TECHNICAL REPORT IN SOFTWARE ENGINEERING

Towards a Behavioral Software Engineering

PER LENBERG

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2016

Towards a Behavioral Software Engineering
PER LENBERG

Technical report 153L
ISSN 1652-8565
Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31-772 1000

Chalmers Reproservice
Göteborg, Sweden 2016

Towards a Behavioral Software Engineering
Technical report in Software Engineering
PER LENBERG
Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

The human aspects of software development have repeatedly been recognized as important. This has, however, not been reflected in the research community, where studies that emphasize human aspects are rare compared to studies with technology or process focus. Software development is a human-centered activity and should be recognized as such. If we overlook the human aspects in software engineering research, we risk producing incomplete results where key factors are not uncovered.

The social sciences have been studying human behavior for over a century and we believe that much of the rich knowledge gained could be used to improve software development. Thus, the overall purpose of our research is to explore the possibility to improve software development effectiveness by using theories, concepts and knowledge developed within the social sciences.

The result of this report is based on five separate studies. In these studies, we present a definition of the behavioral software engineering (BSE) research area, which highlights the study of cognitive, behavioral and social aspects at different levels of software engineering work. In addition, using a systematic literature review and by interviewing practitioners, we have identified several software engineering improvement areas and activities where BSE is considered to be a core part of the solution. Together these results form a solid platform for future BSE research.

For one identified improvement area, organizational change, we combined social science and software engineering research to compile first-order models that predict attitudes towards organizational change. The models were verified using industry data. Our results also revealed that there, for the specific change that we investigated, were statistically significant attitude differences between roles, and that the software developers had a more positive attitude towards the change compared to the line managers.

In addition to continue our research on organizational change, we will, in our future research, strive for a deeper understanding of software development effectiveness by exploring what BSE concepts that affect it. Our ambition is to compile a multi-level model that enables software engineering organizations to maximize their development effectiveness.

Keywords: Software, Engineering, Behavioral Software Engineering, Psychology, Sociology

Acknowledgements

I would like to express my sincere gratitude to my supervisors Prof. Robert Feldt and Ph.D. Lars Göran Wallgren Tengberg for their continuous support of my studies, their patience, motivation and knowledge. I could not have imagined having better mentors.

I also acknowledge the support of Saab AB, Swedish Armed Forces, Swedish Defence Materiel Administration and Swedish Governmental Agency for Innovation Systems in the project "Aligning Requirements and Verification Practices in Air Traffic Control Systems" (project number 2013-01199).

Thesis

This thesis consists of an extended summary and the following appended papers:

Paper A P. Lenberg, R. Feldt, and L.-G. Wallgren. "Towards a Behavioral Software Engineering". *Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2014*. ACM, 2014, pp. 48–55. ISBN: 978-1-4503-2860-9. DOI: 10.1145/2593702.2593711

Paper B P. Lenberg, R. Feldt, and L. G. Wallgren. "Human factors related challenges in software engineering - An industrial perspective". *Proceedings of the 8th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2015*. 2015, pp. 43–49. DOI: 10.1109/CHASE.2015.13

Paper C P. Lenberg, R. Feldt, and L.-G. Wallgren. Behavioral software engineering: A definition and systematic literature review. *Journal of Systems and Software* **107** (2015), 15–37. DOI: 10.1016/j.jss.2015.04.084

Paper D P. Lenberg, L.-G. Wallgren, and R. Feldt. Software Engineers' Attitudes Towards Organizational Change - an Industrial Case Study. *In submission to a journal* (2016). arXiv:1601.05837

Paper E P. Lenberg et al. "An Initial Analysis of Differences in Software Engineers' Attitudes Towards Organizational Change". *Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2016*. IEEE. 2016. DOI: http://dx.doi.org/10.1145/2897586.2897592

Contents

Part I
Extended Summary

1 Introduction

For years, reports of industrial software development efforts have collectively painted a rather bleak picture indicating that many projects fail to reach their intended aims [41, 2, 23, 25]. For example, a global survey of over 400 business and IT executives in 2013 showed that although the companies recognized that effective software development is crucial to achieving competitive advantage, only 25% consider themselves to be effective today [2]. Even if we acknowledge that such type of statistics have received heavy critique [26, 44, 34], at least, it indicates that there is room for improvement [23] and that the industry calls for more effective software development.

In order to increase the effectiveness, researchers have sought to identify factors that would positively affect the software development projects success-rate [6, 4, 64, 71, 19]. Much of this research and practice have been technological or process-related, while studies that emphasize human-oriented aspects, for example organizational, social or psychological factors, are rare [43, 60]. Even if the introduction of agile methods has somewhat highlighted the importance of people, teams and collaboration [40, 12, 61], these aspects cannot be consider to be in the research mainstream.

Researchers acknowledge that software development is a human-centered activity [71, 37, 38, 3, 29, 17] and, as such, highly affected and controlled by the feelings, attitudes and behaviors of the humans involved. Thus, to gain knowledge of how to improve software development effectiveness we need not only to understand the technical and process-related aspect, but also in detail grasp what affects the behaviors of the humans involved. If we overlook these aspects, we risk building theories, methods, processes and models in which possible key factors are missing. As an example, the human reluctance to change [58] might be more important to consider in a software process improvement effort than exactly which process change is made or which tool is introduced. However, most research on software process improvement focus on the actual change rather than the people that will have to change their behavior [72].

Software development is composed of a multitude of diverse activities, and to streamline these are beyond the scope of a single research discipline and therefore require an interdisciplinary approach. The social sciences, for example psychology or sociology, have been studying human behavior for over a century [65]. There are, nonetheless, indications that software engineering researchers seldom use theories developed and proven within these sciences. For example, agile development emphasizes team aspects and collaboration, which clearly is associated with sociology and social psychology [18]. However, a brief analysis of the publications included in four literature review studies [42, 9, 68, 67] related to agile development reveal that none used sociology or social psychology theories.

