme Programming and Agile Methods.

[SLR194] Langenberg L and Kuldszun K, "The development of Communities of Practice: knowledge exchange within transnational organizations," *ZWF Zeitschrift fur Wirtschaftlichen Fabrikbetrieb, vol*, no. Carl Hanser Verlag, pp. vol, 100, no.

[SLR195] Langford, L. K., "Surf"s Up: The Project Management Process: Engineering Management Review, IEEE," *Engineering Management Review, IEEE*, vol. 36, no. 2, pp. 125–126, 2008.

[SLR196] Last, M. Z., Daniels, M., Hause, M. L., and Woodroffe, M. R., "Learning from students: continuous improvement in international collaboration," 2002. In *ITiCSE '02: Proceedings of the 7th annual conference on Innovation and technology in computer science education*, 136-140. New York, NY, USA: ACM.

[SLR197] Lee, L.-C. and Wei, W.-J., "Case Study of Breakdown Analysis on Identification of Remote Team Communication Problems," *Computer Supported Cooperative Work in Design II*, pp. 122–130, 2006.

[SLR198] Leinonen, P., Järvelä, S., and Häkkinen, P., "Conceptualizing the Awareness of Collaboration: A Qualitative Study of a Global Virtual Team," *Computer Supported Cooperative Work (CSCW)*, vol. 14, no. 4, pp. 301–322, Aug. 2005.

[SLR199] Leonardo Pilatti and Jorge Luis Nicolas Audy, "Global Software Development Offshore Insourcing Organizations Characteristics: Lessons Learned from a Case Study: Global Software Engineering, 2006. ICGSE '06. International Conference on," *Global Software Engineering, 2006. ICGSE '06. International Conference on*, pp. 249–250, 2006.

[SLR200] Levine L and Syzdek G, "Across the divide: two organisations form a virtual team and codevelop a product," *Diffusing Software Product and Process Innovations*, pp. pp, IFIP TC8 WG8.

[SLR201] Lewkowicz, M., Wijnhoven, F., and Draghici, A., "Misunderstandings in Global Virtual Engineering Teams: Definitions, Causes, and Guidelines for Knowledge Sharing and Interaction," *Methods and Tools for Effective Knowledge Life-Cycle-Management*, pp. 145–157, 2008.

[SLR202] Li, J., Li, M., Wu, Z., and Wang, Q., "A Metamodel for the CMM Software Process," *Parallel and Distributed Processing and Applications*, pp. 446–450, 2005.

[SLR203] Lim, S., "GSSD-Enterprise for Multinational Corporations," *Mapping Sustainability*, pp. 209–228, 2007.

[SLR204] Lindqvist, E., Lundell, B., and Lings, B., "Distributed development in an intra-national, intra-organisational context: an experience report," 2006. In *GSD '06: Proceedings of the 2006 international workshop on Global software development for the practitioner*, 80-86. New York, NY, USA: ACM.

[SLR205] Ling Huang, Jinhong Zhong, Zhiwei Ni, and Shanlin Yang, "Mixed Backlogging and Outsourcing Models with Nondecreasing Inventory Capacity: Intelligent Systems Design and Applications, 2006. ISDA '06. Sixth International Conference on," *Intelligent Systems Design and Applications, 2006. ISDA '06. Sixth International Conference on*, vol. 1, pp. 1161–1166, 2006.

[SLR206] Lings, B., Lundell, B., Ågerfalk, P., and Fitzgerald, B., "Ten Strategies for Successful Distributed Development," *The Transfer and Diffusion of Information Technology for Organizational Resilience*, pp. 119–137, 2006.

[SLR207] Lormans, M., van Dijk, H., van Deursen, A., Nocker, E., and Zeeuw, A. de, " Managing evolving requirements in an outsourcing context: an industrial experience report: Software Evolution, 2004. Proceedings. 7th International Workshop on Principles of," *Software Evolution, 2004. Proceedings. 7th International Workshop on Principles of*, pp. 149–158, 2004.

[SLR208] Macke, D. and Galinac, T., "Optimized Software Process for Fault Handling in Global Software Development," *Making Globally Distributed Software Development a Success Story*, pp. 395–406, 2008.

[SLR209] Maher S, "The outsourcing option [outsourcing of project administration]," *Australian Mining, vol*, no. Reed Business Information, pp. vol, 98, no.

[SLR210] Maldonado, J., Carver, J., Shull, F., Fabbri, S., Dória, E., Martimiano, L., Mendonça, M., and Basili, V., "Perspective-Based Reading: A Replicated Ex-

periment Focused on Individual Reviewer Effectiveness," *Empirical Software Engineering*, vol. 11, no. 1, pp. 119–142, Mar. 2006.

[SLR211] Maloney, M. and Zellmer-Bruhn, M., "Building bridges, windows and cultures: Mediating mechanisms between team heterogeneity and performance in global teams," *Management International Review*, vol. 46, no. 6, pp. 697–720, Dec. 2006.

[SLR212] Mann, A., "Information Technology and the Related Services Industry: Evaluating India's Success Factors: System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference on," *System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference on*, pp. 1–10, 2009.

[SLR213] Martin, D., Rooksby, J., Rouncefield, M., and Sommerville, I., "Cooperative work in software testing," 2008. In *CHASE '08: Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering*, 93-96. New York, NY, USA: ACM.

[SLR214] Martins, P. and da Silva, A., "SPI Methodology for Virtual Organizations," *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, pp. 824–833, 2006.

[SLR215] Marttiin, P., Lehto, J. A., and Nyman, G., "Understanding and evaluating collaborative work in multi-site software projects - a framework proposal and preliminary results: System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on," *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, pp. 283–292, 2002.

[SLR216] Mathrani, A. and Parsons, D., "Management of Knowledge Transfer in Distributed Software Organizations: The Outsourcers' Perspective," *Decision Support for Global Enterprises*, pp. 75–89, 2007.

[SLR217] Matsubara, T., "Japan: a huge IT consumption market: Software, IEEE," *Software, IEEE*, vol. 18, no. 5, pp. 77–80, 2001.

[SLR218] Maurer, F. and Holz, H., "Integrating process support and knowledge management for virtual software development teams," *Ann. Softw. Eng.*, vol. 14, no. 1-4, pp. 145–168, 2002.

[SLR219] Maurer, F. and Holz, H., "Integrating Process Support and Knowledge Management for Virtual Software Development Teams," *Annals of Software Engineering*, vol. 14, no. 1, pp. 145–168, Dec. 2002.

[SLR220] McGuire EG and McKeown KA, "Implementing practical metrics in an outsourcing environment," *Challenges of Information Technology Management in the 21st Century*, pp. pp, 2000 Information Resources Management Association International Conference.

[SLR221] McQuay, W. K., "Distributed collaborative environments for systems engineering," *IEEE Aerosp Electron Syst Mag*, vol. 20, no. 8 I, pp. 7–12, 2005.

[SLR222] Mei He, Mingshu Li, Qing Wang, Ye Yang, and Kai Ye, "An investigation of software development productivity in China," *Making Globally Distributed Software Development a Success Story*, pp. ICSP 2008, International Conference on Software Process, ICSP 2008.

[SLR223] Meister, D. and Gronski, C., "Action Research in A Virtual Setting," *Information Systems Action Research*, pp. 217–239, 2007.

[SLR224] Meyer, B., "Design and code reviews in the age of the internet," *Commun. ACM*, vol. 51, no. 9, pp. 66-71, 2008.

[SLR225] Mezgar, I., "Trust Building for Enhancing Collaboration in Virtual Organizations," *Network-Centric Collaboration and Supporting Frameworks*, pp. 173–180, 2006.

[SLR226] Midwinter, T. and Sheppard, P. J., "eCollaboration — 'The Drive for Simplicity'," *BT Technology Journal*, vol. 18, no. 2, pp. 107–115, Apr. 2000.

[SLR227] Mikawa, S. P., Cunnington, S. K., and Gaskins, S. A., "Removing barriers to trust in distributed teams: understanding cultural differences and strengthening social ties," 2009. In *IWIC '09: Proceeding of the 2009 international workshop on Intercultural collaboration*, 273-276. New York, NY, USA: ACM.

[SLR228] Mikulovic, V. and Heiss, M., "How do I know what I have to do?": the role of the inquiry culture in requirements communication for distributed software development projects," 2006. In *ICSE '06: Proceedings of the 28th international conference on Software engineering*, 921-925. New York, NY, USA: ACM.

[SLR229] Moe, N. and Šmite, D., "Understanding Lacking Trust in Global Software Teams: A Multi-case Study," *Product-Focused Software Process Improvement*, pp. 20–34, 2007.

[SLR230] Morgan, R., Walny, J., Kolenda, H., Ginez, E., and Maurer, F., "Using Horizontal Displays for Distributed and Collocated Agile Planning," *Agile Processes in Software Engineering and Extreme Programming*, pp. 38–45, 2007.

[SLR231] Mortensen, M., Woolley, A., and O'Leary, M., "Conditions Enabling Effective Multiple Team Membership," *Virtuality and Virtualization*, pp. 215–228, 2007.

