Contextual Factors Affecting Software Development Practice Efficacy: A Practitioners' Perspective

Diana Kirk[1] and Kelly Blincoe[2]
[1]*School of Computer Science, University of Auckland, Auckland, New Zealand*
[2]*Department of Electrical, Computer and Software Engineering, University of Auckland, Auckland, New Zealand*

Keywords: Software Practices, Software Development Context, Repertory Grid Technique.

Abstract: Practitioners tailor software development methodologies to suit project contexts. Tailoring is often at the level of the project, where individual practices are adapted, often according to tacit knowledge. As the research community does not yet have a good understanding of which contextual factors contribute most to project and practice outcomes, we are not in a position to provide advice and support to practitioners. In this paper, we present the findings from an investigation into the contextual factors that affect practice efficacy. We took a practitioner-centric approach because it is recognised that a person's beliefs and viewpoints affect the way they approach tasks and so are likely to contribute to outcomes. We interviewed twelve participants from three software organisations using the Repertory Grid Technique (RGT). We found that participants varied in the richness of the data provided. Categories 'Task' and 'Team Efficacy' were the most represented, with opposing viewpoints stated for some aspects. The research contribution is that we have exposed a personal element in the way practitioners view context and this has implications for research into practice efficacy.

1 INTRODUCTION

Practitioners adapt and modify methodologies to suit local, project-specific contexts (Kuhrmann and Münch, 2019; MacCormack et al., 2012; Müller et al., 2009; Petersen and Wohlin, 2009; de Azevedo Santos et al., 2011). For example, Diebold et al. studied variations in how Scrum is implemented and found that "only 55% use "pure" Scrum as it has initially been described" (Diebold et al., 2015). Adaptation generally involves modifying specific *practices*.

Many studies have investigated the relationships between project environment and process tailoring (Avison and Pries-Heje, 2007; Clarke and O'Connor, 2012; Klünder et al., 2020; Kruchten, 2013; Masood et al., 2020; Petersen et al., 2021). However, while this research has shown that adaptation of practices can be necessary, there is little knowledge on the context of why changes are needed or why certain practices are selected based on the project's context. In a recent study of practice variations, Masood et al. examined how, when, and why Scrum practices vary (Masood et al., 2020). However, the related literature has not considered the specific contexts in which practices are perceived to be effective. To fulfill this gap, we investigated practitioners perceptions around software practice efficacy,guided by the following research question:

What are the contextual factors that affect software practice efficacy, as perceived by software practitioners?

We used the Repertory Grid Technique (RGT) (Kelly, 2017) as our research methodology because it has been shown to elicit a rich body of information on participant's perceptions while reducing researcher influence and bias (Jankowicz, 2004). We interviewed twelve software practitioners from three organisations.

When asked about the contextual factors that affect practice suitability and efficacy, we expected that responses would focus on aspects of the working environment, for example, whether team members were co-located and the required quality of the product. We found that participants varied in the kinds of information provided, with some discussing factors in the working environment and others focusing on characteristics of the task and team. We also found variation in the complexity of the information given, with some providing extremely rich and complex responses. The main research contribution is that we have exposed a personal element in the way practitioners view context. According to Kelly, this is important because a

Kirk, D. and Blincoe, K.
Contextual Factors Affecting Software Development Practice Efficacy: A Practitioners' Perspective.
DOI: 10.5220/0011076200003176
In *Proceedings of the 17th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2022)*, pages 461-468
ISBN: 978-989-758-568-5; ISSN: 2184-4895

person's beliefs will shape how they approach future events, i.e. will influence outcomes (Kelly, 2017; Tan and Hunter, 2002). An implication is that this must be factored in to future research on practice efficacy.

2 RESEARCH APPROACH

The research question (RQ) is:

RQ: What are the contextual factors that affect software practice efficacy, as perceived by software practitioners?

'Efficacy' relates to the capacity to produce a desired effect. For 'practice efficacy', we mean a combination of effectiveness and efficiency, i.e. produces the desired effect in a timely manner.

