Towards a Behavioral Software Engineering

Per Lenberg
Div. of Software Engineering
Chalmers University
Gothenburg, Sweden
perle@chalmers.se

Robert Feldt
Dept. of Software Engineering
Blekinge Inst. of Technology
Karlskrona, Sweden
robert.feldt@bth.se

Lars Göran Wallgren
Dept. of Psychology
Gothenburg University
Gothenburg, Sweden
larsgoran.wallgren@psy.gu.se

ABSTRACT

Throughout the history of Software Engineering (SE) it has been repeatedly found that the humans involved, i.e. the engineers and developers in addition to other stakeholders, are a key factor in determining project outcomes and success. However, the amount of research that focuses on human aspects has been limited compared to research with technology or process focus. With increasing maturity of the field, interest in agile methods and a growing dissatisfaction with the continued challenges of developing high-quality software on time, the amount of SE research putting human aspect in primary focus has increased.

In this paper we argue that a synthesized view of the emerging human-focused SE research is needed and can add value through giving focus, direction and help identify gaps. Taking cues from the addition of Behavioral Economics as an important part of the area of Economics we propose the term Behavioral Software Engineering (BSE) as an umbrella concept for research that focus on behavioral and social aspects in the work activities of software engineers. We propose that a model based on three units of analysis can give structure and point to concepts that are important for BSE. To add detail to this model we are conducting a systematic review to map out what is currently known. To exemplify the model and the area we here present the results from a subset of the identified concepts.

Categories and Subject Descriptors

J.4 [**Computer Applications**]: SOCIAL AND BEHAVIORAL SCIENCES—*Psychology, Sociology*; D.2.9 [**Software Engineering**]: Management—*Programming Teams*

General Terms

Human aspects, Management

Keywords

Behavioral Software Engineering, Psychology

To appear in 7th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE 2014), Hyderabad, India, June 2-3, 2014

1. INTRODUCTION

It is sometimes claimed that knowledge-intensive organizations, e.g. IT and Software industry, stand for and symbolize the future of business and working life. The increased interest in knowledge-intensive firms is related to their continued growth and overall importance for society. However, despite this increased interest there are relatively few studies that focus on the professional software developers and the organizational psychology and social aspects around their work [68, 16].

Warfield [69] noted that there is a debate regarding which research methodologies are adequate for this type of research. Researchers based in technology such as computer science often leans to quantitative methodologies while researchers based in behavioral science leans toward qualitative methodologies. As the information systems and technology field (IS/IT) exist in the space between both organizational behavior and technology, a combination of quantitative and qualitative research methodologies are required to advance the knowledge of both disciplines as one.

This has been recognized throughout the years and also in Software Engineering (SE) where research approaches from multiple fields have been utilized. Still, only relatively few studies in SE focus on the humans involved or the social aspects of the groups in which they organize themselves, compared to the number of papers that has a primarily technological perspective.

In this paper we argue that a clearer definition is needed of what this area of SE should be and the many aspects it may need to consider. To identify such aspects, we look to the behavioral sciences and the many concepts that has been studied there. In an ongoing study we are performing a literature review to identify which SE research exist that relates to this proposed area and which concepts they study. In this paper we present initial results to show the potential benefits of a synthesized and more complete view.

After a Background in Section 2 we propose a definition of Behavioral Software Engineering in Section 3. Section 4 then presents the method we have used to create an initial taxonomy and initial results. Finally, Section 5 discusses our findings while Section 6 concludes.

2. BACKGROUND

Since the importance of human and social aspects in software engineering has been recognized since long there is a multitude of related work and topic-specific conferences. In the following section we briefly describe what we have deemed most relevant and that has affected our proposed

definition and model of Behavioral Software Engineering. These main areas are work and organization psychology and behavioral economics. We also review how these topics have been described in different conferences and sub-areas within software engineering.

Conferences on human aspects in SE. A number of workshops and conferences have addressed concerns close to BSE over the years. For example, the CHASE workshops have highlighted two main strands: (a) human and (b) cooperative aspects of SE and emphasized that SE activities typically happen in the context of a group or team [52]. A number of conferences ('Human-centered SE' (HCSE) and 'Conference on Human Factors in Computing Systems' (CHI) among them) have names that allude to this type of research but focus mainly on the human-computer interaction or on usability; we do not focus on this aspect since it is relatively well developed.

Similarly, we acknowledge that there are many connections to the area of socio-technical systems and there have been several proposals for adopting such approaches to the design of software and information systems [5]. Even though there are plenty of and varied results and proposals in this area they are generally more focused on the system to be delivered and the people and organization that will use it rather than the people and organization that develop it; our focus is the opposite.

Psychology of Programming. According to Sajaniemi [48] psychology of programming (PoP) is an interdisciplinary science that dates back to the late 1970s. The aim of PoP, which covers research in (1) computer programmers' cognition, (2) tools and methods for programming related activities and (3) programming education, was originally to make the programmers work more efficiently and to produce better software.