[SLR232] Muhsin, Z. J., El-Mousa, A. H., and Al-Taee, M. A., "A distributed workflow management system utilizing computational intelligence for decision support," *Int. Multi-Conference Syst., Signals Devices, SSD*, 2008.

[SLR233] Mullally, B. and Stapleton, L., "Methodology Usage by Virtual Information Systems Development Teams," *Information Systems Development*, pp. 37–48, 2009.

[SLR234] Nair KGK and Prasad PN, "Offshore outsourcing: a swot analysis of a state in India," *Information Systems Management, vol*, no. Auerbach Publications, pp. vol, 21, no.

[SLR235] Nakayama, M., Binotto, E., and Pilla, B., "Trust in Virtual Teams: A Performance Indicator," *Education for the 21st Century — Impact of ICT and Digital Resources*, pp. 105–113, 2006.

[SLR236] Nguyen, T. N., "Component-Based Software Update Process in Collaborative Software Development: Software Engineering Conference, 2008. APSEC '08. 15th Asia-Pacific," *Software Engineering Conference, 2008. APSEC '08. 15th Asia-Pacific*, pp. 437–444, 2008.

[SLR237] Nguyen TN, "Component-based software update process in collaborative software development," *2008 15th Asia Pacific Software Engineering Conference*, pp. NJ USA, IEEE.

[SLR238] Nicolopoulou, K., Koštomaj, M., and Campos, A., "How to address group dynamics in virtual worlds," *AI & Society*, vol. 20, no. 3, pp. 351–371, Jun. 2006.

[SLR239] Nikula, U. and Sajaniemi, J., "Tackling the complexity of requirements engineering process improvement by partitioning the improvement task: Software Engineering Conference, 2005. Proceedings. 2005 Australian," *Software Engineering Conference, 2005. Proceedings. 2005 Australian*, pp. 48–57, 2005.

[SLR240] Nikula U and Sajaniemi J, "Tackling the complexity of requirements engineering process improvement by partitioning the improvement task," *Proceedings*, pp. pp, 2005 Australian Software Engineering Conference.

[SLR241] Ning, A., Hou, H., Hua, Q., Yu, B., and Hao, K., "Requirements Engineering Processes Improvement: A Systematic View," *Unifying the Software Process Spectrum*, pp. 151–163, 2006.

[SLR242] Nisar, M. F. and Hameed, T., "Agile methods handling offshore software development issues: Multitopic Conference, 2004. Proceedings of INMIC 2004. 8th International," *Multitopic Conference, 2004. Proceedings of INMIC 2004. 8th International*, pp. 417–422, 2004.

[SLR243] Noll, J. and Scacchi, W., "Supporting software development in virtual enterprises," *J. Digit. Inf.*, vol. 1, no. 4, 1999.

[SLR244] Ocker, R., Huang, H., Trauth, E., and Purao, S., "The Tension Between Expectations of Availability and the Reality of Availability in Hybrid Teams," *Virtuality and Virtualization*, pp. 119–131, 2007.

[SLR245] Ocker, R. J. and Fjermestad, J., "Communication differences in virtual design teams: findings from a multi-method analysis of high and low performing experimental teams," *SIGMIS Database*, vol. 39, no. 1, pp. 51-67, 2008.

[SLR246] Ocker, R. J., "Influences on creativity in asynchronous virtual teams: a qualitative analysis of experimental teams: Professional Communication, IEEE Transactions on," *Professional Communication, IEEE Transactions on*, vol. 48, no. 1, pp. 22–39, 2005.

[SLR247] Oshri, I., Kotlarsky, J., and Willcocks, L., "Missing links: building critical social ties for global collaborative teamwork," *Commun. ACM*, vol. 51, no. 4, pp. 76-81, 2008.

[SLR248] Oza NV, Hall T, Rainer A, and Grey S, "Trust in software outsourcing relationships: An empirical investigation of Indian software companies," *Information and Software Technology, vol*, no. Elsevier, pp. vol, 48, no.

[SLR249] Papadaki, K., Polemi, N., and Damilos, D. K., "A Meta-process for Information Security Risk Management," *Global E-Security*, pp. 257–264, 2008.

[SLR250] Papadakis A, Psaromiligkos Y, and Crowe M, "A Web-based software configuration management tool," *Proceedings of the 2nd International Scientific Conference on Information Technology and Quality*, pp. pp, University of Paisley.

[SLR251] Paravicini M, "Offshore 2.0," *Information Age, June 2008, pp*, no. Infoconomy Ltd., pp. June 2008 pp, 16-18.

[SLR252] Paulish, D. J., "Methods and Tools for Collaboration in GSE Environments: Global Software Engineering, 2006. ICGSE '06. International Conference on, " *Global Software Engineering, 2006. ICGSE '06. International Conference on*, pp. 233-233, 2006.

[SLR253] Pesendorfer, E.-M., Graf, A., and Koeszegi, S., "Relationship in electronic negotiations: Tracking behavior over time," *Zeitschrift für Betriebswirtschaft*, vol. 77, no. 12, pp. 1315–1338, Dec. 2007.

[SLR254] Phillips D, "Working with distributed teams: or managing virtual teams," *Information Technology and Project Management '96*, pp. pp, Conference Proceedings.

[SLR255] Pichler, H., "Be successful, take a hostage or "outsourcing the outsourcing Manager": Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on," *Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on*, pp. 156–161, 2007.

[SLR256] Pilatti L and Audy JLN, "Software offshore development: a criteria definition for references models comparison," *ICEIS 2008*, pp. Control and Communication, Proceedings of the Tenth International Conference on Enterprise Information Systems.

[SLR257] Pilatti L and Nicolas Audy JL, "Global software development offshore insourcing organizations characteristics: lessons learned from a case study," *2006 1st IEEE International Conference on Global Software Engineering*, pp. pp, IEEE Computer Society.

[SLR258] Pillis, E. de and Furumo, K., "Counting the cost of virtual teams," *Commun. ACM*, vol. 50, no. 12, pp. 93-95, 2007.

[SLR259] Poehler, L. and Schumacher, T., "The Virtual Team Challenge: Is It Time for Training? Management of Engineering and Technology, Portland Interna-

tional Center for," *Management of Engineering and Technology, Portland International Center for*, pp. 2205–2211, 2007.

[SLR260] Potter RE and Balthazard PA, "Virtual team interaction styles: assessment and effects," *International Journal of Human Computer Studies, vol*, no. Academic Press, pp. vol, 56, no.

[SLR261] Prasad, B., "System Integration Techniques of Sharing and Collaboration among Work-groups, Computers and Processes," *Journal of Systems Integration*, vol. 9, no. 2, pp. 115–139, Aug. 1999.

[SLR262] Prikladnicki R, Audy JLN, and Evaristo R, "Global software development in practice lessons learned," *Software Process Improvement and Practice, vol*, no. Wiley, pp. vol, 8, no.

[SLR263] Prikladnicki R, Evaristo R, Damian D, and Audy JLN, "Conducting qualitative research in an international and distributed research team: challenges and lessons learned," *2008 41st Annual Hawaii International Conference on System Sciences*, pp. NJ USA, IEEE.

[SLR264] "Product-Focused Software Process Improvements. Proceedings 8th International Conference, PROFES 2007," *Springer Verlag*, pp. Germany, Berlin, Germany.

[SLR265] Purvis, M., Purvis, M., and Cranefield, S., "Educational experiences from a Global Software Engineering (GSE) project," 2004. In *ACE '04: Proceedings of the sixth conference on Australasian computing education*, 269-275. Darlinghurst, Australia, Australia: Australian Computer Society, Inc.

[SLR266] Ralyte, J., Lamielle, X., Arni-Bloch, N., and Leonard, M., "A framework for supporting management in distributed information systems development: Research Challenges in Information Science, 2008. RCIS 2008. Second International Conference on," *Research Challenges in Information Science, 2008. RCIS 2008. Second International Conference on*, pp. 381–392, 2008.

[SLR267] Rama, J. and Bishop, J., "A survey and comparison of CSCW groupware applications," 2006. In *SAICSIT '06: Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*, 198-205. Republic of South Africa: South African Institute for Computer Scientists and Information Technologists.

[SLR268] Ramasubbu N, Krishnan MS, and Kompalli P, "Leveraging global resources: a process maturity framework for managing distributed development," *IEEE Software, vol*, no. IEEE, pp. vol, 22, no.

[SLR269] Rao, B. and Mulloth, B., "Building a Global Brand: The Case of Wipro: Management of Engineering and Technology, Portland International Center for, " *Management of Engineering and Technology, Portland International Center for*, pp. 1372–1385, 2007.

[SLR270] Rao, B. S. and Sarda, N. L., "Effort drivers in maintenance outsourcing-an experiment using Taguchi's methodology: Software Maintenance and Reengineering, 2003. Proceedings. Seventh European Conference on," *Software Maintenance and Reengineering, 2003. Proceedings. Seventh European Conference on*, pp. 271–280, 2003.

[SLR271] Ratcheva, V. and Vyakarnam, S., "The challenges of virtual partnerships: Critical success factors in the formation of inter-organisational teams," *AI & Society*, vol. 15, no. 1, pp. 99–116, Mar. 2001.