The body of literature on software context has resulted in models or frameworks that have been derived from literature studies, feedback from practitioners and researcher experience (see Section 5). Although discussions with practitioners have been open-ended, for example, with semi-structured interviews, these have generally been supported by questions that suggest categories and effectively guide and focus responses. We wanted to explore the topic from the perspective of those implementing the practices. This prompted the decision to base information gathering on one-on-one interviews and to implement RGT to support interviews.

2.1 Repertory Grid Technique (RGT)

RGT is a form of structured interviewing and is the recommended approach for eliciting a rich body of information that is not influenced by the viewpoints of the interviewers (Kelly, 2017). The main components of a grid are the *topic*, *elements*, *constructs*, and *ratings* (Jankowicz, 2004; Tan and Hunter, 2002). The *topic* defines the domain of interest and *elements* are the items of interest within this domain. The topic and elements may be chosen by either the researcher or elicited from interviewees. When the goal is to identify emerging themes, participants select both elements and constructs (full grids), whereas elements are selected by the researcher when the goal is to compare and/or analyse (partial grids). For example, Young et al. applied RGT to investigate the relationship between personalities and roles in an XP team (Young et al., 2005). As the authors wanted to restrict the study to roles within the organisation, the elements (roles) were chosen by the researchers. For our study, the topic is 'Situated Software Practices' and the elements are specific software practices.

Constructs are elicited from participants to capture their perspectives on characteristics of elements (Young et al., 2005). Kelly postulated that a person creates mental constructions based on similarity and difference and so constructs manifest as bipolar descriptions. Poles may represent either *opposites* or *contrasts*. For example, when identifying constructs for the practice 'design sprint', a developer might identify the contrasting terms 'Team collaborated well' and 'Disfunctional team'. The recommended approach for identifying constructs is to ask the participant to consider elements in sets of three (triads) and, for each set, to identify in what way the first two are similar and the third different. In this way, the two poles of a construct emerge.

Ratings enable people to express a viewpoint on the extent to which a characteristic applies and are used when comparing or amalgamating viewpoints. We did not include ratings in our study.

According to Kelly, a person's belief system comprises a hierarchy of concepts, characterised by cause-and-effect ('implies') relationships and with more abstract concepts at the top, 'implied by' lower level concepts. A 'laddering up' process is used to elicit concepts nearer to the top of the hierarchy and involves asking questions such as 'Why is that important to you?'. 'Laddering down' is used to help clarify meaning and involves asking questions such as 'In what way are the bipolar values different?'.

2.2 Study Design

Edwards et al. describe a set of design decisions that must be made to ensure a) grid reliability and validity and b) the suitability of the grid for the research questions (Edwards et al., 2009). These include deciding who chooses the topic and elements, how many elements to include in the study, whether the recommended triadic approach of comparing three elements will be adopted or replaced by a simpler comparison of two elements (diadic) and whether laddering will be implemented.

We chose the topic (situated software practices) and asked participants to select elements (practices) that were of interest to them. According to Edwards et al., this would result in a richer set of constructs (Edwards et al., 2009). We expected that elements might include practices such as 'design review' or 'quality sign-off'. We guided selection by asking participants to consider both successful and unsuccessful practices (Tan and Hunter, 2002).

The recommended approach for identifying constructs is for participants to identify six elements and consider these in sets of three (see Section 2.1). However, a trial study revealed that using six elements would cause us to exceed the time limit we had set for interviews. We replaced the six elements by four (Edwards et al., 2009), yielding four sets of three for comparison. We implemented both laddering up and laddering down. This was expected to yield a set of *threads*, i.e. constructs in a linear, 'causal' relationship. For analysis, our intention was to combine threads for a participant whenever a construct appeared in both threads, as this would provide insight into the participant's main contexts of interest (Pankratz and Basten, 2018). The steps we used for information gathering are summarised as:

1. We chose the topic in advance and asked the participant to discuss in a general way practices regularly carried out as part of their role.
2. We asked the participant to select two 'successful' and two 'unsuccessful' practices that they had recently been involved in. To support the process, we labelled the 'successful' practices A and B, and the 'unsuccessful' practices C and D.
3. We asked the participant to focus on the two 'successful' (A and B) and one 'unsuccessful' (C) practice. We asked 'In what way are A and B similar, and different from C?' We assured the participant there was no 'correct' answer. When happy that we understood, we created a bi-polar construct with the similar pole on the left.
4. We commenced a process of laddering by asking the participant 'Why did this matter?' or 'What was the result of this?' (laddering up) and 'Can you be more specific?' (laddering down). There was no pre-defined ordering for these questions, rather we tried to adapt to the thinking of the participant. As each element was defined, we linked it to the existing nodes.
5. We repeated steps 4-6 for ACD, ABD and BCD.

Interviews were attended by one or both of the authors and audio recorded. The first author transcribed the audio, and checks were carried out on each transcript by the second author. We implemented a number of standard ethics research protocols, for example, relating to permission, anonymity and confidentiality.

2.3 Participants

As our approach is essentially one of building theory from cases, *theoretical* as opposed to *random* sampling is appropriate (Eisenhardt and Graebner, 2007). We approached organisations known to the authors. However, the organisations varied in size and structure (small, stand-alone and group within larger organisation), application area (health, AI platforms and security), and the participants represented a variety of software engineering roles. In Table 1, we overview the participants. We have included only a subset of information for reasons of confidentiality.

Table 1: Study participants.

Role	Org	Role
Product Manager	Small	>5 yrs
Business Analyst	Large	>10 yrs
Product Owner	Large	<5 yrs
Software Architect	Large	<5 yrs
UX Architect	Large	>10 yrs
Developer	Large	<5 yrs
Developer	Large	>5 yrs
Developer	Small	>5 yrs
Software Engineer	Large	? yrs
QA Analyst	Large	<5 yrs
Test Analyst	Large	>5 yrs
Specialist Tester	Large	? yrs

The main contact for each organisation, the *manager*, was a senior manager with authority to make decisions. The names of individuals were provided by the manager. We asked the manager to provide participants covering a range of roles and backgrounds. Talking to employees other than those provided by management was not considered both for formal ethical reasons and because we would view this as a serious breach of trust.

2.4 Data Analysis

As stated above, our intention was to identify the participant's main contexts of interest by combining threads for a participant whenever a construct appeared in both threads (Pankratz and Basten, 2018). However, we observed that participants varied enormously in the richness of, and variation in, the information provided. This was reflected in the shape of the threads for individual triads. In some cases, the resulting thread was linear, i.e. was what we might expect after laddering. In other cases, the information provided was extremely rich, with branches emerging and combining, creating a network of constructs. We could not separate branches into a set of more simple, linear threads because the concepts appeared to be strongly connected in the mind of the participant.

In Figure 1, we illustrate this phenomenon with three examples. In each, we have highlighted in red the nodes at the top of the networks as these represent the participants' high level concerns. We show in blue the nodes that were first-mentioned as these may represent factors perceived as most influential. Network *C* illustrates a simple network. The discussion started at the centre node. Laddering down exposed