The Psychology of Programming Interest Group was established in the late 1980s [48]. Even though it defines the term programming quite broadly to include any aspects of software development the annual workshop series the group hosts mostly emphasize the individual perspective of programming. The research methods discussed and used most often have been adopted from cognitive psychology [48].

Work and Organizational Psychology. Work and organizational psychology (WOP) has its roots in applied psychology. In the mid 1950s WOP could be defined by two simple epithets - fitting the person to the job and fitting the job to the person [17, 2]. Since then the research area has broadened. A source of confusion is that WOP has a lot of synonyms, e.g. applied psychology, occupational psychology, industrial psychology and work psychology, and that no unified definition exist [17, 2].

Furnham [17] suggests three ways to understand organizational psychology. One way is by analyzing the contributing founding scientific disciplines. It is commonly agreed that WOP is interdisciplinary where the major founding disciplines are psychology, social psychology, sociology and anthropology [17, 2].

Another way of understanding work and organizational psychology is by analyzing different definitions and descriptions. An example of a description by Robertson et al. [2] is "Work psychology is about people's behavior, thoughts and emotion related to their work" (p. xvii). Furnham [17] (p. 2) defines it as follows:

> ... the study of how individuals are recruited, selected and socialized into organizations; how they are rewarded and motivated; how organizations are structured formally and informally into groups, sections and teams; and how leaders emerges and behave. It also examines how the organization influences the thoughts, feelings and behavior of all the employees by the actual, imagined or implied behaviours of others in their organization. Organizational psychology is the study of the individual in the organization, but it is also concerned with small and large groups and the organization as a whole as it impacts on the individual....

The third way of understanding this area of research is by listing its subfields. For example Muchinsky [37] describes that organizational psychologists work in one of six subfields; training and development, employee selection, ergonomics, performance management, work life or organizational development.

Behavioral Economics (BE) is an interdisciplinary science which, using models and knowledge from neighbouring sciences, aims to establish descriptively accurate findings about human cognitive ability and social interaction with implications on economic behaviors and processes. The most influential neighbouring science has been psychology [45]. Some scientists argue that psychological economics is a separate strand of behavioral economics which borrows solely from psychology, especially cognitive psychology [61].

According to Camerer [9] the ideas in behavioral economics are not new. Before the 20th century, when psychology did not exist as a discipline in science, economics giants like Adam Smith and Jeremy Bentham included behavioral aspects in their work. At the beginning of the 20th century the economists were affected by the natural scientist and the psychological aspects were gradually toned down.

In the second half of the 20th century one particular type of economic models had the greatest impact on the emergence of behavioral economics - the models of decision making under uncertainty. These models originally assumed that people are rational and are driven by the principle of expected utility maximization. However, behavioral research systematically documented how people violated such assumptions [9].

The early papers in behavioral economics often had a similar pattern. First, models frequently used by economics were chosen and anomalies were identified. The anomalies were used to create alternative theories and finally a better economic model was proposed based on more realistic views of behavior. The early studies relied heavily on evidence generated by experiments [14]. Specifically, Cramer notes two most influential publications in behavior economics published by Tversky and Kahneman. One demonstrated the anchoring effect [63], a psychological heuristic that influences how people intuitively assess probabilities, and the second paper presented prospect theory [24], a theory of how people make choice under uncertainty.

3. BEHAVIORAL SOFTWARE ENGINEERING

Given the background it is clear that there are a number of research areas that study or touch on different human aspects in relation to the development of software. How-

ever, the relation and/or overlap between them and what they cover is unclear. While research within psychology of programming focuses on the individual and one specific software engineering activity (programming) it puts less focus on other important activities or on the group and organizational levels. Even though both work and organization psychology as well as socio-technical systems highlight all these three main units of analysis (individual, group and organization), the former has no specific focus on software development while the latter focus on the use of the developed system.

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 the different units of analysis in software development. An inspiration is Behavioral Economics and the importance that this sub-field of Economics discipline has gained in recent years. In addition to the scientific value of having a clearly defined area of discourse we also argue that the definition is needed for political reasons. We need a clearly defined area and key concepts in arguments externally, to funding agencies and the society at large, as well as internally, to other SE researchers more focused on technical or process/method aspects of SE work.

Thus we define *Behavioral Software Engineering as the study of behavioral and social aspects of software engineering activities performed by individuals, groups or organizations.*

Behavioral and social aspects primarily include psychology, sociology and anthropology which cover people's behavior, thoughts and emotions but also other areas and concepts.

Groups and organizations on different levels, size and connectivity can be studied, e.g. teams as well as more loosely connected communities and ecosystems of weakly interacting developers.