[SLR272] Rees, M. J., "A feasible user story tool for agile software development? Software Engineering Conference, 2002. Ninth Asia-Pacific," *Software Engineering Conference, 2002. Ninth Asia-Pacific*, pp. 22–30, 2002.

[SLR273] Reijers, H., Song, M., and Jeong, B., "Analysis of a collaborative workflow process with distributed actors," *Information Systems Frontiers*.

[SLR274] Reijers, H., Song, M., and Jeong, B., "On the Performance of Workflow Processes with Distributed Actors: Does Place Matter?" *Business Process Management*, pp. 32–47, 2007.

[SLR275] Rice, D., Davidson, B., Dannenhoffer, J., and Gay, G., "Improving the Effectiveness of Virtual Teams by Adapting Team Processes," *Computer Supported Cooperative Work (CSCW)*, vol. 16, no. 6, pp. 567–594, Dec. 2007.

[SLR276] Roehling ST, Collofello JS, Hermann BG, and Smith Daniels DE, "System dynamics modeling applied to software outsourcing decision support," *Software Process Improvement and Practice, vol*, no. Wiley, pp. vol, 5, no.

[SLR277] Romero, M., Vizca\ino, A., and Piattini, M., "Toward a definition of the competences for global requirements elicitation," 2008. In *ITiCSE '08: Proceedings of the 13th annual conference on Innovation and technology in computer science education*, 364-364. New York, NY, USA: ACM.

[SLR278] Sabetzadeh, M., Nejati, S., Easterbrook, S., and Chećhik, M., "Global consistency checking of distributed models with TReMer+," 2008. In *ICSE Companion '08: Companion of the 30th international conference on Software engineering*, 815-818. New York, NY, USA: ACM.

[SLR279] Sakthivel, S., "Managing risk in offshore systems development," *Commun. ACM*, vol. 50, no. 4, pp. 69-75, 2007.

[SLR280] Salisbury, W. D., Carte, T. A., and Chidambaram, L., "Cohesion in virtual teams: validating the perceived cohesion scale in a distributed setting," *SIGMIS Database*, vol. 37, no. 2-3, pp. 147-155, 2006.

[SLR281] Sangwan, R. S. and Ros, J., "Architecture leadership and management in globally distributed software development," 2008. In *LMSA '08: Proceedings of the first international workshop on Leadership and management in software architecture*, 17-22. New York, NY, USA: ACM.

[SLR282] Schadewitz, N. and Zakaria, N., "Cross-cultural collaboration Wiki: evolving knowledge about international teamwork," 2009. In *IWIC '09: Proceeding of the 2009 international workshop on Intercultural collaboration*, 301-304. New York, NY, USA: ACM.

[SLR283] Sebrechts, M. M., "Visualizing Information in Virtual Space: Prospects and Pitfalls," *Knowledge and Information Visualization*, pp. 136–166, 2005.

[SLR284] Seifert, M. and Eschenbaecher, J., "Predictive Performance Measurement in Virtual Organisations," *Emerging Solutions for Future Manufacturing Systems*, pp. 299–306, 2005.

[SLR285] Sengupta, B., Chandra, S., and Sinha, V., "A research agenda for distributed software development," 2006. In *ICSE '06: Proceedings of the 28th international conference on Software engineering*, 731-740. New York, NY, USA: ACM.

[SLR286] Sengupta, K. and Leon Zhao, J., "Designing workflow management systems for virtual organizations: an empirically grounded approach: System Sciences, 1998., Proceedings of the Thirty-First Hawaii International Conference on," *System Sciences, 1998., Proceedings of the Thirty-First Hawaii International Conference on*, vol. 4, pp. 365-373 vol.4, 1998.

[SLR287] Setamanit, S.-o. and Raffo, D., "Identifying Key Success Factors for Globally Distributed Software Development Project Using Simulation: A Case Study," *Making Globally Distributed Software Development a Success Story*, pp. 320–332, 2008.

[SLR288] Setamanit, S.-o., Wakeland, W., and Raffo, D., "Exploring the Impact of Task Allocation Strategies for Global Software Development Using Simulation," *Software Process Change*, pp. 274–285, 2006.

[SLR289] Setamanit, S.-o., Wakeland, W., and Raffo, D., "Planning and improving global software development process using simulation," 2006. In *GSD '06: Proceedings of the 2006 international workshop on Global software development for the practitioner*, 8-14. New York, NY, USA: ACM.

[SLR290] Seybold, C. and Keller, R. K., "Aligning Software Maintenance to the Offshore Reality: Software Maintenance and Reengineering, 2008. CSMR 2008. 12th European Conference on," *Software Maintenance and Reengineering, 2008. CSMR 2008. 12th European Conference on*, pp. 33–42, 2008.

[SLR291] Sharp, J. H. and Ryan, S. D., "A Preliminary Conceptual Model for Exploring Global Agile Teams," *Agile Processes in Software Engineering and Extreme Programming*, pp. 147–160, 2008.

[SLR292] Shen, Y., "Transactive memory system development in virtual teams: the potential role of shared identity and shared context," 2007. In *SIGMIS-CPR '07: Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel research*, 228-230. New York, NY, USA: ACM.

[SLR293] Siakas KV, Maoutsidis D, and Siakas E, "Trust facilitating good software outsourcing relationships," *Software Process Improvement*, pp. EuroSPI 2006, 13th European Conference, EuroSPI 2006.

[SLR294] Singh, L. P., Bhardwaj, A., and Sachdeva, A., "The Impact of ISO Implementation on Output Parameters in SME's in India: Management of Engineering and Technology, Portland International Center for," *Management of Engineering and Technology, Portland International Center for*, pp. 2031–2037, 2007.

[SLR295] Singh, Y., Bhatia, P. K., Kaur, A., and Sangwan, O., "Application of Neural Networks in Software Engineering: A Review," *Information Systems, Technology and Management*, pp. 128–137, 2009.

[SLR296] Siva Namasivayam, "Profiting from business process outsourcing: IT Professional," *IT Professional*, vol. 6, no. 1, pp. 12–18, 2004.

[SLR297] Šmite, D., Moe, N., and Torkar, R., "Pitfalls in Remote Team Coordination: Lessons Learned from a Case Study," *Product-Focused Software Process Improvement*, pp. 345–359, 2008.

[SLR298] Smite D and Borzovs J, "A framework for overcoming supplier related threats in global projects," *Software Process Improvement*, pp. EuroSPI 2006, 13th European Conference, EuroSPI 2006.

[SLR299] Smith, J. L., Bohner, S. A., and McCrickard, D. S., "Project management for the 21st century: supporting collaborative design through risk analysis," 2005. In *ACM-SE 43: Proceedings of the 43rd annual Southeast regional conference*, 300-305. New York, NY, USA: ACM.

[SLR300] Smith, J. L., Bohner, S. A., and McCrickard, D. S., "Toward introducing notification technology into distributed project teams: Engineering of Computer-Based Systems, 2005. ECBS '05. 12th IEEE International Conference and Workshops on the," *Engineering of Computer-Based Systems, 2005. ECBS '05. 12th IEEE International Conference and Workshops on the*, pp. 349–356, 2005.

[SLR301] Smith, J. L., Bohner, S. A., Scott McCrickard, D., Rozenblit J., O'Neill T., and Peng J., "Toward introducing notification technology into distributed project teams," *Proc. Eng. Comput. Based Syst.*, pp. 349–356, 2005.

[SLR302] Steinfield, C., Jang, C.-Y., and Pfaff, B., "Supporting virtual team collaboration: the TeamSCOPE system," 1999. In *GROUP '99: Proceedings of the international ACM SIGGROUP conference on Supporting group work*, 81-90. New York, NY, USA: ACM.

[SLR303] Sureshchandra, K. and Shrinivasavadhani, J., "Adopting Agile in Distributed Development: Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on," *Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on*, pp. 217–221, 2008.

[SLR304] Suzuki, J. and Yamamoto, Y., "Leveraging distributed software development: Computer," *Computer*, vol. 32, no. 9, pp. 59–65, 1999.

[SLR305] Tanger, R., Kauff, P., Schreer, O., Pavy, D., Louis Dit Picard, S., and Saugis, G., "Team Collaboration Mixing Immersive Video Conferencing With Shared

Virtual 3D Objects," *Distributed Cooperative Laboratories: Networking, Instrumentation, and Measurements*, pp. 3–10, 2006.

[SLR306] Taylor, P., Des Greer, Sage, P., Coleman, G., McDaid, K., Lawthers, I., and Corr, R., "Applying an Agility/Discipline Assessment for a Small Software Organisation," *Product-Focused Software Process Improvement*, pp. 290–304, 2006.

[SLR307] Thissen, M. R., Page, J. M., Bharathi, M. C., and Austin, T. L., " Communication tools for distributed software development teams," 2007. In *SIGMIS-CPR '07: Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel research*, 28-35. New York, NY, USA: ACM.

[SLR308] Thomas, D. M., Bostrom, R. P., and Gouge, M., "Making knowledge work in virtual teams," *Commun. ACM*, vol. 50, no. 11, pp. 85-90, 2007.