We strongly believe that the rich knowledge gained by the social sciences could be used to improve software engineering research both directly, through the use of existing theories and concept and, also, indirectly through research design and methodology. Thus, the

overall purpose of this research is to *explore the possibility to improve software engineering effectiveness by using theories, concepts and knowledge developed and proven within the social sciences*. We do this by first identifying activities in software development organizations where human factors have considerable influence, and then determine whether there exist any applicable social science theories or concepts that could provide clues for how to improve these activities.

Our field of research is to be considered young and we need to build-up our knowledge to make informed decisions and maximize future efforts. Instead of directing research questions towards a specific phenomena or problem, we have therefore chosen a rather broad and open strategy with the purpose of gaining familiarity and acquire new insight that will help us to develop relevant hypothesis. We acknowledge that our approach has potential drawbacks and that conducting research without a distinct direction could easily lead to vague and fragmented results that fail to provide rich and profound insights. However, in an immature research area such as ours, where the problems are in a preliminary stage, we argue that a broader understanding of this field of research as a whole can be as beneficial as deep knowledge of a single limited phenomenon.

This thesis is made up of five research papers (A to E). In the next section, we present an overview of these papers. Then we present the research methods that we have used, after which we briefly summarize the results from each paper. Finally, we present a collective discussion and outline our future work.

2 Overview

The content of this report is, as stated in the introduction, based on the result of five separate studies (A to E). Of these studies, four have been published (A, B, C and E) and one is in submission (D). In this current section, we present an overview of these appended studies and provide a narrative in which we explain the underlying logical reasoning that justify our research questions. The studies we have conducted are, to a certain extent, dependent on each other and aligned with the overall purpose of our research. However, since our research is conducted in close collaboration with the industry, our choices of studies are partially controlled by external events. For example, to collect relevant data, our organizational change case studies (D and E), had to be synchronized with an actual change in one of our partner companies. Furthermore, the result of one study motivates, and in some cases also provides input to, the subsequent studies. Therefore, to make sense of the workflow, we need not only to understand the purpose of each study, but we also need to briefly know its result.

In an attempt to delineate the scope of our research, in our first paper (Paper A), we proposed a definition of our research area, which we, inspired by Behavioral Economics [8], named Behavioral Software Engineering (BSE). This definition, together with the result of a pilot systematic literature review, was presented and discussed at the CHASE workshop in June of 2014.

In our second study (Paper B), we (a) proposed an updated BSE definition based on the in-depth knowledge that we have acquired, and (b) presented a more detailed overview of the BSE research conducted so far. In addition to understand *what* have

been studied and *how* the studies have been conducted, we also aimed to identify gaps in current research, identify trends and point to directions for our future research.

Even if the overview presented in paper B was thorough and rich, it provided little information or clues about what areas or activities in software engineering that would gain the most from considering the human aspects. Therefore, to complement the SLR results and to identify possible discrepancies between current BSE research and the actual need, we conducted an industrial case study (Paper C). Results from this study identified organizational change as one of four areas of special industrial interest together with customer relations, one-dimensional solutions and communication.

Thus, to meet the software engineering industry's need identified in paper C, we conducted two industrial case studies related to organizational change (Paper D and Paper E). We chose to focus on attitude towards change, since research has identified attitudes as one of the most critical factors in the change process [63, 59].

3 Method

The research methods of the appended studies are presented in table 3.2. We have chosen to describe them using five sub-features (research type, data type, collection method, analysis method and number of respondents) based on the guidance by Easterbrook et al. [20] and Figgou et al. [30]. We acknowledge the definitions of the sub-features, which are presented in table 3.1, are not crystal clear; however, we think that it adds value as a general overview of our research. As shown, we have used a variety of methods. The choice of research method has been determined based on what type of research question we aimed to answer. In a PhD education, however, the variety adds value in it own, since we consider research design as an important part of the training towards becoming an independent researcher.

Our first two studies, i.e. paper A and B, are identical in terms of research method. We used theoretical reasoning to define a research area concerned with human factors of software engineering, and we used systematic literature review (SLR) to create a common platform for future research by identifying gaps and trends in the current research. The procedures for the SRL, which aimed to create a fair evaluation by using a trustworthy, rigorous, and auditable methodology, were based on the guidelines described by Kitchenham [45], which, in turn, have been derived from practices in medical research and adjusted to suit software engineering.

When conducting the SLR, we used a predefined review protocol to reduce the possibility of researcher bias. The protocol described the review process, which included the following stages (also shown in figure 3.1); (1) analyzing the need for a systematic literature review, (2) selecting data sources, (3) selecting search string, (4) defining research selection criteria, (5) defining research selection process and (6) defining data extraction and synthesis.

Sub-feature	Description
Research Type	Classified the publications as either empirical or as conceptual research. In the former, the results are drawn based on analysis of empirical data, whereas the result of the latter are based on a theoretical reasoning.
Data Type	Specifies how data were collected, and is therefore only applicable to empirical research. Data are classified as either quantitative, qualitative or literature review. Quantitative data are anything that can be expressed as a number or quantified, that may be represented by ordinal, interval or ratio scales and lend themselves to statistical manipulation. Qualitative data, on the other hand, cannot be expressed as a number.
Collection Method	Defines what type of method that was used to collect the data, e.g. interviews, focus groups, questionnaires, observations.
Analysis Method	The data analysis method e.g. systematic literature review (SLR) thematic analysis, grounded theory, regression, analysis of variance (ANOVA), regression and structural equation modeling (SEM).

Table 3.1: Description of sub-features.