Even though the actual user and other customer-related stakeholders will directly and indirectly influence the development activities the actual human-computer interaction is already amply covered in many existing research activities and communities and the concept of BSE would risk becoming too diluted if it were to also encompass usability and interaction-related aspects on software systems development.

We argue that the three units of analysis of individual, group and organization are a relevant starting point also for a taxonomy or overview model to give structure to Behavioral Software Engineering. Even though software development is different from many other types of work it is unlikely to constitute a whole different type of human endeavor. Rather, the basic levels at which human aspects are studied in other areas [17, 4, 5] are likely relevant also for software development. At least, such a model can be viewed as a relevant first approximation. Specific psychological and social concepts can then be linked to the units of analysis.

4. TOWARDS A TAXONOMY

In this section we present a pilot study for a systematic literature review [26] in Behavioral Software Engineering that aims to (1) create and validate an overview model and taxonomy of the area and (2) study what previous research has been done and identify gaps. Below, we give a brief overview of the review method used and present the initial results for a subset of relevant psychological concepts.

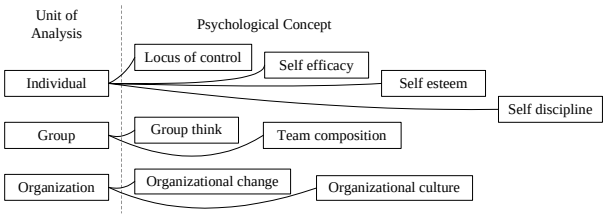

Figure 1: Psychological concepts in pilot study and their relation to main units of analysis

4.1 Review Method

The review method was based on the guidelines described by Kitchenham [26] and included the following six distinct stages; (1) defining research questions, (2) analysing the need for a systematic literature review (SLR), (3) identifying relevant psychological concepts, (4) selecting data sources and search string, (5) defining research selection criteria and process, and (6) data extraction and synthesis. The pilot study focused on two of the stages; (3) identifying relevant psychological concepts and (4) selecting data sources and search string.

Identifying relevant psychological concepts. As a starting point in the process of identifying psychological concepts we used the psychological areas of contribution used by Furnham [17]. These contribution areas are related to the same three units of analysis as could be found in our definition of Behavioral Software Engineering (individual, group and organization).

For each of the contribution areas we identified psychological concepts by seeking information in the literature and by interviewing experts in the field of social psychology. The literature consisted mainly of text books in occupational and organizational psychology. Our intentions were not to compile a complete list, but to include the main concepts that can act as a starting point. After some screening and merging, in total, approximately 60 concepts from psychology and social sciences have been identified.

For the systematic literature review pilot presented here we choose the subset of psychological concepts shown in Figure 1. The concepts were chosen based on two criteria. First, we wanted to include all three units of analysis. Second, we choose concepts that have not previously been systematically reviewed in SE (e.g.the pilot review excluded motivation, previously reviewed in [6]).

Selecting data sources and search string. The review was interdisciplinary, therefore we selected databases likely to cover both more technical as well as psychological research; IEEE Xplore Digital library, ACM Digital library, PsycINFO and Google Scholar.

One search string was generated for each concept by combining the population (different synonyms used to describe software engineers) and the psychological concept with the logical AND operator.

4.2 Results of the Pilot Study

The pilot study selection process resulted in 52 papers selected for data extraction and analysis. Of those, 15 papers focus on the individual: locus of control (3 papers found), self efficacy (9), self esteem (3), self discipline (no papers),

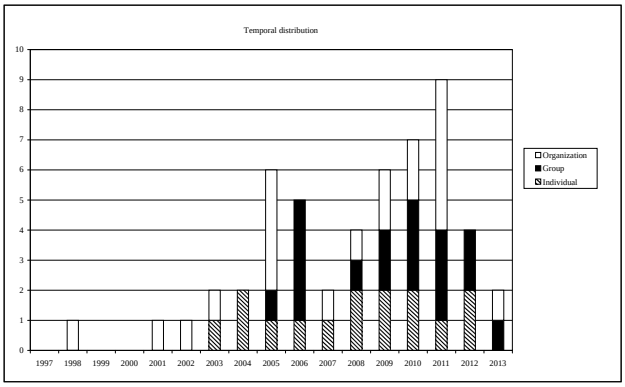

Figure 2: Temporal distribution of included papers per unit of analysis

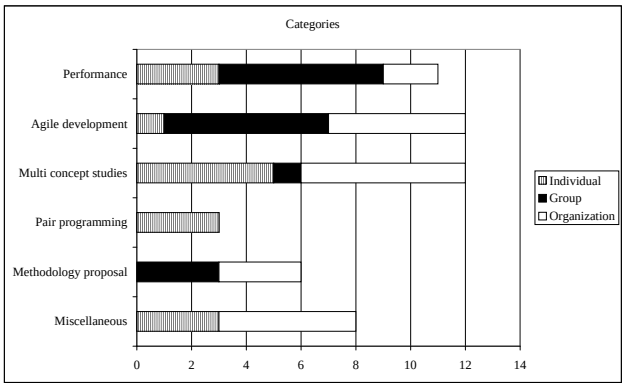

Figure 3: Included papers per category for the psychological concepts of the pilot study

17 papers focus on the group: group think (4), team composition (13), and 20 papers focus on the organization: organizational change (6) and organizational culture (14).