[SLR309] Thomsen J, Levitt RE, and Nass CI, "The virtual team alliance (VTA): extending Galbraith's information-processing model to account for goal incongruency," *Computational & Mathematical Organization Theory, vol*, no. Kluwer Academic Publishers, pp. vol, 10, no.

[SLR310] Tindell, K., Kopetz, H., Wolf, F., and Ernst, R., "Safe Automotive Software Development," 2003. In *DATE '03: Proceedings of the conference on Design, Automation and Test in Europe*, 10616. Washington, DC, USA: IEEE Computer Society.

[SLR311] Trammell, J., "Application Response Time SLA's: Lessons Learned by a Major Outsourcer: Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP," *Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP*, pp. 1–15, 2006.

[SLR312] Tremaine, M., Milewski, A., Egan, R., and Zhang, S., "A Tale of Two Teams: Success and Failure in Virtual Team Meetings," *Usability and Internationalization. HCI and Culture*, pp. 442–451, 2007.

[SLR313] Tsai, W., Xiao, B., Huang, Q., and Chen, Y., "Collaborative software design in an SOA environment," *Science in China Series F: Information Sciences*, vol. 49, no. 6, pp. 821–842, Dec. 2006.

[SLR314] Ungerleider, J., "Conflict," *Effective Multicultural Teams: Theory and Practice*, pp. 211–238, 2008.

[SLR315] Urdangarin R, Fernandes P, Avritzer A, and Paulish D, "Experiences with agile practices in the global studio project," *2008 IEEE International Conference on Global Software Engineering (ICGSE)*, pp. NJ USA, IEEE.

[SLR316] Valetto, G., Helander, M., Ehrlich, K., Chulani, S., Wegman, M., and Williams, C., "Using Software Repositories to Investigate Socio-technical Congruence in Development Projects," 2007. In *MSR '07: Proceedings of the Fourth International Workshop on Mining Software Repositories*, 25. Washington, DC, USA: IEEE Computer Society.

[SLR317] Välimäki, A. and Kääriäinen, J., "Patterns for Distributed Scrum — A Case Study," *Enterprise Interoperability III*, pp. 85–97, 2008.

[SLR318] Välimäki, A. and Kääriäinen, J., "Requirements Management Practices as Patterns for Distributed Product Management," *Product-Focused Software Process Improvement*, pp. 188–200, 2007.

[SLR319] van de Burgt B, "The effect of outsourcing and offshoring on tests," *Informatie, vol*, no. Kluwer Bedrijfswetenschappen, pp. vol, 48, no.

[SLR320] van der Duim, L., Andersson, J., and Sinnema, M., "Good Practices for Educational Software Engineering Projects," 2007. In *ICSE '07: Proceedings of the 29th international conference on Software Engineering*, 698-707. Washington, DC, USA: IEEE Computer Society.

[SLR321] van Deursen, A., Parviainen, P., Kommeren, R., and van Solingen, R., " Workshop on technology for supporting software engineers in globally distributed contexts," *Proc. - Int. Conf. Global Softw. Eng., ICGSE*, pp. 286–287, 2007.

[SLR322] Varma, P., "Stub Libraries for Software Migration and Development: Software Technology and Engineering Practice, 2005. 13th IEEE International Workshop on," *Software Technology and Engineering Practice, 2005. 13th IEEE International Workshop on*, pp. 118–126, 2005.

[SLR323] Vitale V, "IBM software development leveraging geographically distributed teams - the Interactive Solution Marketplace 2.0 (ISM) case study," *2006 1st IEEE International Conference on Global Software Engineering*, pp. pp, IEEE Computer Society.

[SLR324] Vizcaino A, Favela J, Piattini M, and Garcia F, "Supporting software maintenance in Web repositories through a multi-agent system," *Advances in Web Intelligence*, pp. pp, First International Atlantic Web Intelligence Conference.

[SLR325] Wahyudin, D., Heindl, M., Eckhard, B., Schatten, A., and Biffl, S., "In-Time Role-Specific Notification as Formal Means to Balance Agile Practices in Global Software Development Settings," *Balancing Agility and Formalism in Software Engineering*, pp. 208–222, 2008.

[SLR326] Weber DW, "Data topology and process patterns for distributed development," *Software Configuration Management*, pp. pp, ICSE Workshops SCM 2001 and SCM 2003.

[SLR327] Wei, K., "Sharing Knowledge in Global Virtual Teams," *Virtuality and Virtualization*, pp. 251–265, 2007.

[SLR328] Weihua Gan, "Analysis on the Costs of IT-Outsourcing: Service Operations and Logistics, and Informatics, 2006. SOLI '06. IEEE International Conference on," *Service Operations and Logistics, and Informatics, 2006. SOLI '06. IEEE International Conference on*, pp. 785–789, 2006.

[SLR329] Weitzel T, Konig W, and Busse R, "Towards an industrialization of financial processes - Web service based external straight through processing for Master Trusts," *Information Management & Consulting, vol*, no. IMC GmbH, pp. vol, 19, no.

[SLR330] Whitehead, J., "Collaboration in Software Engineering: A Roadmap," 2007. In *FOSE '07: 2007 Future of Software Engineering*, 214-225. Washington, DC, USA: IEEE Computer Society.

[SLR331] Wiegers K, "See you in court [outsourcing]," *Software Development, vol*, no. CMP Media Inc, pp. vol, 11, no.

[SLR332] Wilson JM, "Making a case for outsourcing," *Proceedings*, pp. pp, Conference.

[SLR333] Wong, F., Fernandez, G., and McGovern, J., "CONFER: towards groupware for building consensus in collaborative software engineering," 2007. In *AUIC '07: Proceedings of the eight Australasian conference on User interface*, 31-38. Darlinghurst, Australia, Australia: Australian Computer Society, Inc.

[SLR334] Wongthongtham, P., Chang, E., and Aseeri, A., "Towards Social Network Based Approach for Software Engineering Ontology Sharing and Evolution," *On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops*, pp. 1233–1243, 2007.

[SLR335] Wongthongtham, P., Chang, E., and Dillon, T., "Multi-site Distributed Software Development: Issues, Solutions, and Challenges," *Computational Science and Its Applications – ICCSA 2007*, pp. 346–359, 2007.

[SLR336] Xu, B., Hu, H., Ling, Y., Yang, X., He, Z., and Ma, A., "Achieving Better Collaboration in Global Software Design with Micro Estimation," *Computer Supported Cooperative Work in Design III*, pp. 357–366, 2007.

[SLR337] Yalaho, A. and Nahar, N., "The ICT-Supported Unified Process Model of Offshore Outsourcing of Software Production: Exploratory Examination and Validation: Management of Engineering and Technology, Portland International Center for," *Management of Engineering and Technology, Portland International Center for*, pp. 1466–1483, 2007.

[SLR338] Yilmaz, C., Krishna, A. S., Memon, A., Porter, A., Schmidt, D. C., Gokhale, A., and Natarajan, B., "Main effects screening: a distributed continuous quality assurance process for monitoring performance degradation in evolving software systems," 2005. In *ICSE '05: Proceedings of the 27th international conference on Software engineering*, 293-302. New York, NY, USA: ACM.

[SLR339] Zhenyuan Liu and Yanqi Zheng, "Resource-constrained multiple projects scheduling with the objective of minimizing activities cost: Control and Decision Conference, 2008. CCDC 2008. Chinese," *Control and Decision Conference, 2008. CCDC 2008. Chinese*, pp. 1027–1032, 2008.

[SLR340] Zwanzig C, Kirchhoff U, and Schulz H S, "Software process improvement experiment concerning effective outsourcing mechanisms (ESSI project no. 24176 OutSource)," *European Software Measurement Conference*, no. European Software Control and Measurement Conference). Proceedings . Technologisch Instituut Vzw. 1998, pp. pp, FESMA 98.

Bibliography

[1] J.-N. Lee, M. Q. Huynh, R. C.-W. Kwok, and S.-M. Pi, "IT outsourcing evolution---," *Commun. ACM*, vol. 46, no. 5, pp. 84–89, May 2003.

[2] P. Keil, D. J. Paulish, and R. S. Sangwan, "Cost estimation for global software development," *Proc. 2006 Int. Work. Econ. driven Softw. Eng. Res. - EDSER '06*, p. 7, 2006.

[3] W. Gan, "Analysis on the Costs of IT-Outsourcing," *2006 IEEE Int. Conf. Serv. Oper. Logist. Informatics*, pp. 785–789, Nov. 2006.

[4] K. G. K. Nair and P. N. Prasad, "Offshore Outsourcing: A Swot Analysis of a State in India," *Inf. Syst. Manag.*, vol. 21, no. 3, pp. 34–40, Jun. 2004.

[5] B. L. Kirkman and C. B. Gibson, "THE IMPACT OF TEAM EMPOWERMENT ON VIRTUAL TEAM PERFORMANCE : THE MODERATING ROLE OF FACE-TO-FACE INTERACTION University of North Carolina at Chapel Hill," *Acad. Manag. J.*, vol. 47, no. 2, pp. 175–192, 2004.