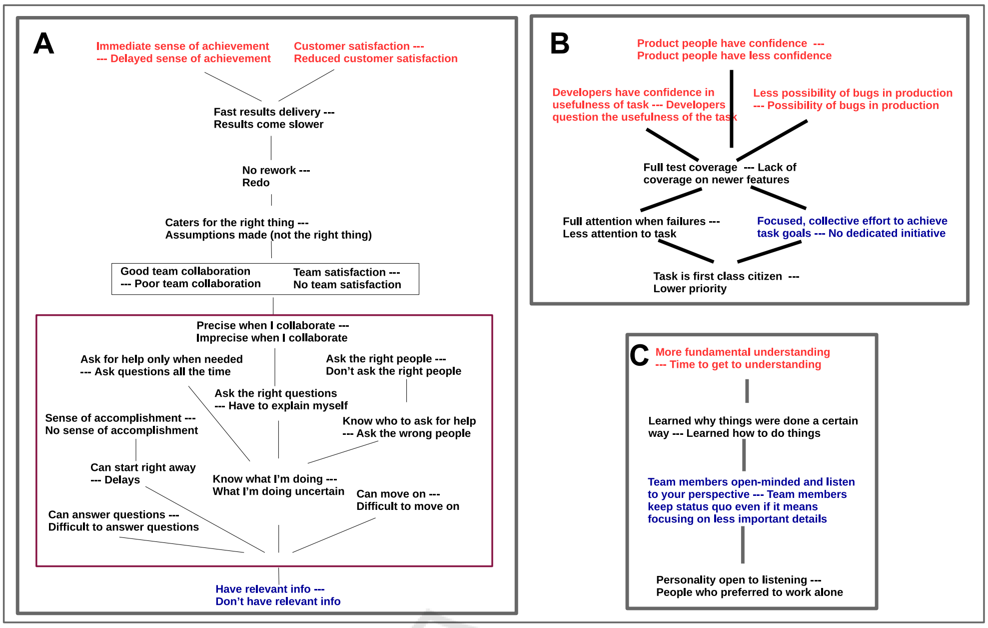

Figure 1: Examples of thread networks, each from a single triad.

a root cause (relating to personalities) and laddering up clarified the outcome (the effort required for understanding). Network *B* shows a more complex situation, where the bottom node branched to two nodes, which led back to a single causal line. Network *A* illustrates a more complex case. The initial construct was quickly and easily identified. However, the laddering prompt often yielded multiple nodes.

An outcome of this observation was that we could not carry out our planned analysis strategy of combining threads for a participant to identify common nodes. We, therefore, focused on the content of the network nodes. Our data set comprised 54 networks containing 249 constructs (nodes). We applied *thematic analysis* to the nodes (Braun and Clarke, 2012). We used a deductive approach based on the commonly stated categories for software development context, i.e. 'Organisation', 'Project', 'Product', 'Team' and 'Individual'. Coding was carried out by the first author and refined in a process of discussion and negotiation with the second author. We analysed the node structures with respect to categories and codes and looked for the most commonly cited categories, as these may represent key factors.

3 FINDINGS

As expected, the category structure altered as coding progressed. We introduced a 'Task' category. We found that factors for 'Team' actually described the team's effectiveness with regard to a specific task and we renamed to 'Team Efficacy'. In Figures 2, 3 and 4, we show the resulting category structure. For example, 'Product' has three sub-categories (Scope, Maturity and Complexity), with codes shown in the rectangles below. We describe the categories in Table 2, with acronyms to aid analysis.

In Table 3, we show the sub-categories that were discussed by more than four participants. Most were in *Task* and *Team Efficacy*. However, not all *codes* were represented. The most represented codes were (number of participants in brackets):

3.1 Task

Focus: Leadership Priority (3). The project manager's focus on sales or quality affected product quality and fitness for purpose. Willingness to follow expert guidance led to a clearer understanding of risk and improved outcomes.

Approach: Communications (4). Face-to-face contact and informal communications were preferred

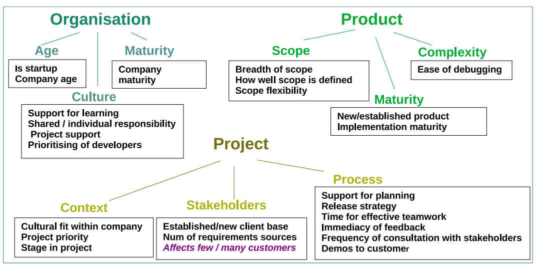

Figure 2: Organisation, Product and Project categories.

Table 2: Context category descriptions.