Table 1 presents the sources, the number of found and included papers, their context and research for the psychological concepts. The research methods and contexts are based on definitions previously used by Unterkalmsteiner [65].

Observe that no papers related to self discipline were found and that only one of all of the included papers used an experimental research method. Worth noticing is also the difference in research methods between the units of analysis. The organizational unit includes 10 papers that are solution proposals, whereas the individual unit includes no paper using that method. Of the 52 included papers, six considered more than one unit analysis [62, 28, 41, 7, 57, 18]. Of these, only two addressed all three levels [57, 18].

Figure 2 presents the temporal distribution of the included papers per unit of analysis. Even though we searched for papers from 1997 and onwards the majority (87%) were published 2005 or later, indicating an increasing interest in the psychological concepts [1].

The papers were classified into six software engineering categories; performance, agile development, multi concept studies, pair programming, method proposal and miscellaneous. The number of included papers per category can be seen in figure 3. Below we briefly describe the papers, structured based on the unit of analysis and SE category they investigate.

4.2.1 Individual

Performance: One study [12] indicated no correlation between locus of control and programming performance. Regarding self efficacy, Ramalingam et al. [46] showed that high self efficacy had a positive effect on programming performance, and Ambrose et al. [1] argued that downsizing affected self efficacy negatively, which in turn decreased programming performance.

Agile development: Only one study related to the individual unit of analysis was found. It showed that among managers a high level of self efficacy was positively associated with a high level of agile orientation [50].

[1] Although, we need to consider the overall trend in the number of SE papers to analyze this in more detail.

Multi concept studies: This category included papers in which several psychological concepts were studied in relation to each other, in a software engineering context. A survey study performed by Calisir et al. [8] showed that internal locus positively influenced job satisfaction. Ozer [42] concluded that employees' locus of control did not moderate the leader-member exchange (LMX) [19] to job satisfaction relationship. The relationship between self efficacy and other psychological concepts were examined in three studies. Self efficacy was found to be positively associated to high motivation in open source projects [21] and to organizational commitment [3]. Further, creative self efficacy was negatively correlated to degree of centrality [71].

Pair programming: Three of the studied examined the effects of self esteem on pair programming. The results were coherent and showed that paired students' self esteem did not influence pair compatibility [70, 49, 25].

Miscellaneous: The research and results of the following studies were quite independent and could not be grouped into any specific category. Three of these studies were related to self efficacy. In a cross-cultural survey on students from Germany, France and Greece concludes that the French students had a (relative) drop in self efficacy compared with the other measured variables [20]. A study in India showed that perceived self efficacy is positively related to intentions to reuse software assets [33], and a study that examined the knowledge sharing behavior among software developers showed that knowledge sharing self efficacy has a positive effect on knowledge sharing behavior [62].

4.2.2 Group

Performance: Team composition was identified as a performance factor in five studies [27, 34, 32, 43, 11]. One study stated that personality diversity was among the strongest predictors of project success [43]. Another study concluded that the more degrees of freedom in the composition process the better the result will be, but that there is always a tradeoff between including the best available experts from different sites and the corresponding communication and coordination problems [27]. A systematic mapping survey showed that team composition criteria related to human factors, such as personality and behavior, presented the strongest correlations to project success [11].

Table 1: Searched sources, number of found and included papers, their context and research method for the four psychological concepts in focus for the pilot study

Psychological concept	Num search results				Included papers	Context			Research method						
									Quantitative					None-Quant.	
	Google Scholar	IEEE Xplore	ACM Digital	Psyc-Info		Indu-stry	Non-industry	None	Case study	Action report	Industry report	Sur-vey	Experi-ment	SLR	Solution proposal
Locus of control	3 060	5	48	2	3	2	1	0	3	0	0	2	1	0	0
Self efficacy	7 370	13	161	6	9	5	2	2	8	0	0	8	0	1	0
Self esteem	7 330	5	97	1	3	0	2	0	2	0	0	2	0	1	0
Self discipline	1 980	10	26	0	0	0	0	0	0	0	0	0	0	0	0
Group think	2 570	2	1	1	4	2	0	1	3	0	0	2	0	1	1
Team composition	8 000	18	256	3	13	4	3	3	5	0	1	6	0	3	4
Organizational change	17 200	40	200	17	6	3	0	3	3	0	0	3	0	0	3
Organizational culture	16 600	48	197	6	14	6	0	8	4	0	0	6	0	1	7
Total	64 110	141	986	36	52	22	8	17	28	0	1	29	1	7	15