[6] R. Kommeren and P. Parviainen, "Philips experiences in global distributed software development," *Empir. Softw. Eng.*, vol. 12, no. 6, pp. 647–660, Sep. 2007.

[7] S. L. Jarvenpaa and D. E. Leidner, "Communication and Trust in Global Virtual Teams," *Organ. Sci.*, vol. 10, no. 6, pp. 791–815, Nov. 1999.

[8] T. M. Griffin, "Effective Intercultural Communication," *Intercult. Commun.*, pp. 173–210, 2008.

[9] C. Gutwin, R. Penner, and K. Schneider, "Group awareness in distributed software development," *Proc. 2004 ACM Conf. Comput. Support. Coop. Work - CSCW '04*, p. 72, 2004.

[10] D. G Boyer, M. J Handel, and J. Herbsleb, "Virtual Community Prescence Awareness," *ACM Siggr. Bull.*, vol. 19, no. 3, pp. 1–4, 1998.

[11] A. Draghici, "Influence of Multi-Culturality in Virtual Teams," *Knowl. Creat. Diffus. Util.*

[12] A. S. Bachmann, "Melting pot or tossed salad? Implications for designing effective multicultural workgroups," *Manag. Int. Rev.*, vol. 46, no. 6, pp. 721–748, Dec. 2006.

[13] N. Schadewitz, “Cross-cultural Collaboration Wiki – Evolving Knowledge about International Teamwork,” *Int. Bus.*, pp. 301–304, 2009.

[14] G. Piccoli and B. Ives, “Trust and the unintended effects of behavior control in virtual teams1,” *MIS Q.*, vol. 27, no. 3, 2003.

[15] S. L. Jarvenpaa, T. R. Shaw, and D. S. Staples, “Toward Contextualized Theories of Trust: The Role of Trust in Global Virtual Teams,” *Inf. Syst. Res.*, vol. 15, no. 3, pp. 250–267, Sep. 2004.

[16] P. Kanawattanachai, “Dynamic nature of trust in virtual teams,” *J. Strateg. Inf. Syst.*, vol. 11, no. 3–4, pp. 187–213, Dec. 2002.

[17] D. ŠMITE, “GLOBAL SOFTWARE DEVELOPMENT IMPROVEMENT,” UNIVERSITY OF LATVIA, 2007.

[18] A. Begel, “Effecting Change : Coordination in Large-Scale Software Development,” *CHASE 08 Proc. 2008 Int. Work. Coop. Hum. Asp. Softw. Eng.*, pp. 17–20, 2008.

[19] D. Damian, A. Hadwin, and B. Al-Ani, “Instructional design and assessment strategies for teaching global software development,” *Proceeding 28th Int. Conf. Softw. Eng. - ICSE ’06*, p. 685, 2006.

[20] M. R. Thissen, J. M. Page, M. C. Bharathi, and T. L. Austin, “Communication tools for distributed software development teams,” *Proc. 2007 ACM SIGMIS CPR Conf. 2007 Comput. Pers. Dr. Consort. Res. Conf. Glob. Inf. Technol. Work. - SIGMIS-CPR ’07*, p. 28, 2007.

[21] M. M. Tremaine, A. Milewski, R. Egan, and S. Zhang, “A Tale of Two Teams : Success and Failure in Virtual Team Meetings,” *Manager*, pp. 442–451, 2007.

[22] J. D. Herbsleb and A. Mockus, “An empirical study of speed and communication in globally distributed software development,” *IEEE Trans. Softw. Eng.*, vol. 29, no. 6, pp. 481–494, Jun. 2003.

[23] G. Hertel, S. Geister, and U. Konradt, “Managing virtual teams: A review of current empirical research,” *Hum. Resour. Manag. Rev.*, vol. 15, no. 1, pp. 69–95, Mar. 2005.

[24] C. M. Beise, “IT project management and virtual teams,” *Proc. 2004 Conf. Comput. Pers. Res. Careers, Cult. ethics a networked Environ. - SIGMIS CPR ’04*, p. 129, 2004.

[25] D. DeLuca and J. S. Valacich, “Virtual teams in and out of synchronicity,” *Inf. Technol. People*, vol. 19, no. 4, pp. 323–344, 2006.

[26] V. Casey and I. Richardson, “Project Management within Virtual Software Teams,” *2006 IEEE Int. Conf. Glob. Softw. Eng.*, pp. 33–42, Oct. 2006.

[27] C. Ebert, “Global Software Engineeringb,” *IEEE ReadyNote*, 2006.

[28] J. L. Smith, S. A. Bohner, and D. S. Mccrickard, “Project Management for the 21 st Century : Supporting Collaborative Design through Risk Analysis,” *Design*, pp. 300–305, 2005.

[29] J. Li, M. Li, Z. Wu, and Q. Wang, “A Metamodel for the CMM Software Process *,” *Program*, vol. 1, no. 60273026, pp. 446–450, 2004.

[30] Z. Haag, R. Foley, J. Newman, and G. G, “A Deontic Formalism for Co-ordinating Software Development in Virtual Software Corporations Department of Computer Studies 70 Cowcaddens Road,” *Virtual Softw. Corp. Proc. WET ICE 98, IEEE*.

[31] M. A. Babar and M. Zahedi, “Understanding Structures and Affordances of Extended Teams in Global Software Development,” *2013 IEEE 8th Int. Conf. Glob. Softw. Eng.*, pp. 226–235, Aug. 2013.

[32] J. Xiao and Q. Wang, “Contract-Driven Cooperation Development Process for Software Outsourcing,” *Fifth Int. Conf. Comput. Inf. Technol.*, pp. 1104–1108, 2005.

[33] K. Mu, W. Liu, Z. Jin, R. Lu, A. Yue, and D. Bell, “A Merging-Based Approach to Handling Inconsistency in Locally Prioritized Software Requirements,” *Viewpoints*, pp. 103–114, 2007.

[34] Y. Wang and D. Redmiles, “Understanding cheap talk and the emergence of trust in global software engineering: An evolutionary game theory perspective,” *2013 6th Int. Work. Coop. Hum. Asp. Softw. Eng.*, pp. 149–152, May 2013.

[35] J. M. Bhat, M. Gupta, and S. N. Murthy, “Overcoming Requirements Engineering Challenges: Lessons from Offshore Outsourcing,” *IEEE Softw.*, vol. 23, no. 5, pp. 38–44, Sep. 2006.

[36] E. Carmel, “Global software teams: collaborating across borders and time zones,” vol. ISBN:0–13-, Jan. 1999.

[37] P. M. Alexander, “Teamwork , Time , Trust and Information,” *Inf. Syst.*, pp. 65 – 74, 2002.

[38] M. J. Hawthorne and D. E. Perry, “Software engineering education in the era of outsourcing, distributed development, and open source software: challenges and opportunities,” *Proceedings. 27th Int. Conf. Softw. Eng. 2005. ICSE 2005.*, pp. 643–644.

[39] L. van der Duim, J. Andersson, and M. Sinnema, *Good Practices for Educational Software Engineering Projects*. IEEE, 2007, pp. 698–707.

[40] O. Gotel, V. Kulkarni, L. C. Neak, C. Scharff, and S. Seng, "Introducing Global Supply Chains into Software Engineering Education," no. September 2006, pp. 44–58.

[41] K. T. Dirks and D. L. Ferrin, "The Role of Trust in Organizational Settings," *Organ. Sci.*, vol. 12, no. 4, pp. 450–467, Jul. 2001.

[42] D. Bandow, "Time to Create Sound Teamwork," *J. Qual. Particip.*, vol. 24, no. 2, pp. 41–47.

[43] K. T. Chang, "Out of Sight but Not Out of Mind ? Informal Networks , Communication and Media Use in Global Software Teams 1 Introduction," *Networks*, pp. 1–12, 2007.

[44] Y. Cao, F. Sun, L. Wang, and T. Chen, "Promotion of Offshore Software Outsourcing Enterprise Value Chain Based on Knowledge Transfer Effects," *2008 4th Int. Conf. Wirel. Commun. Netw. Mob. Comput.*, pp. 1–4, Oct. 2008.

[45] M. E. Helander and N. Ohlsson, "Planning models for software reliability and cost," *IEEE Trans. Softw. Eng.*, vol. 24, no. 6, pp. 420–434, Jun. 1998.

[46] M. Adya, D. Nath, A. Malik, and V. Sridhar, *Bringing global sourcing into the classroom*. New York, New York, USA: ACM Press, 2007, p. 20.

[47] R. S. Sangwan and J. Ros, *Architecture leadership and management in globally distributed software development*. New York, New York, USA: ACM Press, 2008, p. 17.

[48] A. Nguyen-Duc and D. S. Cruzes, "Coordination of Software Development Teams across Organizational Boundary -- An Exploratory Study," *2013 IEEE 8th Int. Conf. Glob. Softw. Eng.*, pp. 216–225, Aug. 2013.

[49] M. Hackett, *Building Effective Global Software Test Teams through Training*. IEEE, 2007, pp. 293–294.