Category	Sub-Category	Description
Organisation	Maturity	Maturity with respect to software development processes.
(Org)	Age	Number of years of company operations.
	Culture	Software development culture (traditional or agile).
Project	Stakeholders	Characteristics of external stakeholders.
(Proj)	Context	Relationship with organisation, for example, priority, cultural fit.
	Process	Descriptions of process-related factors.
Product	Maturity	Stage in product life cycle (e.g. new development, mature product).
(Prod)	Scope	Characteristics of product requirements.
	Complexity	Complexity of product implementation.
Task	Characteristics	Inherent characteristics of the task.
(Tsk)	Stakehlder Comms.	Approaches to, and issues with, communications with external stakeholders.
	Focus	Factors that affect the focus for task implementation.
	Approach	Descriptions of the way the team approached the task.
Team Efficacy	Cohesiveness	Degree to which team members have common expectations and preferences.
(TE)	Task Capability	Capability of team with respect to the task.
	Preparedness	Factors affecting how well prepared the team members are for task execution.
Individual	Capabilities	Team member experiences and inherent capabilities.
(Ind)	Preferences	Team member preferences.
	Characteristics	Team member personality attributes that might affect task performance.

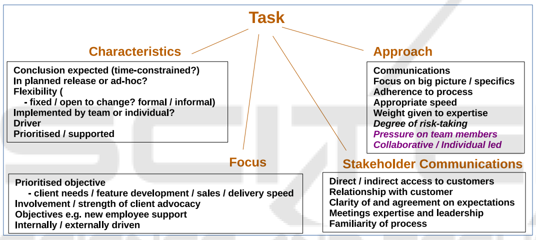

Figure 3: Task category.

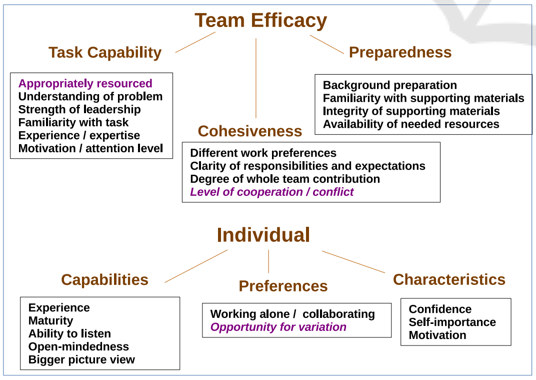

Figure 4: Team categories.

and support a more intuitive understanding of team member understanding. It was important to ensure that all relevant information is communicated to stakeholders.

Characteristics: Flexibility (7). Some saw flexibility as leading to greater success, i.e. open tasks and tasks supported by guidelines rather then rules. Others provided reasons for a more restricted, guided approach.

3.2 Team Efficacy

Task Capability: Understanding of Problem (5). Teams without a clear understanding were unable to plan adequately or make informed debate. An inadequate understanding resulted in build issues and delays and affected collaboration and subsequent product quality and delivery issues.

Task Capability: Experience/Expertise (4). Participants noted the importance of working within their discipline and area of expertise and in familiar roles. The importance of having a subject area expert on the team was cited.

Task Capability: Motivation / Attention Level (3). Issues were reported when teams members were unable to give full attention to the task, were distracted or not in a good mind frame.

Task Capability: Strength of Leadership (3). The

Table 3: Context category representation by participants.

Sub-category	Participants
Proj:Process	06, 07, 08, 09, 11, 12
Tsk:Chars	01, 02, 04, 05, 06, 07, 10, 11
Tsk:Focus	02, 05, 08, 09, 12
Tsk:Approach	01, 02, 03, 04, 05, 07, 08, 09, 12
TE:Tsk Capblty	01, 02, 03, 05, 06, 07, 09, 10, 11, 12
TE:Cohesivenss	01, 02, 03, 04, 06, 07, 12
TE:Preparednss	01, 05, 10, 11, 12

importance of expert involvement in tasks and strong role-specific leadership were discussed.

Cohesiveness: Level of Co-operation / Conflict (6). Good team collaboration and alignment supported sound decision-making and enabled newer team members to gain confidence. Poor collaboration led to ad hoc decision-making, contradictions and the need for context-switching when talking to different team members.

Cohesiveness: Clarity of Expectations (3). Unclear roles and expectations led to ineffective communications, causing delays and frustration.