Agile development: Three papers studied group think in relation to agile development. They indicated an intrinsic vulnerability of an agile team to group think and discussed that it might be a consequence of high cohesiveness [10, 31, 44]. Regarding team composition, two studies stated that team composition was a success factor in agile development [51, 36], and another presents a tool for support managers in composing the teams [28].
Multi concept studies: One study examined the relationship between trust and group composition. The study demonstrated that in culturally homogeneous groups individualism had a negative impact on the level of interpersonal trust, and that individualism in culturally heterogeneous groups had a positive impact on interpersonal trust among members [29].
Method proposal: In total, six of the studies proposed method or guidelines. Of these, three were related to team composition [59, 66, 41].
Miscellaneous: Regarding group think, McAvoy [30] examined project post-mortems before the post-mortem has occurred. The study concluded that hierarchical group think had a detrimental affect on teams' view of project post-mortems.

4.2.3 Organization

Performance: Two studies identified organizational culture as a factor affecting performance [57, 32].
Agile development: Nerur et al. [39] argues that organizations must carefully assess their readiness before treading the path of agility and states that organizations conducive to innovation may embrace agile methods more easily than those founding on bureaucracy and formalization. Four agile development papers were related to organizational culture. One showed that specific organizational cultures correlates with effective use agile methods [56] and three of the papers the relationship were discussed, and obstacles and facilitators were identified [23, 60, 18].
Multi concept studies: Five studies investigated the relationship to software process improvement (SPI). Four of which verified that organizational culture had an influence on SPI [53, 40, 54, 38], while one organizational change study presented methods to increase the applicability of SPI [64]. Briggs and Little [7] concluded that an understanding of the organizational culture could improve the decision-making process.
Method proposal: In total, six of the studies proposed method or guidelines. Of these, two were related to organizational culture [67, 22] and one was related to organizational change [55].
Miscellaneous: Robinson and Sharp [47] explored the interaction between organizational culture and extreme programming (XP). They concluded that although the interaction between organizational culture and XP is complex, XP can thrive in a range of cultures. Three uncategorized studies were related to organizational change. Dirksen et al. [13] states that ideas of change are often implemented as instruments of modernity and demonstrated how this can lead to an artificialization of organizational life. They suggest that fostering people's passion for knowledge and their identification with the organization could be a more productive alternative. A study by Thomas et al. [58] was based on the assumption that organizational change is an ongoing process and occurs in everyday interactions. By studying communication patterns he developed a model that shows how particular communicative practices can lead to generative dialogue, within which resistance plays a facilitative role. In two case studies Middleton [35] concluded that to obtain organizational change there is a need for fast results from low cost actions and that the lean techniques might be a way to achieve that.

5. DISCUSSION

This paper has argued that there is a need to more clearly identify and define an interdisciplinary area of research called Behavioral Software Engineering that should leverage recent results in behavioral science, primarily Psychology but also Sociology and related areas, for a better understanding of the humans involved in software development. We have proposed such a definition and also proposed that a taxonomy of key concepts of this field should be structured based on three main units of analysis. Finally, the paper presented initial results from a systematic literature review for a few concepts in the BSE taxonomy.

The overview model is not complete and we think it will need to evolve over time as a richer picture of BSE is synthesized. We think it will also be key to consider multiple concepts at several units of analysis together. Thus, the model should not be interpreted as static and as defining isolated 'islands' of concerns; to consider the interplay be-

tween multiple concepts at possibly several units of analysis will be important for a better understanding.

The main result from the pilot of the systematic literature review is that it confirms the need for a more complete review of BSE. There are currently many psychological and sociological concepts relevant for Software Engineering for which there is no synthesized overview of the state-of-the-art; a full systematic literature review could both point to where there are solid results but also show gaps and areas where research is missing. There has been work on consolidating results on motivation in software engineering [6] but in other areas for which there is quite a lot of research, e.g. trait-based personality of software developers and engineering [15], there is a lack of summary studies.

The pilot review also makes it clear that there are areas with no or little existing research, e.g. no papers related to self discipline was found. Compared to the many relevant identified concepts there are probably several that have not been considered at all in Software Engineering up to now. We believe a complete systematic literature review could both point to where there are solid results, but also show gaps and areas where research is missing. Also, even if our results indicate a growing trend with more SE studies that consider psychological and sociological aspects, the area is still young and is likely to see continued growth.

In more detail, our results so far indicate 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. We argue that by having a unified view of BSE and by learning from more mature areas such as socio-technical systems and work and organization psychology we can get a richer analytic framework and consider several concepts, levels and units of analysis to better understand software engineering process, methods and tools. Thus, a systematic literature review and better and richer models of what constitutes BSE can both help direct efforts when selecting new studies and enrich the analysis of data elicited in those studies.