Publication	Research Type	Data Type	Collection Method	Analysis Method	No of respondents
Paper A	Conceptual and Empirical	Quantitative	Literature Review	SLR	52 publications
Paper B	Conceptual and Empirical	Quantitative	Literature Review	SLR	250 publications
Paper C	Empirical	Qualitative	Interview	Thematic	9
Paper D	Empirical	Quantitative	Literature Review and Questionnaire	SLR and Regression	57 (quan)
Paper E	Empirical	Quantitative and Qualitative	Questionnaire and Interview	ANOVA and Thematic	50 (quan) 11 (qual)

Table 3.2: Description of research methods per publications using five sub-features, which are presented in table 3.1.

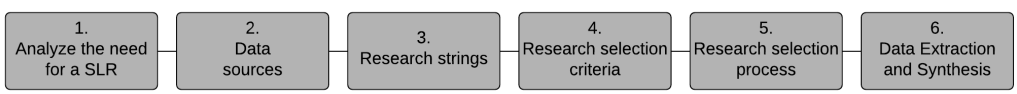

Figure 3.1: Overview of the steps in the systematic literature review. The process was used in paper A, B and D.

Furthermore, the purpose of our next study, i.e. paper C, was to identify the human factors related challenges that the practitioners consider the most important to address.

Since we aimed to gain a deeper understanding of the underlying concepts that influence software engineers' behaviour, we chose a qualitative research method based on interviews and coding of the answers. This type of method can provide more in-depth results than a quantitative approach with the downside of prohibiting statistical analysis of significance.

We chose a semi-structured interview method since we wanted the interviews to be flexible but also to have a certain amount of guidance. The transcribed data was analyzed using an inductive thematic procedure developed by Braun et al. [7] and Hellström [39]. Thematic analysis differs from grounded theory in that it aims to summarize data into descriptive themes, rather than necessarily developing a novel theory to describe the findings [66].

In paper D, we aimed to create, verify and validate a model that predicts software engineers' attitude towards organizational change. In order to create the model, we used the *traditional measurement development procedure*; a method developed for such purposes and proven within social science research [73]. As suggested by these procedures, we first conducted a systematic literature review (SLR) to gain a better understanding of the domain and, also, to identify possible factors or concepts that could affect software engineers' attitudes.

By combining the software engineering domain knowledge, gained through the SLR, with existing organizational psychology change theories, we compiled two first-order models and verified them using industrial data collected at a Swedish software development company undergoing an organizational change. Since we aimed to use statistical analysis, we chose to use a quantitative research design with questionnaires.

Confirmatory factor analysis (CFA) was used to test whether measures of the items in the questionnaire are consistent with our understanding of that item's nature, and, as such, to test whether the data fit our hypothesized models [27]. We have chosen CFA for our analysis since it is a well established and broadly used method in social science and psychological research. Factor analysis is sometimes criticized [28, 22] and several alternatives have been proposed [49, 31, 57, 13]. However, we note that the criticism primarily focuses on exploratory factor analysis (EFA) rather than the CFA that we have employed. Even though these two methods have similarities they are both conceptually and statistically different. Still, in future work, we can consider analyzing our data with one or more of these alternative methods. However, since we tested for normality and used state-of-the-art model fitting assessment procedures we do not consider the choice of CFA as a threat to the validity of our results.

Furthermore, Cronbach's alpha, a statistical measurement calculated from the pairwise correlations between the items, was used as a lower-bound estimate of the internal consistency. Internal consistency is the degree to which every item measures the same construct.

In order to test the proposed models, we used multiple linear regression analysis [55]. The choice of analysis method was based on the guidelines developed by Gefen et al. [33]. According to these guidelines, second generation data analysis techniques, such as partial least squares path analysis and LISREL, require a sample size at least 10 times the number of latent variables in the model. We recognize that this is just a rule of thumb and that there is no general consensus on the appropriate method for determining adequate sample size [74, 75, 76]. The choice of analysis method was by no means obvious, but rather a

border line case and we, therefore, acknowledge that using an alternative analysis method could have been a viable option that we will consider in future work. However, since our data passed the conditions for regression analysis we do not consider our choice as a major threat to the validity.

In the last of the appended studies (Paper E), we aimed to identify differences in attitudes towards organizational change between roles in a software engineering organization and explore what factors that contribute to these differences. We used a mixed research design [14] and thus combined a quantitative and a qualitative approach, where data from the former were used as input to the latter. First, we verified the presumed difference in attitudes using analysis of variance (ANOVA) [1]. Second, to gain an understanding of what factors that contribute to these differences, we conduced semi-structured in-depth interviews with a few software engineers per role. This interview data was analyzed using thematic analysis.

4 Results

In this section we present an overview of the result of the five appended papers.

Paper A The result of this study consists of two main parts. In the first part, we present a definition and a motivation of a new research area that we named Behavioral Software Engineering (BSE). We argue that the software engineering discipline would gain from a clearly defined area of research concerned with realistic notions of human behaviour that emphasizes different units of analysis (UoA) in software development. A major inspiration to BSE is behavioral economics [62, 70] and the importance that this sub-field of the economics discipline has gained in recent years. Thus, we define *Behavioral Software Engineering as the study of behavioral and social aspects of software engineering activities performed by individuals, groups or organizations.*

Work and organizational psychology uses three UoA (individual, group and organization) in order to give structure to the activities it studies [32]. Even though software development is different from many other types of work, it is unlikely to constitute a whole different type of human endeavor. Hence, we argue that these three aspects also form a relevant structure to BSE. The terms organization and group should here be considered in a general sense, i.e. the latter also includes different types of teams and other task-focused groups, while the former also includes more loose connections of multiple individuals such as communities.

In the second part of this study, we present the results from a pilot systematic literature review of the BSE research area that focused on a few selected psychological concepts. In the end, 52 papers were selected for data extraction and analysis. The study confirms the need for a more complete review of BSE. In more detail, the result indicates that existing research on BSE are scattered on several concepts and/or focused on one unit of analysis. It is more rare to find studies that consider both the individual level and the group level, for example. If they exist they often focus heavily on one of the units while covering others in brief.