[50] S. S. W. Raffo, "Planning and improving global software development process using simulation," in in *Proceedings of the 2006 international workshop on Global software development for the practitioner*, Shanghai, China: ACM, 2006, pp. 8–14 TS – EndNote Tagged Import Format.

[51] L. Poehler and T. Schumacher, "The Virtual Team Challenge: Is It Time for Training?," *PICMET '07 - 2007 Portl. Int. Conf. Manag. Eng. Technol.*, pp. 2205–2211, Aug. 2007.

[52] D. J. Rice, B. D. Davidson, J. F. Dannenhoffer, and G. K. Gay, "Improving the Effectiveness of Virtual Teams by Adapting Team Processes," *Comput. Support. Coop. Work*, vol. 16, no. 6, pp. 567–594, Sep. 2007.

[53] J. Ding, "A framework for global collaboration in teaching software engineering," *2013 3rd Int. Work. Collab. Teach. Glob. Distrib. Softw. Dev.*, pp. 30–34, May 2013.

[54] N. Bos, N. S. Shami, J. S. Olson, A. Cheshin, and N. Nan, *In-group/out-group effects in distributed teams*. New York, New York, USA: ACM Press, 2004, p. 429.

[55] S. Fricker, T. Gorschek, and P. Myllyperkiö, "Handshaking between software projects and stakeholders using implementation proposals," *Proc. 13th Int. Work. Conf. Requir. Eng. Found. Softw. Qual.*, vol. 4542, pp. 144–159, 2007.

[56] K. Nicolopoulou, M. Koštomaj, and A. Campos, "How to address group dynamics in virtual worlds," *AI Soc.*, vol. 20, no. 3, pp. 351–371, Feb. 2006.

[57] M. Sinnema, "Good practices for Educational Software Engineering Projects Louwarnoud van der Duim," *Engineering*, 2007.

[58] I. Oshri, J. Kotlarsky, and L. Willcocks, "Missing links," *Commun. ACM*, vol. 51, no. 4, pp. 76–81, Apr. 2008.

[59] J. B. Ellis, K. Luther, K. Bessiere, and W. a. Kellogg, "Games for virtual team building," *Proc. 7th ACM Conf. Des. Interact. Syst. - DIS '08*, pp. 295–304, 2008.

[60] J. B. Ellis, S. Wahid, C. Danis, W. A. Kellogg, and V. Tech, "Task and Social Visualization in Software Development : Evaluation of a Prototype," pp. 577–586, 2007.

[61] S. P. Mikawa, S. K. Cunnington, and S. A. Gaskins, "Removing Barriers to Trust in Distributed Teams : Understanding Cultural Differences and Strengthening Social Ties," *Analysis*, pp. 273–276, 2009.

[62] B. I. Oshri, J. Kotlarsky, and L. Willcocks, "Missing Links : CRITICAL SOCIAL TIES for," *Commun. ACM*.

[63] S. P. Fang He, "Time Pressure and Reward Inspiration as Outcome Controls for Information Sharing in Problem-Solving Virtual Teams." IEEE Computer Society, 03-Jan-2007.

[64] A. Ferscha and C. Scheiner, "Collective Choice in Virtual Teams," pp. 96–101, Jun. 1999.

[65] A. Mathrani and D. Parsons, "Management of Knowledge Transfer in Distributed Software Organizations : The Outsourcers ' Perspective," *Development*.

[66] V. Ratcheva and S. Vyakarnam, “The challenges of virtual partnerships: Critical success factors in the formation of inter-organisational teams,” *AI Soc.*, vol. 15, no. 1–2, pp. 99–116, Mar. 2001.

[67] D. Šmite and J. Borzovs, *Software Process Improvement*, vol. 4257. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 50–61.

[68] K. Wei, “Sharing Knowledge in Global Virtual Teams How Do Chinese Team Members Perceive the Impact of,” vol. 236, pp. 251–265.

[69] E. Lindqvist and B. Lings, “Distributed Development in an Intra-national , Intra-organisational Context : An Experience Report,” *Hum. Factors*, pp. 80–86, 2006.

[70] B. Lings, B. Lundell, and B. Fitzgerald, “TEN STRATEGIES FOR SUCCESSFUL,” *Technology*, vol. 206, pp. 119–137.

[71] R. S. Sangwan and J. Ros, “Architecture leadership and management in globally distributed software development,” *Proc. first Int. Work. Leadersh. Manag. Softw. Archit. - LMSA ’08*, p. 17, 2008.

[72] V. Casey and I. Richardson, “Virtual teams: understanding the impact of fear,” *Softw. Process Improv. Pract.*, vol. 13, no. 6, pp. 511–526, Nov. 2008.

[73] M. A. Babar and M. Niazi, “Implementing Software Process Improvement Initiatives: An Analysis of Vietnamese Practitioners’ Views,” *2008 IEEE Int. Conf. Glob. Softw. Eng.*, pp. 67–76, Aug. 2008.

[74] D. G. Boyer, M. J. Handel, J. Herbsleb, and C. C. Rd, “Virtual Community Presence Awareness,” *Communications*, no. 3, pp. 1–4.

[75] J. D. Herbsleb, D. J. Paulish, and M. Bass, “Global software development at siemens: experience from nine projects,” *Proceedings. 27th Int. Conf. Softw. Eng. 2005. ICSE 2005.*, pp. 524–533, 2005.

[76] M. K. Nakayama, E. Binotto, and B. S. Pilla, “Trust in Virtual Teams : A Performance Indicator,” *Education*, vol. 210, pp. 105–113, 2006.

[77] K. Dullemond and B. van Gameren, “What Distributed Software Teams Need to Know and When: An Empirical Study,” *2013 IEEE 8th Int. Conf. Glob. Softw. Eng.*, pp. 61–70, Aug. 2013.

[78] T. Dingsøyr and D. Smite, “Managing Knowledge in Global Software Development Projects,” *IT Prof.*, pp. 1–1, 2013.

[79] G. N. Aranda, A. Vizcaíno, A. Cechich, and M. Piattini, “How to Choose Groupware Tools Considering Stakeholders ’ Preferences During Requirements Elicitation ?,” pp. 319–327.

[80] G. N. Aranda, A. Vizcaíno, A. Cechich, and M. Piattini, "A Cognitive Perspective for Choosing Groupware Tools and Elicitation Techniques in Virtual Teams," pp. 1064 – 1074.

[81] S. Fricker, T. Gorschek, and P. Myllyperkiö, "Handshaking Between Software Projects and Stakeholders Using Implementation Proposals," pp. 144 – 159.

[82] M. Geisser and T. Hildenbrand, "A Method for Collaborative Requirements Elicitation and Decision-Supported Requirements Analysis," *Front. A J. Women Stud.*, vol. 219, pp. 108–122.

[83] F. Calefato, "SocialCDE : A Social Awareness Tool for Global Software Teams Categories and Subject Descriptors," pp. 587–590, 2013.

[84] M. Heindl and S. Biffl, "Risk management with enhanced tracing of requirements rationale in highly distributed projects," *Proc. 2006 Int. Work. Glob. Softw. Dev. Pract. - GSD '06*, p. 20, 2006.

[85] A. Välimäki and J. Kääriäinen, "Patterns for Distributed Scrum – A Case Study," *Enterp. Interoperability III*.

[86] A. Borges, S. Soares, S. Meira, H. Tomaz, and C. Costa, "Ontologies Supporting the Distributed Software Development : a Systematic Mapping Study Categories and Subject Descriptors," pp. 153–164, 2013.

[87] B. Al-ani, "Trust in Virtual Teams : Theory and Tools," pp. 301–306, 2013.

[88] D. Edgar, "Lessons Learned: Outsourcing Support Equipment Development," *2006 IEEE Autotestcon*, no. Ll, pp. 80–85, Sep. 2006.

[89] D. Damian, F. Lanubile, and T. Mallardo, "The Role of Asynchronous Discussions in Increasing the Effectiveness of Remote Synchronous Requirements Negotiations," *Communication*, pp. 917–920, 2006.

[90] F. Duarte, C. Pires, C. A. de Souza, J. P. Ros, R. M. M. Leao, E. de Souza e Silva, J. Leite, V. Cortellessa, D. Mosse, and Y. Cai, *Experience with a New Architecture Review Process Using a Globally Distributed Architecture Review Team*. IEEE, 2010, pp. 109–118.

[91] G. ZU, H. TAIRA, K. MAKINO, T. KANO, and S. MATSUMOTO, *The Supporting Technology of Business Document Proofreading based on Intercultural Differences*. IEEE, 2007, pp. 91–98.

[92] B. Meyer, "Design and code reviews in the age of the internet," *Commun. ACM*, vol. 51, no. 9, p. 66, Sep. 2008.

[93] A. Bhadauria, S. Bhattacharjee, C. B. Anandkumar, and S. Puthiyonnan, "Sustaining High Performance in an Offshore Team in Globally Distributed

Development: A Success Story," *2013 IEEE 8th Int. Conf. Glob. Softw. Eng.*, pp. 120–123, Aug. 2013.

[94] J. Edgerly, "Managing the Challenges of Geographically Distributed Work Groups."