The review method of the pilot systematic literature review is not fully described in here. However, since it is based on well-known and accepted guidelines for how to conduct this type of study we do not consider this a major limitation. Furthermore, only a few psychological concepts for each of the three units of analysis were covered by the review and thus no complete analysis of the BSE field has been conducted. Thus our findings should be considered suggestive rather than conclusive.

6. CONCLUSIONS

Even though human aspects on software engineering work have been recognized as important for a long time there is not a clearly identified area and concept for studying it. We have argued that naming and defining such an area is important both scientifically and politically. Thus, we have proposed the concept of Behavioral Software Engineering (BSE) which focuses on more realistic notions of human nature in software development work.

Furthermore, we propose that the three main units of analysis that have been used in other behavioral sciences (individual, group and organization) should be the basis also for a taxonomy of Behavioral Software Engineering. To detail such a taxonomy we are performing a large systematic literature review based on a total of approximately 60 concepts from psychology and social sciences. The initial results, presented here in brief, indicate that there has been an increased interest in Behavioral Software Engineering for the last 5 to 10 years, that different research methods and types of papers are typically used for different units of analysis and that few studies focus on more than one unit of analysis. Also, the identified papers focus on relatively few aspects of software engineering. We think the time is ripe to put a Behavioral Software Engineering much more squarely and clearly in focus for future research to improve software development.

7. ACKNOWLEDGMENTS

We 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).

8. REFERENCES

[1] P. J. Ambrose and A. Chiravuri. A socio-cognitive interpretation of the potential effects of downsizing on software quality performance. *Information Systems Journal*, 20(3):239–265, 2010.

[2] J. Arnold, I. T. Robertson, and C. L. Cooper. *Work psychology: understanding human behaviour in the workplace*. Financial Times/Prentice Hall, 1991.

[3] B. Arya, M. Sharma, and S. Singh. Moderating effect of gender role orientation on the relationship between organizational commitment and self efficacy. *commitment*, 42:3–96178.

[4] E. Babbie. *The practice of social research*. Cengage Learning, 2012.

[5] G. Baxter and I. Sommerville. Socio-technical systems: From design methods to systems engineering. *Interacting with Computers*, 23(1):4–17, 2011.

[6] S. Beecham, N. Baddoo, T. Hall, H. Robinson, and H. Sharp. Motivation in software engineering: A systematic literature review. *Information and Software Technology*, 50(9):860–878, 2008.

[7] C. Briggs and P. Little. Impacts of organizational culture and personality traits on decision-making in technical organizations. *Systems Engineering*, 11(1):15–26, 2008.

[8] F. Calisir, C. Gumussoy, and I. Iskin. Factors affecting intention to quit among it professionals. In *Industrial Engineering and Engineering Management, 2009. IEEM 2009. IEEE International Conference on*, pages 49–53. IEEE, 2009.

[9] C. F. Camerer and G. Loewenstein. Behavioral economics: Past, present, future. *Advances in behavioral economics*, 3, 2004.

[10] S. Coyle, K. Conboy, and T. Acton. Losing the plot: Decision behaviours in agile systems development.

[11] F. Q. da Silva, A. C. C. França, M. Suassuna, L. M. de Sousa Mariz, I. Rossiley, R. C. de Miranda, T. B. Gouveia, C. V. Monteiro, E. Lucena, E. S. Cardozo, et al. Team building criteria in software projects: A mix-method replicated study. *Information and Software Technology*, 2012.

[12] D. P. Darcy and M. Ma. Exploring individual characteristics and programming performance: Implications for programmer selection. In *System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on*, pages 314a–314a. IEEE, 2005.

[13] V. Dirksen, A. Huizing, and B. Smit. A cultural critique of organizational change: getting in touch with reality. In *Proceedings of the 4th International Critical Management Studies Conference*, 2005.

[14] A. Etzioni. Behavioural economics: Next steps. *Journal of Consumer Policy*, 34(3):277–287, 2011.

[15] R. Feldt, L. Angelis, R. Torkar, and M. Samuelsson. Links between the personalities, views and attitudes of software engineers. *Information and Software Technology*, 52(6):611–624, 2010.

[16] R. Feldt, R. Torkar, L. Angelis, and M. Samuelsson. Towards individualized software engineering: empirical studies should collect psychometrics. In *Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering*, pages 49–52. ACM, 2008.

[17] A. Furnham. *The psychology of behaviour at work*. Psychology Press, 2005.

[18] T. J. Gandomani, H. Zulzalil, A. A. A. Ghani, A. B. M. Sultan, and M. Z. Nafchi. Obstacles in moving to agile software development methods; at a glance. *Journal of Computer Science*, 9(5):620, 2013.

[19] C. R. Gerstner and D. V. Day. Meta-analytic review of leader–member exchange theory: Correlates and construct issues. *Journal of applied psychology*, 82(6):827, 1997.