Paper B The result of this publication is an extension of the result in our first study. First, we propose an update to the definition of the BSE research area that (a) more clearly highlights the cognitive aspects of software engineering and (b) also includes research of the individuals, groups and organizations themselves, i.e. not only research in relation to the software engineering activities. Hence, we redefine *Behavioral Software Engineering as the study of cognitive, behavioral and social aspects of software engineering performed by individuals, groups or organizations.*

Second, in addition to update the BSE definition, we report the result of a systematic literature review that considers an extensive part the BSE research area. Through seeking information in books and publications, and by interviewing experts in the field of organizational psychology and social psychology, we identified 55 psychological concepts that we deemed to relate to the BSE research area and that were used to form the basis of the search strings in the literature review. On average, more than 500 papers were screened per BSE concept and after all the screening and filtering steps, a total of 250 papers were finally included for further analysis.

The result indicates that the BSE research area is growing and considering an increasing number of concepts from psychology and social science. In addition, our review shows that there are gaps in BSE research. Several concepts that are widely considered to be part of organizational and work psychology, have not yet been studied in software engineering. There are also a number of software engineering areas where no BSE research has been performed. Furthermore, the result indicates that the research performed so far have been unbalanced, with a heavy focus on a few BSE concepts on a limited number of SE areas.

Finally, the result reveals that 32 of the publications, approximately one in eight, had a researcher from a social science faculty, and ten publications, one in 25, had researchers from both the software engineering and social sciences faculties.

Paper C This qualitative study shows that (a) the interviewed participants think that the cognitive, behavioral and social aspects should generally be considered more in all software engineering activities, and also that (b) all roles would benefit from an increased knowledge about these human factors. In more detail, the thematic analysis of the interview transcripts resulted in the following four main, descriptive themes.

Customer Relations: The participants stated that software development was special in that it requires continuous cooperation between customer and supplier throughout the development process. One problem related to customer relations was that software deliveries often introduced changes in the customer's organization. The participants thought that the customer employees felt threatened by these changes and that they, therefore, as a psychological defense, developed a negative attitude towards the supplier. In addition, the participants felt that the quality of the customer relation is related to the contract type, where, for example, fixed price contracts decrease the parties' willingness to take responsibility and also induce tension between them, which in the long run can lead to conflicts.

Organizational Change: The interview participants indicated that organizational changes are more frequent in Software Engineering companies compared to other businesses, and that this affects both the software engineers' psychosocial health and their attitude towards change. Many participants also stated that organizational change efforts often fail

because they do not consider all aspects. Typically they only recognize the organizational aspects, while the group and individual aspects are ignored.

In addition, the participants identified a discrepancy in technical knowledge between the software engineers and the management, and claimed that this discrepancy contributes to that the changes efforts often are bottom-up driven, i.e. initiated by the engineers.

One-dimensional Solutions: The participants indicated that their companies sometimes had too narrow a focus and, as a consequence, tried to solve complex development related problems using too simple solutions. An example of such a problem was cost and time estimation of development activities, where the companies sought solutions that only considered the individual developer's perspective, whereas organizational factors such as culture and politics were ignored. Furthermore, the theories regarding motivation and stress were also considered as too limited and narrow. The interviewed managers and one project manger claimed that motivation is fairly well understood at the individual level, but that there is a need for a theory and for practices that also incorporate group aspects.

Communications: Communications was mentioned both as an area of improvement, but also as an important psychological concept in itself. The participants saw verbal and written communication as central when creating a common understanding of the scope of work. Closely related to the scope of work is the requirement specification, which was seen as "pure communication between you [supplier] and the customer, and a major area of problems". In addition, in order to make the requirement specification less equivocal, the participants suggested that it should be examined from a psychological perspective.

Paper D In this case study, we used industrial data collected from a Swedish software development company undergoing an organizational change to examine if the *knowledge* about the intended organizational change outcome, the understanding of the *need for change*, and the feelings of *participation* affect software engineers' *openness to change* and *readiness for change* respectively; two commonly used attitude constructs. The result of two separate multiple regression analysis showed that *openness to change* is predicted by all three concepts, while *readiness for change* is predicted by *need for change* and *participation*. It indicated that software engineers' *knowledge* about the outcome of the change impacts their attitudes more than their understanding of the *participation* (only valid for *openness to change*), and that *participation*, in turn, has more impact compared to their feeling of *need for change* in the process.

Paper E This study confirmed that there were differences in attitudes towards organizational change between roles in a software engineering organization, where the software developers had a more positive attitude towards the change compared to the line managers. In addition, result of in-depth interviews suggested that discrepancies between roles in terms of the norms and values; degree of impact; and experience of past changes contribute to the differences in attitudes.

5 Discussions

As stated in the introduction, the aim of our research was to *explore the possibility to improve software engineering effectiveness by using theories, concept and knowledge developed and proven within the social sciences.* Altogether, our research has confirmed that there is an industrial and an academical need for a deeper understanding of the underlying mechanisms and factors that govern software engineers' behavior. We have identified several software engineering improvement areas and activities where such knowledge is considered to be a core part of the solution. For one of these identified improvement area, organizational change, we have combined social science and software engineering research to compile first-order models that predict attitudes towards a specific organizational change. We have also shown that attitudes towards organizational change differ between software engineering roles. As a whole, the results of the appended studies provide a platform for future behavioral software engineering research.

We initially chose a broad research approach to gain knowledge, insights and understanding of our research area. In an attempt to delineate the scope, in our two first paper (Paper A and B), we proposed a definition of the research area that we, inspired by behavioral economics [8], named Behavioral Software Engineering (BSE). Even if we consider our definition an qualified starting-point for further BSE research, we appreciate the definition will most certainly have to evolve as it is discussed further at conferences and workshops, and as the knowledge in the area refines.