4 DISCUSSION

The richness and complexity of some networks suggest two avenues for future exploration. The observation that the inter-related nodes in the large rectangle in Figure 1, network *A* tend to describe aspects of teamwork lead us to question whether the many factors that affect team efficacy can be studied in isolation or should be abstracted into a small set of 'team efficacy' measures. The variation in the total number of constructs provided by each participant (between 9 and 51) lead us to hypothesise a possible relationship between the richness of information provided and either the personality of the individual or their role.

In Figures 2, 3 and 4, the sub-category codes shown in non-italic represent factors that were either cited first by participants or elicited as leaf nodes during laddering down. These represent the factors of most concern or those perceived as primitive. Their range and number is large, given the relatively small number of participants. This is to be expected given the variation of participant roles (see Table 1) but many codes do not appear to be role-specific. The implication is that individuals' perceptions of the key contextual factors that affect task outcomes vary enormously. This finding supports Kelly's postulate that people construct events differently, with no two people creating identical systems (Kelly, 2017). It may indicate that an objective assessment of task efficacy is not achievable because identifying which contextual factors matter is not straightforward.

Our analysis of all network nodes in Section 3 indicated that factors relating to *Task* and *Team Efficacy* were discussed by larger numbers of participants. One observation is that the factors found to be most discussed tend to be people-centric. This is unsurprising as we would expect team-related factors to appear as a result of factors lower in the causal network. However, a less expected finding relates to *Task:Characteristics:Flexibility*. Some participants reasoned that more open, creative tasks supported better outcomes and others believed structure to be crucial. This leads us to posit a correlation between a person's preferences and the kind of task they perceive as being effective. As the RGT approach has at its core the belief that a person's beliefs affect the way they behave and ultimately outcomes, there may be a conflict between forming teams with varying personalities to improve outcomes and assigning, for example, a highly creative person to a task that is essentially structured in nature.

Furthermore, if efficacy is influenced by the characteristics of those participating, it may not be possible to 'prove' one practice is better than another. It no longer makes sense to define a practice as a standalone concept, but instead we must consider it in conjunction with the characteristics of the people enacting the practice. For example, practitioners have reported many implementation issues with the Scrum 'Standup Meeeting'. Efficacy may be affected if the team contains introverted or shy developers who find it difficut to share thoughts in a public way (Geekbot, 2020). This observation supports our earlier investigation into the ontology of software projects (Kirk and MacDonell, 2016). The possibility that personalities may play a part in determining 'success' may explain the lack of consistency in findings from studies investigating practice efficacy. Indeed, it is possible that the vigour with which the community has debated the merits of 'plan driven versus agile' may have been based on research affected by personal preference.

We noted in Section 3 that the inclusion of all network nodes resulted in the addition of a small number of (sub-)categories and codes. Additional codes are shown in italic in Figures 2, 3 and 4. New sub-categories were found for *Project* (*Objectives* and *Characteristics*) and *Task* (*Outcomes*). However, there were many instances of nodes describing Task Outcomes. This is unsurprising, given the laddering up process reveals the consequences of earlier nodes. For space considerations, we have not included these in our analysis.

4.1 Threats to Validity

Our analysis is based on interviews with only twelve software practitioners, and we do not know if our results are generalizable or complete. While we did find an initial set of factors, replications of this study in the future can further validate and expand our findings.

A second threat relates to the choice of RGT. We experienced several difficulties in applying this technique, for example, finding that participants struggled with thinking in terms of similarities and differences. These challenges and potential mitigation strategies are fully reported elsewhere (Kirk and Blincoe, 2021). Despite these issues, we did succeed in eliciting a large and rich set of information. However, future work can follow our suggested mitigation strategies, distilled from these challenges, to replicate this study and validate or expand our findings.

Another potential threat is researcher bias. While RGT aims to minimize researcher interference and bias, it is possible that bias was introduced in the analysis. The analysis was primarily done by one researcher. To minimize bias, the factors were iteratively discussed and refined with the research team.