[20] M. N. Giannakos, P. Hubwieser, and A. Ruf. Is self-efficacy in programming decreasing with the level of programming skills? In *Proceedings of the 7th Workshop in Primary and Secondary Computing Education*, pages 16–21. ACM, 2012.

[21] G. Hertel, S. Niedner, and S. Herrmann. Motivation of software developers in open source projects: an internet-based survey of contributors to the linux kernel. *Research policy*, 32(7):1159–1177, 2003.

[22] J. Iivari and M. Huisman. The relationship between organizational culture and the deployment of systems development methodologies. *Mis Quarterly*, 31(1):35–58, 2007.

[23] J. Iivari and N. Iivari. The relationship between organizational culture and the deployment of agile methods. *Information and Software Technology*, 53(5):509–520, 2011.

[24] D. Kahneman and A. Tversky. Prospect theory: An analysis of decision under risk. *Econometrica: Journal of the Econometric Society*, pages 263–291, 1979.

[25] N. Katira, L. Williams, E. Wiebe, C. Miller, S. Balik, and E. Gehringer. On understanding compatibility of student pair programmers. *ACM SIGCSE Bulletin*, 36(1):7–11, 2004.

[26] B. Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33:2004, 2004.

[27] S. Lasser and M. Heiss. Collaboration maturity and the offshoring cost barrier: the tradeoff between flexibility in team composition and cross-site communication effort in geographically distributed development projects. In *Professional Communication Conference, 2005. IPCC 2005. Proceedings. International*, pages 718–728. IEEE, 2005.

[28] S. Licorish, A. Philpott, and S. G. MacDonell. Supporting agile team composition: A prototype tool for identifying personality (in) compatibilities. In *Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering*, pages 66–73. IEEE Computer Society, 2009.

[29] P. B. Lowry, D. Zhang, L. Zhou, and X. Fu. Effects of culture, social presence, and group composition on trust in technology-supported decision-making groups. *Information Systems Journal*, 20(3):297–315, 2010.

[30] J. McAvoy. Evaluating the evaluations: Preconceptions of project post-mortems. *Electronic Journal of Information Systems Evaluation*, 9(2):65–72, 2006.

[31] J. McAvoy and T. Butler. The role of project management in ineffective decision making within agile software development projects. *European Journal of Information Systems*, 18(4):372–383, 2009.

[32] L. McLeod and S. G. MacDonell. Factors that affect software systems development project outcomes: A survey of research. *ACM Computing Surveys (CSUR)*, 43(4):24, 2011.

[33] V. Mellarkod, R. Appan, D. R. Jones, and K. Sherif. A multi-level analysis of factors affecting software developersâĂŹ intention to reuse software assets: An empirical investigation. *Information & Management*, 44(7):613–625, 2007.

[34] C. Melo, D. S. Cruzes, F. Kon, and R. Conradi. Agile team perceptions of productivity factors. In *Agile Conference (AGILE), 2011*, pages 57–66. IEEE, 2011.

[35] P. Middleton. Lean software development: two case studies. *Software Quality Journal*, 9(4):241–252, 2001.

[36] S. C. Misra, V. Kumar, and U. Kumar. Success factors of agile software development. In *Software Engineering Research and Practice*, pages 233–239, 2006.

[37] P. M. Muchinsky. *Psychology applied to work: An introduction to industrial and organizational psychology* . Thomson Brooks/Cole Publishing Co, 1997.

[38] S. D. Muller, P. Kraemmergaard, and L. Mathiassen. Managing cultural variation in software process improvement: A comparison of methods for subculture assessment. *Engineering Management, IEEE Transactions on*, 56(4):584–599, 2009.

[39] S. Nerur, R. Mahapatra, and G. Mangalaraj. Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5):72–78, 2005.

[40] O. Ngwenyama and P. A. Nielsen. Competing values in software process improvement: an assumption analysis of cmm from an organizational culture perspective. *Engineering Management, IEEE Transactions on*, 50(1):100–112, 2003.

[41] M. Omar and S.-L. Syed-Abdullah. Identifying effective software engineering (se) team personality types composition using rough set approach. In *Information Technology (ITSim), 2010 International Symposium in*, volume 3, pages 1499–1503. IEEE, 2010.

[42] M. Ozer. Personal and task-related moderators of leader-member exchange among software developers. *Journal of Applied Psychology*, 93(5):1174, 2008.

[43] V. Pieterse, D. G. Kourie, and I. P. Sonnekus. Software engineering team diversity and performance. In *Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*, pages 180–186. South African Institute for Computer Scientists and Information Technologists, 2006.

[44] M. Pinkowska. It software project management: impact of team cohesiveness on productivity and performance. *Proceedings of the 14th Interdisciplinary Information Management Talks*, 2006.

[45] M. Rabin. A perspective on psychology and economics. *European economic review*, 46(4):657–685, 2002.