Having defined BSE, we conducted a systematic literature review (SLR) to get an overview of previous research. The SLR revealed that BSE researchers, so far, have been focused on a few concepts, where the most frequently considered were communication, personality and job satisfaction. In addition, few studies considered concepts from more than one unit of analysis (UoA). Of the included 250 SLR publications, 16% covered concepts from more than one UoA and less than 1% included BSE concepts from all three UoAs (individual, group and organization). The need for multi-level BSE research was confirmed in our industrial case study (Paper C). According to the interviewed practitioners, solutions developed to manage problems in software engineering organizations often have a too narrow a focus. For example, when conducting organizational changes, the industry often emphasizes the organizational aspects, whereas the individual and group aspects are more or less ignored.

Furthermore, results from the SLR in Paper B show that researchers from software engineering faculties conduct the vast majority of the studies. A possible consequence of this faculty homogeneity was shown in the attitude study (Paper D), which revealed that standard change models in work- and organizational psychology, such as Kotter [47], were only used in one of the 42 included publications [69] related to agile transition and organizational change.

To complement the SLR results and to identify possible discrepancies between what BSE research that the researchers currently are undertaking and the actual industrial need, we conducted a case study (Paper C) where we interviewed practitioners from six different software engineering organizations. Results from the study identified organizational change as an area of special interest. Even if the problem of organizational change seems to be universal, our study indicates that there are software engineering unique elements

that need to be considered in order for the change efforts to become successful. For example, software development organizations need to mitigate the possible negative effects that the relative high organizational change rate have on the employees. They also need to recognize that many of the changes are initiated by the employees; not the management. The latter was confirmed in our attitude case study (Paper E), which indicated that a discrepancy in knowledge between the software developers and management contributed to this effect.

To meet the need identified in paper C, we conducted two industrial case studies related to organizational change (paper D and Paper E). The result of the first study showed that, for this particular organizational change, software engineers' attitudes can be predicted by *knowledge*, *need for change* and *participation*. The second attitude study (Paper E), which was based on the same data set, investigated differences in attitudes towards organizational change between roles in a software engineering organization and explored what factors that contribute to these differences. The result confirmed that there were statistically significant differences between roles and that the software developers had a more positive attitude towards the change compared to the line managers. Theses results complement existing research and *suggests that employees in software organizations evaluate the planned change in relation to the norms, values and standards of their peer group*, meaning that an employee will have a positive attitude towards a change if its result is likely to make, or has made, it easier for him/her to uphold the peer group's norms and values.

Our results have implications for researchers. First, we believe, as stated above, that future BSE research would benefit from focusing on several units of analysis since the behavior of humans is too complex to be described using only one. The need for such multi-level research is supported researchers in work and organizational psychology [46, 48, 15]. Kozlowski et al. acknowledge state that organizations, teams, and individuals are bound together in a multilevel system and that this often is neglected in research and practice. Crowstone [15] claims that many organizational issues are multi-UoA and thus incompletely captured by single-UoA theories.

Second, we consider BSE research to be interdisciplinary and argue that BSE studies would benefit from becoming more interdisciplinary, which supports previous software engineering researchers conclusions [16]. The social sciences have over one hundred years of experience in the study of behavior [65], and their gained knowledge could be used to improve BSE research. On the other hand, we acknowledge that software development is a highly complex activity and that software engineering researchers' domain knowledge is imperative. But without a broad, serious and systematic consideration of social science results and methods, software engineering researchers risk having to reinvent the wheel.

Third, our studies of organizational change show that we need to better understand how norms and values are formed and maintained in software engineering groups, e.g. agile development teams. Agile development emphasizes the importance of autonomous and self-organizing development teams [11]. Not managed properly, strengthening the teams' autonomy might have undesirable consequences for the organizations. For example, if the teams get too separated from the rest of the organizations, they risk developing a broad and divers set of norms that are not aligned between the teams nor with the organizational culture, which could lead to inter-group conflicts. Thus, for an organization

to maintain a culture that applies to the whole company, knowledge about how group norms arise and develop over time is impaired.

Furthermore, our research also has implications for practitioners. Overall, it shows that the cognitive, behavioral and social aspects should generally be considered more in software engineering activities, and that all roles would benefit from an increased knowledge about human behaviors. In addition, our research shows that many of the problems that practitioners are struggling with require interdisciplinary solutions. However, since such research is quite uncommon, practitioner cannot solely rely on consulting software engineering research. In order to find solution, they must instead gather and compile research result from several academic disciplines.

Our research also provides guidance for organizations conducting organizational changes. Change managers can increase their chances of successfully implementing change initiatives by focusing on increasing software engineers' *knowledge* about the change, their *participation* in the change process and, finally, their understanding of the *need for change*.

Our research has several limitations. We acknowledge that the included publications in the systematic literature review (paper B) do not cover the entire BSE research area. The result of the SLR is directly related to the identified BSE concepts, which only acts as a starting point and do not cover all conceivable concepts. Since we cannot guarantee that some papers have not mistakenly been excluded or missed we have been careful, in our analysis, not to draw any conclusions based on a single paper or a single concept. Rather, the discussions and conclusions are related to the BSE research area as a whole and reflect the overall trends. In the two qualitative case studies, i.e. paper C and paper E, only a relatively small sample of practitioners were interviewed, although the data collected was both rich and detailed. The small sample particularly affects the validity of the quantitative data results. Regarding the two attitude studies, we have made the assumption that there is causality between attitude and behavior, and that a positive attitude towards organizational change will ultimately lead to a successful organizational change. This assumption needs to be verified. Finally, a limitation to our studies is the generalization of our results. All participants in the studies were working in Sweden, although many of them had previously worked in other countries and cultures.

In our future research we will, in the short term, continue to focus our efforts on exploring attitudes towards organizational change. In a longitudinal study, we plan to evaluate if interventions can influence the factors in our attitude models (paper D) and thereby also improve on the employees' attitudes towards the change. We acknowledge that our models cannot be considered complete; rather, they are to be recognized as first-order approximations that capture the most significant effects. Therefore, we also aim to extend the models with other relevant factors.