5 RELATED WORK

5.1 Software Development Context

There have been many studies relating tailoring outcomes to situational factors. Some have investigated at an organisational level. Clarke and O'Connor proposed a comprehensive reference framework with eight classifications: *Personnel*, *Requirements*, *Application*, *Technology*, *Organisation*, *Operation*, *Management* and *Business*, divided into 44 factors (Clarke and O'Connor, 2012). Avison and Pries-Heje aimed to support selection of a project-specific methodology (Avison and Pries-Heje, 2007). The methodology is inferred by plotting position on a radar graph with dimensions *Task*, *Knowledge*, *Environment*, *Team*, *Calendar time*, *Stakeholders*, and *Quality and Criticality*. Petersen et al. proposed a comprehensive checklist for practitioners characterising context and researchers aggregating literature studies (Petersen et al., 2021). Facets of the framework are *Organisation*, *Product*, *Stakeholder*, *Development method and technology* and *Business and market*.

Some authors have focused on agile or hybrid approaches. Krutchten proposed a model for guiding the adoption of agile development practices, particularly in environments that are "outside of the agile sweet spot" (Kruchten, 2013). Klünder et al. applied a statistical approach to investigate context for hybrid development methods (Klünder et al., 2020).

Studies vary in the research approach taken and data source used. Avison and Pries-Heje collaborated with five practitioners from an organisation (Avison and Pries-Heje, 2007). The intitial framework was formed from interview notes and the literature. Petersen et al. collaborated with researchers from three institutions to identify factors and evaluated the resulting checklist with eight practitioners. Many studies have sourced data from the literature, for example, (Clarke and O'Connor, 2012). Some frameworks have been based on author experience and expertise and evaluated within organisations (Kruchten, 2013).

5.2 RGT Studies

Two survey papers examined RGT studies in empirical software engineering (Edwards et al., 2009) and Information Systems (IS) (Tan and Hunter, 2002). Edwards et al. note that RGT sits with interpretive research rather than with positivist approaches (Edwards et al., 2009). Both papers concluded that RGT can be a useful technique in the computing domain.

Edwards et al. identified eight studies that apply RGT in the field of Software Engineering (Edwards et al., 2009). Tan and Hunter describe two additional studies in the Information Systems field (Tan and Hunter, 2002). Topics included team member characteristics, project risks, software process improvement and rules in expert systems. In most, the elements and sometimes even the constructs were provided by the researchers. Full grids, where the participants provide both the elements and constructs, as was done in our study, were used in only three of the ten studies.

The studies applying RGT in the computing domain outside of these two survey papers mostly use a full grid (Tofan et al., 2011; Ryan and O'Connor, 2009; Geraldi et al., 2011; Pankratz and Basten, 2018), with only one using a partial grid (Young et al., 2005). Topics included the relationship between personalities and roles in an XP team (Young et al., 2005), situational factors that influence team performance (Ryan and O'Connor, 2009), and managers' perceptions of IS project success mechanisms (Pankratz and Basten, 2018). Two studies were more technical in nature and focused on tacit architectural knowledge (Tofan et al., 2011) and quality attributes in IT projects (Geraldi et al., 2011).

6 SUMMARY

In this paper, we describe a study to explore the contextual factors that affect software development practice efficacy. As we wanted to understand the practitioners' perspective on context, we implemented an approach based on the Repertory Grid Technique (RGT). RGT is a technique aimed at eliciting rich information that is not influenced by the viewpoints of the interviewer. We interviewed twelve practitioners from three sofware organisations. We found that participants varied in the richness of the data provided and named a large range of factors. Aspects of

'Task' and 'Team Efficacy' were the most highly represented, with opposing viewpoints stated for some aspects. We posit a possible correlation between personality and perceptions of task efficacy. The research contribution is that we have exposed a personal aspect in the way in which practitioners view context. This has implications for research into practice efficacy and may affect our ability to advise industry based on objective evidence. In future work, we will explore the possibility of relationships that might exist between named outcomes, the richness of information provided and aspects of personality and roles.

ACKNOWLEDGEMENTS

We thank the research participants for their time and contributions to this study.

REFERENCES

Avison, D. and Pries-Heje, J. (2007). Flexible information systems development: Designing an appropriate methodology for different situations. In *Int'l Conf. Enterprise Information Systems*, pages 212–224. Springer.