[46] V. Ramalingam, D. LaBelle, and S. Wiedenbeck. Self-efficacy and mental models in learning to program. In *ACM SIGCSE Bulletin*, volume 36, pages 171–175. ACM, 2004.

[47] H. Robinson and H. Sharp. Organisational culture and xp: three case studies. In *Agile Conference, 2005. Proceedings*, pages 49–58. IEEE, 2005.

[48] J. Sajaniemi. Psychology of programming: Looking into programmers' heads. *The Problems of Professionals*, page 3, 2008.

[49] N. Salleh, E. Mendes, and J. Grundy. Empirical studies of pair programming for cs/se teaching in higher education: A systematic literature review. *Software Engineering, IEEE Transactions on*, 37(4):509–525, 2011.

[50] T. Seger, O. Hazzan, and R. Bar-Nahor. Agile orientation and psychological needs, self-efficacy, and perceived support: a two job-level comparison. In *Agile, 2008. AGILE'08. Conference*, pages 3–14. IEEE, 2008.

[51] M. Senapathi and A. Srinivasan. Sustained agile usage: a systematic literature review. In *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*, pages 119–124. ACM, 2013.

[52] H. Sharp, R. Prikladnicki, A. Begel, and C. D. Souza. Call for papers – CHASE 2014. IEEE ICSE Workshops, 2014.

[53] C.-C. Shih and S.-J. Huang. Exploring the relationship between organizational culture and software process improvement deployment. *Information & Management*, 47(5):271–281, 2010.

[54] K. V. Siakas. What has culture to do with spi? In *Euromicro Conference, 2002. Proceedings. 28th*, pages 376–381. IEEE, 2002.

[55] M. R. Strens and J. S. Chudge. Using responsibility modeling to match organizational change to changes in the systems development process. *SIGSOFT Softw. Eng. Notes*, 23(3):102–104, May 1998.

[56] D. E. Strode, S. L. Huff, and A. Tretiakov. The impact of organizational culture on agile method use. In *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on*, pages 1–9. IEEE, 2009.

[57] G. P. Sudhakar, A. Farooq, and S. Patnaik. Soft factors affecting the performance of software development teams. *Team Performance Management*, 17(3/4):187–205, 2011.

[58] R. Thomas, L. D. Sargent, and C. Hardy. Managing organizational change: Negotiating meaning and power-resistance relations. *Organization Science*, 22(1):22–41, 2011.

[59] E. Tinelli, S. Colucci, E. Di Sciascio, and F. M. Donini. Knowledge compilation for automated team composition exploiting standard sql. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 1680–1685. ACM, 2012.

[60] C. Tolfo, R. S. Wazlawick, M. G. G. Ferreira, and F. A. Forcellini. Agile methods and organizational culture: Reflections about cultural levels. *Journal of Software Maintenance and Evolution: Research and Practice*, 23(6):423–441, 2011.

[61] J. F. Tomer. What is behavioral economics? *The Journal of Socio-Economics*, 36(3):463–479, 2007.

[62] M.-T. Tsai and N.-C. Cheng. Programmer perceptions of knowledge-sharing behavior under social cognitive theory. *Expert Systems with Applications*, 37(12):8479–8485, 2010.

[63] A. Tversky and D. Kahneman. Judgment under uncertainty: Heuristics and biases. *science*, 185(4157):1124–1131, 1974.

[64] M. Umarji and C. Seaman. Predicting acceptance of software process improvement. In *ACM SIGSOFT Software Engineering Notes*, volume 30, pages 1–6. ACM, 2005.

[65] M. Unterkalmsteiner, T. Gorschek, A. M. Islam, C. K. Cheng, R. B. Permadi, and R. Feldt. Evaluation and measurement of software process improvementâĂŤa systematic literature review. *Software Engineering, IEEE Transactions on*, 38(2):398–424, 2012.

[66] R. van Cann, S. Jansen, and S. Brinkkemper. Team composition in distributed software development. Technical report, Information and Computing Sciences, University of Utrecht.

[67] C. Verma and S. Ali Amin. Significance of healthy organizational culture for superior risk management during software development. In *Developments in E-systems Engineering (DESE), 2010*, pages 182–189. IEEE, 2010.

[68] L. G. Wallgren and J. J. Hanse. Job characteristics, motivators and stress among information technology consultants: A structural equation modeling approach. *International journal of industrial ergonomics*, 37(1):51–59, 2007.

[69] D. WARFIELD. Is/it research: A research methodologies review. *Journal of Theoretical & Applied Information Technology*, 13, 2010.

[70] L. Williams, L. Layman, J. Osborne, and N. Katira. Examining the compatibility of student pair programmers. In *Agile Conference, 2006*, pages 10–pp. IEEE, 2006.

[71] H.-L. Yang and H.-H. Cheng. Creative self-efficacy and its factors: An empirical study of information system analysts and programmers. *Computers in Human Behavior*, 25(2):429–438, 2009.