In addition to continue our research on organizational change, we will strive for a deeper understanding of software development effectiveness by exploring what BSE concepts that affect it. According to Hackman [36], effectiveness is multifaceted and emphasizes both internal (i.e., member satisfaction, team viability) and external (i.e., productivity, performance) factors.

Our ambition is to create a model that maximizes software engineering organizations' effectiveness over time, not merely for a single event or delivery. According to Kozlowski et

al. [48] and Mohammad et al. [56] the temporal concerns are vital, but often neglected in team effectiveness research. We think that there might be different psychological concept in play when it comes to creating an organization that should be efficient for a short period of time, compared to when creating an organization that should be efficient over time.

Research suggests that working groups have a finite level of working capacity or energy that could be burnt out, leaving the group scattered and unwilling to cooperate further [36]. We therefore consider team viability, which refers to members' satisfaction and willingness to continue working together [35], to be an important concept. We acknowledge that more basic research is needed to understand team viability over significant periods of time and to identify factors that affect it [5]. Still, it is plausible to presume that by including team viability as a factor in the definition of effectiveness, we will increase the likelihood that our model maximizes software development performance over time.

Furthermore, we have two prerequisites that confine the design of our future research. First, we think that it is imperative that our models, to such large extent as possible, are verified using industrial data. We want the performance measurements to be founded in actual, empirical data collected from industry, not solely based on software engineers' self-estimates.

Second, in alignment with our definition of BSE and also supported by previous research in work and organizational psychology [48], we intend to explore psychological concepts from several units of analysis. We acknowledge that there is no purpose of its own to include concepts from all three levels into the final model. Instead, we regard our multi-level approach as a relevant starting-point that will guide us in our endeavor of finding a model that provide a more relevant portrait of software engineers' behavior.

For example, on the individual level, there is strong support that *motivation*, *trust* and *self-efficacy* are valuable indicators. Regarding the group level, *group norms* have surfaced as vital concepts in our own research (paper C and E). The relationship between group norms and performance has previously been recognized in social psychology [24, 10], but the concept has, so far, gotten little focus from software engineering researchers. Yet another interesting group-related concept that, lately, has received much attention is *psychological safety* [21], which represents a shared belief that the group is safe for interpersonal risk taking. Even though there exists relatively few concepts related to the organizational UoA, we believe that *organizational support* and *organizational culture* at least indirectly govern software development effectiveness.

The process of compiling and verifying our models will be based on the procedures outlined by Viswanathan [73] and include the four distinct stages shown in figure 5.1. First, we need to identify all potential concepts that might affect software development effectiveness. The list of BSE concept that we compiled in our systematic literature review (paper B) will be used as a starting-point. For practical reasons, it will not be possible to include all identified concepts in our studies, and we therefore need to reduce the number of concepts. In this selection (reduction) process, we will evaluated each concept in relation to five criteria; (a) number of studies in software engineering that address the concept, (b) measurability - how easy it is to measure to concept, (c) the researchers (our) interest or belief in the concept, and (d) relevance and importance according to research in psychology and sociology. Each criteria is given a weigh value between 0.0 and 1.0,

under the condition that the sum of all weight equals to 1.0. The identified concepts are, for all four criteria, rated on a five-point scale. Finally, the importance of each concept is calculated by multiplying respective weight with the rated values and summarizes them, giving each concept a score between 0.0 and 1.0.

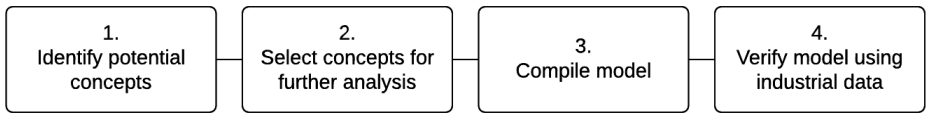

Figure 5.1: Overview of model creation procedure.

When we have completed the selection process, we intend conduct two separate but interdependent studies. The first study will be of a more exploratory nature, as we aim to create an overview of the concepts' importance and the their relationship. Here, we are clearly going for quantity in that we want as many respondents and as many different organizations as possible to participate in the study. To measure the performance we will, in this first study, use software engineer self-estimates. Based on the data collected, we will to compile a model that predicts software development effectiveness. In the final study, our ambition is to verify the model using industrial data.

6 Conclusion

In our research, we present a definition of the behavioral software engineering (BSE) research area, which highlights the study of cognitive, behavioral and social aspects at different levels of software engineering work. We have identified several software engineering improvement areas and activities where BSE is considered to be a core part of the solution. For one of these identified improvement areas, organizational change, we have combined social science and software engineering research to compile a first-order model that predicts attitudes towards a specific organizational change. We have also shown that attitudes towards organizational change differ between software engineering roles.

In addition to continue our research on organizational change, we will, in our future research, strive for a deeper understanding of software development effectiveness by exploring what BSE concepts that affect it. Our ambition is to compile a multi-level model that enables software engineering organizations to maximize their development effectiveness.

References

[1] A. Agresti and M. Kateri. *Categorical data analysis*. Springer, 2011.

[2] M. Albrecht, E. Lesser, and L. Ban. *The software edge*. Tech. rep. IBM Institute for Business Value, 2013.

[3] C. Amrit, M. Daneva, and D. Damian. Human factors in software development: On its underlying theories and the value of learning from related disciplines. A guest editorial introduction to the special issue. *Information and Software Technology* **56**.12 (2014), 1537–1542.

[4] R. D. Banker and C. F. Kemerer. "Factors Affecting Software Maintenance Productivity: an Exploratory Studyl." *ICIS*. 1987, p. 27.

[5] S. T. Bell and B. J. Marentette. Team viability for long-term and ongoing organizational teams. *Organizational Psychology Review* (2011), 2041386611405876.