[95] I. Alfaro, "The Impact of Multiple Team Memberships in Leader – Member Exchange Relationship (LMX)," *Management*, no. Lmx, pp. 195–198, 1998.

[96] K. Egea, "Managing the managers : Collaborative virtual teams with large staff and student numbers," *Reproduction*, vol. 20, 2002.

[97] J. D. Herbsleb, *Global Software Engineering: The Future of Socio-technical Coordination*. IEEE, 2007, pp. 188–198.

[98] Y. Iwami, "How to Measure Quality of Software Developed by Subcontractors (Short Paper)," *2008 Eighth Int. Conf. Qual. Softw.*, pp. 300–304, Aug. 2008.

[99] H. a. Reijers, M. Song, and B. Jeong, "Analysis of a collaborative workflow process with distributed actors," *Inf. Syst. Front.*, vol. 11, no. 3, pp. 307–322, Jul. 2008.

[100] F. Lanubile, F. Calefato, and C. Ebert, "Group Awareness in Global Software Engineering," *IEEE Softw.*, vol. 30, no. 2, pp. 18–23, Mar. 2013.

[101] S. I. Hashmi, F. Ishikawa, and I. Richardson, "A communication process for global requirements engineering," *Proc. 2013 Int. Conf. Softw. Syst. Process - ICSSP 2013*, p. 136, 2013.

[102] N. Oza, T. Hall, a Rainer, and S. Grey, "Trust in software outsourcing relationships: An empirical investigation of Indian software companies," *Inf. Softw. Technol.*, vol. 48, no. 5, pp. 345–354, May 2006.

[103] T. G. G. L. Wohlin, "A Model for Technology Transfer in Practice," *IEEE Software*, vol. 23, pp. 88–95, Nov-2006.

[104] N. B. Moe and D. Šmite, "Understanding Lacking Trust in Global Software Teams : A Multi-case Study," *Development*, pp. 20–34.

[105] P. Wongthongtham, E. Chang, and T. Dillon, "Multi-site Distributed Software Development : Issues , Solutions , and Challenges," *Development*, pp. 346–359, 2007.

[106] A. Deuter, "Slicing the V-Model -- Reduced Effort, Higher Flexibility," *2013 IEEE 8th Int. Conf. Glob. Softw. Eng.*, pp. 1–10, Aug. 2013.

[107] S. Deshpande, S. Beecham, and I. Richardson, “Using the PMBOK Guide to Frame GSD Coordination Strategies,” *2013 IEEE 8th Int. Conf. Glob. Softw. Eng.*, pp. 188–196, Aug. 2013.

[108] G. H. Hofstede, *Culture’s consequences: international differences in work-related values*. SAGE, 1984, p. 327.

[109] C. B. Halverson, S. A. Tirmizi, and J. Ungerleider, *Effective Multicultural Teams: Theory and Practice*, vol. 3. Dordrecht: Springer Netherlands, 2008, pp. 211–238–238.

[110] K. Grimm, “Software technology in an automotive company: major challenges,” in in *Proceedings of the 25th International Conference on Software Engineering*, Portland, Oregon: IEEE Computer Society, 2003, pp. 498–503 TS – EndNote Tagged Import Format.

[111] J. H. Sharp and S. D. Ryan, “A Preliminary Conceptual Model for Exploring Global Agile Teams,” *Rev. Lit. Arts Am.*, pp. 147–160, 2008.

[112] S. L. Limited, “The Challenges of Virtual Partnerships : Critical Success Factors in the Formation of Inter-Organisational Teams Violina Ratcheva and Shailendra Vyakarnam 2 . Strategic Motivations for the Study,” *Business*, pp. 99–116, 2001.

[113] D. Smite, “Global Software Development Improvement,” University of Latvia, 2007.

[114] C. K. Shrum, *CMMI for Development®: Guidelines for Process Integration and Product Improvement*. .

[115] “ISO/IEC 12207:2008: Systems and software engineering -- Software life cycle processes.”

[116] Project Management Institute, *A Guide to the Project Management Body of Knowledge: PMBoK Guide*. 2008.

[117] Thi Thanh Loan Nguyen, “The creation of Uni-REPM A universal model for assessing requirements engineering process maturity.” .

[118] T. Dingsøyr, “An evaluation of Research on Experience Factory.” In [AM00, pp. 55–66, 2000.

[119] M. Broy, “Challenges in automotive software engineering,” in in *Proceedings of the 28th international conference on Software engineering*, Shanghai, China: ACM, 2006, pp. 33–42 TS – EndNote Tagged Import Format.

[120] M. B. C. K. Shrum, *CMMI Guidlines for Process Integration and Product Improvement*. Addison-Wesley Longman Publishing Co., Inc. TS - EndNote Tagged Import Format, 2003.

[121] T. Gorschek, P. Garre, S. Larsson, and C. Wohlin, "A Model for Technology Transfer in Practice," *IEEE Softw.*, vol. 23, no. 6, pp. 88–95, Nov. 2006.

[122] B. C. S. Kitchenham, "Guidelines for performing Systematic Literature Reviews in Software Engineering," *EBSE 2007-001*. 2007.

[123] S. U. K. N. Ahmad, "Critical Success Factors for Offshore Software Development Outsourcing Vendors: A Systematic Literature Review," *Glob. Softw. Eng. Int. Conf.*, vol. 0, pp. 207–216, Jul. 2009.

[124] C. Robson, *Real world research: a resource for social scientists and practitioner*. Wiley-Blackwell, 2002, p. 599.

[125] L. MATHIASSEN, *Collaborative Practice Research*, vol. 14, no. 1. 2002.

[126] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, "Experimentation in software engineering: an introduction," Jan. 2000.

[127] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," pp. 68–77, Jun. 2008.

[128] A. Bryman and E. Bell, *Business Research Methods*. Oxford University Press, 2007, p. 786.

[129] C. Robson, *Real World Research: A Resource for Social Scientists and Practitioner-Researchers*. Wiley, 2002, p. 599.

[130] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empir. Softw. Eng.*, vol. 14, no. 2, pp. 131–164, Dec. 2008.

[131] M. R. Louis and J. M. Bartunek, "Insider/Outsider Research Teams: Collaboration Across Diverse Perspectives," *J. Manag. Inq.*, vol. 1, no. 2, pp. 101–110, Jun. 1992.

[132] R. K. Yin, *Case Study Research: Design and Methods (Google eBook)*. SAGE, 2009, p. 219.

[133] B. G. Glaser and A. L. Strauss, *The Discovery of Grounded Theory: Strategies for Qualitative Research*. 2009.

[134] J. Sa, E. Maslova, and B. Bs, "A Unified Process Support Framework for Global Software Development Faculty of Computing , Engineering and Mathematical Sciences University of the West of England," *Sci. York*.

[135] M. T. Lane and P. J. Agerfalk, *Experiences in Global Software Development - A Framework-Based Analysis of Distributed Product Development Projects*. IEEE, 2009, pp. 244–248.

[136] F. Salger, G. Engels, and A. Hofmann, *Assessments in global software development*, vol. 2. New York, New York, USA: ACM Press, 2010, p. 29.

[137] W. Aalst, J. Mylopoulos, M. Rosemann, M. J. Shaw, C. Szyperski, F. Bomarius, M. Oivo, P. Jaring, P. Abrahamsson, E. Hossain, M. Ali Babar, and J. Verner, *Product-Focused Software Process Improvement*, vol. 32. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 126–140–140.

[138] S. Islam and S. H. Houmb, "Towards a Framework for Offshore Outsource Software Development Risk Management Model," *J. Softw.*, vol. 6, no. 1, pp. 38–47, Jan. 2011.

[139] M. Y. V. William Aspray, Frank Mayadas, "Globalization and Offshoring of Software A Report of the ACM Job Migration Task Force."

[140] M. Ali-Babar, "The application of knowledge-sharing workspace paradigm for software architecture processes," *Proc. 3rd Int. Work. Shar. reusing Archit. Knowl. - SHARK '08*, p. 45, 2008.

[141] N. N. Huda N., Tepandi, J., Deo, P. S., "Key barriers for global software product development organizations," *Manag. Eng. Technol. 2009. PICMET 2009. Portl. Int. Conf.*, pp. 1081–1087, 2009.

[142] A. L. N. Toval, "Risks and Safeguards for the Requirements Engineering Process in Global Software Development," in in *Proceedings of the 2009 Fourth IEEE International Conference on Global Software Engineering*, IEEE Computer Society, 2009, pp. 394–399 TS – EndNote Tagged Import Format.

[143] M. Ivarsson Tony, "Technology transfer decision support in requirements engineering research: a systematic review of REj," *Requir. Eng.*, vol. 14, no. 3, pp. 155–175, Jul. 2009.

[144] M. Ivarsson Tony, "A method for evaluating rigor and industrial relevance of technology evaluations," *Empir. Softw. Eng.*, pp. 1–31, 2010.

[145] I. Pettersson M., Gorschek, T., Öhman, P., F., "A practitioner's guide to light weight software process assessment and improvement planning," *Agil. Prod. Line Eng.*, vol. 81, no. 6, pp. 972–995, 2008.