[6] B. W. Boehm. "Improving software productivity". *Computer*. Citeseer. 1987.

[7] V. Braun and V. Clarke. Using thematic analysis in psychology. *Qualitative research in psychology* **3**.2 (2006), 77–101.

[8] C. F. Camerer and G. Loewenstein. Adam Smith, behavioral economist. *The Journal of Economic Perspectives* **19**.3 (2005), 131–145.

[9] F. K. Chan and J. Y. Thong. Acceptance of agile methodologies: A critical review and conceptual framework. *Decision Support Systems* **46**.4 (2009), 803–814.

[10] J. A. Chatman and F. J. Flynn. The influence of demographic heterogeneity on the emergence and consequences of cooperative norms in work teams. *Academy of Management Journal* **44**.5 (2001), 956–974.

[11] T. Chow and D.-B. Cao. A survey study of critical success factors in agile software projects. *Journal of Systems and Software* **81**.6 (2008), 961–971.

[12] A. Cockburn. *Agile software development: the cooperative game*. Pearson Education, 2006.

[13] G. Conti et al. Bayesian exploratory factor analysis. *Journal of econometrics* **183**.1 (2014), 31–57.

[14] J. W. Creswell. *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications, 2013.

[15] K. Crowston. "Processes as theory in information systems research". *In Proceedings of the IFIP WG8.2 International Working Conference*. 2000, pp. 149–164.

[16] S. S. Cruz et al. "Personality in software engineering: Preliminary findings from a systematic literature review". *Evaluation & Assessment in Software Engineering (EASE 2011), 15th Annual Conference on*. IET. 2011, pp. 1–10.

[17] B. Curtis, H. Krasner, and N. Iscoe. A field study of the software design process for large systems. *Communications of the ACM* **31**.11 (1988), 1268–1287.

[18] T. Dingsoyr, T. Dyba, and P. Abrahamsson. "A preliminary roadmap for empirical research on agile software development". *Agile, 2008. AGILE'08. Conference*. IEEE. 2008, pp. 83–94.

[19] T. Dybå and T. Dingsøyr. Empirical studies of agile software development: A systematic review. *Information and software technology* **50**.9 (2008), 833–859.

[20] S. Easterbrook et al. "Selecting empirical methods for software engineering research". *Guide to advanced empirical software engineering*. Springer, 2008, pp. 285–311.
[21] A. Edmondson. Psychological safety and learning behavior in work teams. *Administrative science quarterly* **44**.2 (1999), 350–383.
[22] C. van der Eijk and J. Rose. Risky business: factor analysis of survey data–assessing the probability of incorrect dimensionalisation. *PloS one* **10**.3 (2015), e0118900.
[23] K. El Emam and A. G. Koru. A replicated survey of IT software project failures. *Software, IEEE* **25**.5 (2008), 84–90.
[24] N. Ellemers, D. De Gilder, and S. A. Haslam. Motivating individuals and groups at work: A social identity perspective on leadership and group performance. *Academy of Management Review* **29**.3 (2004), 459–478.
[25] K. Ellis. *Business Analysis Benchmark*. Tech. rep. 2008.
[26] J. Eveleens and C. Verhoef. The rise and fall of the chaos report figures. *IEEE software* **27**.1 (2010), 30.
[27] L. R. Fabrigar et al. Evaluating the use of exploratory factor analysis in psychological research. *Psychological methods* **4**.3 (1999), 272.
[28] L. R. Fabrigar et al. Evaluating the use of exploratory factor analysis in psychological research. *Psychological methods* **4**.3 (1999), 272.
[29] F. Fagerholm et al. Software Developer Experience: Case Studies in Lean-Agile and Open Source Environments (2015).
[30] L. Figgou and V. Pavlopoulos. *Social Psychology: Research Methods*. 2015, pp. 544–552. ISBN: 978-0-08-097087-5. DOI: http://dx.doi.org/10.1016/B978-0-08-097086-8.24028-2.
[31] D. B. Flora and P. J. Curran. An empirical evaluation of alternative methods of estimation for confirmatory factor analysis with ordinal data. *Psychological methods* **9**.4 (2004), 466.
[32] A. Furnham. *The psychology of behaviour at work*. Psychology Press, 2005.
[33] D. Gefen, D. Straub, and M.-C. Boudreau. Structural equation modeling and regression: Guidelines for research practice. *Communications of the association for information systems* **4**.1 (2000), 7.
[34] R. L. Glass. Failure is looking more like success these days. *IEEE Software* 1 (2002), 104.
[35] R. A. Guzzo and M. W. Dickson. Teams in organizations: Recent research on performance and effectiveness. *Annual review of psychology* **47**.1 (1996), 307–338.
[36] J. R. Hackman. "The design of work teams". *Handbook of organizational behavior*. Ed. by J. Lorsch. Englewood Cliffs, NJ: Prentice-Hall, 1987.
[37] T. Hanne and H. Neu. *Simulating human resources in software development processes*. Fraunhofer-Institut für Techno-und Wirtschaftsmathematik, Fraunhofer (ITWM), 2004.
[38] O. Hazzan and I. Hadar. Why and how can human-related measures support software development processes? *Journal of Systems and Software* **81**.7 (2008), 1248–1252.
[39] C. Hellström. Affecting the Future Chronic Pain and Perceived Agency in a Clinical Setting. *Time & Society* **10**.1 (2001), 77–92.
[40] J. Highsmith. *Agile software development ecosystems*. Addison-Wesley Longman Publishing Co., Inc., 2002.

[41] T. S. G. I. Inc. *2015 Chaos Report*. Tech. rep. 2015.
[42] S. Jalali and C. Wohlin. Global software engineering and agile practices: a systematic review. *Journal of Software: Evolution and Process* **24**.6 (2012), 643–659.
[43] M. John, F. Maurer, and B. Tessem. "Human and social factors of software engineering". *International Conference on Software Engineering: Proceedings of the 27 th international conference on Software engineering*. Vol. 15. 21. 2005, pp. 686–686.
[44] M. Jørgensen and K. Moløkken-Østvold. How large are software cost overruns? A review of the 1994 CHAOS report. *Information and Software Technology* **48**.4 (2006), 297–301.
[45] B. Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele University* **33** (2004), 2004.
[46] K. J. Klein, H. Tosi, and A. A. Cannella. Multilevel theory building: Benefits, barriers, and new developments. *Academy of Management review* **24**.2 (1999), 248–253.
[47] J. P. Kotter and D. S. Cohen. *The heart of change: Real-life stories of how people change their organizations*. Harvard Business Press, 2002.
[48] S. W. J. Kozlowski and B. S. Bell. "Work groups and teams in organizations". *Work groups and teams in organizations*. Vol. 12. Wiley, 2013, pp. 412–469.
[49] S.-Y. Lee. A Bayesian approach to confirmatory factor analysis. *Psychometrika* **46**.2 (1981), 153–160.
[50] P. Lenberg, R. Feldt, and L. G. Wallgren. "Human factors related challenges in software engineering - An industrial perspective". *Proceedings of the 8th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2015*. 2015, pp. 43–49. DOI: 10.1109/CHASE.2015.13.
[51] P. Lenberg, R. Feldt, and L.-G. Wallgren. Behavioral software engineering: A definition and systematic literature review. *Journal of Systems and Software* **107** (2015), 15–37. DOI: 10.1016/j.jss.2015.04.084.
[52] P. Lenberg, R. Feldt, and L.-G. Wallgren. "Towards a Behavioral Software Engineering". *Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2014*. ACM, 2014, pp. 48–55. ISBN: 978-1-4503-2860-9. DOI: 10.1145/2593702.2593711.
[53] P. Lenberg, L.-G. Wallgren, and R. Feldt. Software Engineers' Attitudes Towards Organizational Change - an Industrial Case Study. *In submission to a journal* (2016). arXiv:1601.05837.
[54] P. Lenberg et al. "An Initial Analysis of Differences in Software Engineers' Attitudes Towards Organizational Change". *Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering, CHASE 2016*. IEEE. 2016. DOI: http://dx.doi.org/10.1145/2897586.2897592.
[55] L. S. Meyers, G. Gamst, and A. J. Guarino. *Applied multivariate research: Design and interpretation*. Sage, 2006.
[56] S. Mohammed, K. Hamilton, and A. Lim. The incorporation of time in team research: Past, current, and future. *Team effectiveness in complex organizations: Cross-disciplinary perspectives and approaches* (2009), 321–348.

[57] B. Muthén and T. Asparouhov. Bayesian structural equation modeling: a more flexible representation of substantive theory. *Psychological methods* **17**.3 (2012), 313.
[58] S. Oreg. Resistance to change: developing an individual differences measure. *Journal of applied psychology* **88**.4 (2003), 680.
[59] S. Oreg, M. Vakola, and A. Armenakis. Change recipients' reactions to organizational change A 60-year review of quantitative studies. *The Journal of Applied Behavioral Science* **47**.4 (2011), 461–524.
[60] D. E. Perry, N. A. Staudenmayer, and L. G. Votta. People, organizations, and process improvement. *Software, IEEE* **11**.4 (1994), 36–45.
[61] M. Pikkarainen et al. The impact of agile practices on communication in software development. *Empirical Software Engineering* **13**.3 (2008), 303–337.
[62] M. Rabin. A perspective on psychology and economics. *European economic review* **46**.4 (2002), 657–685.
[63] A. E. Rafferty, N. L. Jimmieson, and A. A. Armenakis. Change Readiness A Multilevel Review. *Journal of Management* **39** (2013), 110–135.
[64] A. Rainer and T. Hall. A quantitative and qualitative analysis of factors affecting software processes. *Journal of Systems and Software* **66**.1 (2003), 7–21.
[65] F. D. Richard, C. F. Bond Jr, and J. J. Stokes-Zoota. One Hundred Years of Social Psychology Quantitatively Described. *Review of General Psychology* **7**.4 (2003), 331.
[66] G. W. Ryan and H. R. Bernard. Data management and analysis methods (2000).
[67] P. Sfetsos and I. Stamelos. "Empirical studies on quality in agile practices: A systematic literature review". *Quality of Information and Communications Technology (QUATIC), 2010 Seventh International Conference on the*. IEEE. 2010, pp. 44–53.
[68] S. V. Shrivastava et al. Distributed agile software development: A review. *arXiv preprint arXiv:1006.1955* (2010).
[69] H. Smits and K. Rilliet. "Agile Experience Report: Transition and Complexity at Cisco Voice Technology Group". *Agile Conference (AGILE), 2011*. IEEE. 2011, pp. 274–278.
[70] J. F. Tomer. What is behavioral economics? *The Journal of Socio-Economics* **36**.3 (2007), 463–479.
[71] A. Trendowicz and J. Münch. Factors Influencing Software Development Productivity—State-of-the-Art and Industrial Experiences. *Advances in computers* **77** (2009), 185–241.
[72] M. Unterkalmsteiner et al. Evaluation and Measurement of Software Process Improvement—A Systematic Literature Review. *Software Engineering, IEEE Transactions on* **38**.2 (2012), 398–424.
[73] M. Viswanathan. *Measurement error and research design*. Sage, 2005.
[74] J. C. Westland. Lower bounds on sample size in structural equation modeling. *Electronic Commerce Research and Applications* **9**.6 (2010), 476–487.
[75] J. C. Westland. *Structural Equation Models*. Springer International Publishing, 2006.
[76] E. J. Wolf et al. Sample size requirements for structural equation models an evaluation of power, bias, and solution propriety. *Educational and psychological measurement* **73**.6 (2013), 913–934.