[146] R. A. J. W. W. T. S.-E. T. I. Format, Ed., *Applied multivariate statistical analysis*. Prentice-Hall, Inc., 1988.

[147] J. C. Carver, R. P. Kendall, S. E. Squires, and D. E. Post, "Software Development Environments for Scientific and Engineering Software: A Series of Case Studies," *29th Int. Conf. Softw. Eng.*, pp. 550–559, May 2007.

[148] S. Fricker and P. Grünbacher, "Negotiation Constellations – Method Selection," pp. 37–51.

[149] S. Berner, R. Weber, and R. K. Keller, "Observations and lessons learned from automated testing," *Proceedings. 27th Int. Conf. Softw. Eng. 2005. ICSE 2005.*, pp. 571–579, 2005.

[150] R. E. Cordes, P. D, and T. M. Spine, "Transcending Organizational Boundaries : Virtual Team Approach in UI Guideline Development," *Development*, pp. 3–4, 2007.

[151] B. Hogan, "Lessons Learned from an eXtremely Distributed Project," *Agil. 2006*, pp. 321–326, 2006.

[152] J. Xu and D. Lippert, "Lesson Learned in Managing IT Departments," *PICMET '07 - 2007 Portl. Int. Conf. Manag. Eng. Technol.*, pp. 2107–2115, Aug. 2007.

[153] C. Seybold and R. K. Keller, "Aligning Software Maintenance to the Offshore Reality," *2008 12th Eur. Conf. Softw. Maint. Reengineering*, pp. 33–42, Apr. 2008.

[154] K. Sureshchandra and J. Shrinivasavadhani, "Adopting Agile in Distributed Development," *2008 IEEE Int. Conf. Glob. Softw. Eng.*, pp. 217–221, Aug. 2008.

[155] P. Cichocki and A. Maccari, "Empirical Analysis of a Distributed Software Development Project," *Ifip Int. Fed. Inf. Process.*, pp. 169–181, 2008.

[156] P. Planning, "Empirical Laws and Rules of Thumb Applying Software Rules of Thumb."

[157] M. He, M. Li, Q. Wang, Y. Yang, and K. Ye, "An Investigation of Software Development Productivity in China *," *Program*, no. 60573082, pp. 381 – 394, 2008.

[158] S. Hole and N. B. Moe, "A Case Study of Coordination in Distributed Agile Software Development," *Work*, pp. 189–200.

[159] N. Jain, "Naresh Jain," *Structure*, pp. 4–10, 2006.

[160] R. Heckman, K. Crowston, and N. Misiolek, "A Structurational Perspective on Leadership in Virtual Teams 1," *Leadership*, vol. 236, pp. 151–168.

[161] S. Setamanit and D. Raffo, "Identifying Key Success Factors for Globally Distributed Software Development Project Using Simulation : A Case Study," pp. 320–332, 2008.

[162] D. Šmite and J. Borzovs, "A Framework for Overcoming Supplier Related," *Context*, pp. 50 – 61, 2006.

[163] E. Lindqvist, B. Lundell, and B. Lings, *Distributed development in an intra-national, intra-organisational context*. New York, New York, USA: ACM Press, 2006, p. 80.

[164] M. Geisser and T. Hildenbrand, "A Method for Collaborative Requirements Elicitation and Decision-Supported Requirements Analysis," *International Federation for Information Processing Digital Library*, vol. 219, no. 1. .

[165] M. Heindl and S. Biffl, *Risk management with enhanced tracing of requirements rationale in highly distributed projects*. New York, New York, USA: ACM Press, 2006, p. 20.

[166] T. N. Nguyen, "Component-Based Software Update Process in Collaborative Software Development," *2008 15th Asia-Pacific Softw. Eng. Conf.*, pp. 437–444, 2008.

[167] P. Paulish Roman, Kuhn, Wolfgang, Daniel J., "Product Line Architectures for Global Software Development," in in *Software Product Lines*, vol. 3154, R. L. Nord, Ed. Springer Berlin / Heidelberg, 2004, pp. 144–146 TS – RIS.

[168] D. Edgar, *Lessons Learned: Outsourcing Support Equipment Development*. IEEE, 2006, pp. 80–85.

[169] F. B. Bass, "Managing variability in software architectures," in in *Proceedings of the 2001 symposium on Software reusability: putting software reuse in context*, Toronto, Ontario, Canada: ACM, 2001, pp. 126–132 TS – EndNote Tagged Import Format.

[170] J. B. F. G. K. H. O. Pohl, "Variability Issues in Software Product Lines," in in *Revised Papers from the 4th International Workshop on Software Product-Family Engineering*, Springer-Verlag, 2002, pp. 13–21 TS – EndNote Tagged Import Format.

[171] J. Bosch, "Software Variability Management," in in *Proceedings of the 26th International Conference on Software Engineering*, IEEE Computer Society, 2004, pp. 720–721 TS – EndNote Tagged Import Format.

[172] K. P. van der L. Metzger, "Software Product Line Variability Management," in in *Proceedings of the 10th International on Software Product Line Conference*, IEEE Computer Society, 2006, p. 219 TS – EndNote Tagged Import Format.

[173] M. S. Bosch, "Issues Concerning Variability in Software Product Lines," in in *Proceedings of the International Workshop on Software Architectures for Product Families*, Springer-Verlag, 2000, pp. 146–157 TS – EndNote Tagged Import Format.

[174] M. W. C. L. L. M. M. Svahnberg, "A Quality-Driven Decision-Support Method for Identifying Software Architecture Candidates," *International Journal of Software Engineering and Knowledge Engineering*, vol. 2003, no. 13(5), pp. 547–573.

[175] D. N. M. A. E. LAWLEY, *Factor Analysis as a Statistical Method*. Butterworths Mathematical Texts.

[176] M. S. van G. Bosch, "A taxonomy of variability realization techniques: Research Articles," *Softw. Pr. Exper.*, vol. 35, no. 8, pp. 705–754, 2005.

[177] R. Koschke, "Survey of Research on Software Clones," *In Proceedings of Dagstuhl Seminar 06301: Duplication, Redundancy, and Similarity in Software, Dagstuhl*, vol. 2006, p. 24pp.

[178] B. S. Baker, "On finding duplication and near-duplication in large software systems," in in *Proceedings of the Second Working Conference on Reverse Engineering*, IEEE Computer Society, 1995, p. 86 TS – EndNote Tagged Import Format.

[179] P. F. K. P. J. B. Angstmann, "Extending the Reflexion Method for Consolidating Software Variants into Product Lines," in in *Proceedings of the 14th Working Conference on Reverse Engineering*, IEEE Computer Society, 2007, pp. 160–169 TS – EndNote Tagged Import Format.

[180] S. D. K. N. H. Schitter, "Variant Comparison - A Technique for Visualizing Software Variants," in in *Proceedings of the 2008 15th Working Conference on Reverse Engineering*, IEEE Computer Society, 2008, pp. 229–233 TS – EndNote Tagged Import Format.

[181] D. Stadel, *Statistical Analysis of Cancer Rates in Florida*. University of Ulm 2007.

[182] M. S. BARTLETT, "The Statistical Conception of Mental Factors," *British Journal of Psychology*, vol. 1954.

[183] M. A. Rothenberger, K. J. Dooley, U. R. Kulkarni, and N. Nada, "Strategies for software reuse: a principal component analysis of reuse practices," *IEEE Trans. Softw. Eng.*, vol. 29, no. 9, pp. 825–837, Sep. 2003.

[184] W. J. Krzanowski, "Principles of multivariate analysis: a user's perspective," Jan. 1988.

[185] C. W. Krueger, "Software reuse," *ACM Comput. Surv.*, vol. 24, no. 2, pp. 131–183, Jun. 1992.

[186] G. H. G. F. Van Loan, *Matrix computations (3rd ed.)*. Johns Hopkins University Press TS - EndNote Tagged Import Format, 1996.

[187] S. Schneider, *Symmetric Eigenvalue Problems*. University of Ulm 2007.

[188] N. Mackey, "Hamilton and Jacobi Meet Again: Quaternions and the Eigenvalue Problem," *SIAM J. Matrix Anal. Appl.*, vol. 16, no. 2, pp. 421–435, 1995.

[189] H. Rutishauser, “Handbook Series Linear Algebra. The Jacobi Method for Real Symmetric Matrices,” *Numerische Mathematik, Springer Verlag*, vol. 1966, no. 9.

[190] L. Guttman, “Some necessary conditions for common-factor analysis,” *Eur. Softw. Process Improv. Conf.*, vol. 19, no. 2, pp. 149–161, Jun. 1954.

[191] H. F. D. K. KAISER, “Analytic determination of common factors,” *American Psychological Reports*, vol. 1959, no. 14, p. 425ff.

[192] R. Cattell, “The Scree Test For The Number Of Factors,” *Multivariate Behav. Res.*, vol. 1, no. 2, pp. 245–276, 1966.

[193] M. Ivarsson Tony, “Technology transfer decision support in requirements engineering research: a systematic review of REj,” *Requir. Eng.*, vol. 14, no. 3, pp. 155–175, Jul. 2009.