Braun, V. and Clarke, V. (2012). Thematic analysis. *American Psychological Association*.

Clarke, P. and O'Connor, R. V. (2012). The situational factors that affect the software development process: Towards a comprehensive reference framework. *Information and Software Technology*, 54:433–447.

de Azevedo Santos, M., de Souza Bermejo, P., de Oliveira, M., and Tonelli, A. (2011). Agile practices: An assessment of perception of value of professionals on the quality criteria in performance of projects. *J. Software Engineering and Applications*, 4:700–709.

Diebold, P., Ostberg, J.-P., Wagner, S., and Zendler, U. (2015). What do Practitioners Vary Using Scrum. In *Proc. Int'l Conf. on Agile Software Development (XP)*, pages 40–51. Springer International Publishing.

Edwards, H. M., McDonald, S., and Young, S. M. (2009). The repertory grid technique: Its place in empirical software engineering research. *Information and Software Technology*, 51:785–798.

Eisenhardt, K. M. and Graebner, M. E. (2007). Theory Building from Cases: Opportunities and Challenges. *Academy of Management Journal*, 50(1):25–32.

Geekbot (2020). Standup meetings: Everything you need to know (standup agenda, purpose, common pitfalls, and more!).

Geraldi, J. G., Kutsch, E., and Turner, N. (2011). Towards a conceptualisation of quality in information technology projects. *International Journal of Project Management*, 29:557–567.

Jankowicz, D. (2004). *The Easy Guide to Repertory Grids*. John Wiley and Sons, Ltd.

Kelly, G. A. (2017). A brief introduction to personal construct theory. *Connstruttivismi*, 4:3–25.

Kirk, D. and Blincoe, K. (2021). Challenges when Applying Repertory Grid Technique for Software Practices. In *Proc. Int'l Conf. on Evaluation and Assessment in Software Engineering*, pages 231–240. ACM.

Kirk, D. and MacDonell, S. G. (2016). An Ontological Analysis of a Proposed Theory for Software Development. *Software Technologies - ICSOFT 2015*, pages 155–171. Springer, Cham.

Klünder, J., Karajic, D., Tell, P., Karras, O., Münkel, C., Münch, J., MacDonell, S. G., Hebig, R., and Kuhrmann, M. (2020). Determining Context Factors for Hybrid Development with Trained Models. In *Proc. Int'l Conf. on Software and System Processes*, pages 61–70. ACM.

Kruchten, P. (2013). Contextualizing agile software development. *Journal of Software: Evolution and Process*, 25(4):351–361.

Kuhrmann, M. and Münch, J. (2019). SPI is Dead, isn't it? Clear the Stage for Continuous Learning! In *Proc. Int'l Conf. on Software and System Processes*, pages 9–13, Montréal, Canada. ACM.

MacCormack, A., Crandall, W., Henderson, P., and Toft, P. (2012). Do you need a new product-development strategy? *Research Technology Management*, 55(1):34–43.

Masood, Z., Hoda, R., and Blincoe, K. (2020). Real world scrum a grounded theory of variations in practice. *IEEE Transactions on Software Engineering*.

Müller, S. D., Kræmmergaard, P., and Mathiassen, L. (2009). Managing Cultural variation in Software Process Improvement: A Comparison of Methods for Subculture Assessment. *IEEE Transactions on Engineering Management*, 56(4):584–599.

Pankratz, O. and Basten, D. (2018). Opening the black box: Managers' perceptions of IS project success mechanisms. *Information and Management*, 55:381–395.

Petersen, K., Carlson, J., Papatheocharous, E., and Wnuk, K. (2021). Context checklist for industrial software engineering research and practice. *Computer Standards & Interfaces*, 78:103541.

Petersen, K. and Wohlin, C. (2009). A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *Journal of Systems and Software*, 82:1479–1490.

Ryan, S. and O'Connor, R. V. (2009). Development of a team measure for tacit knowledge in software development teams. *J. Systems and Software*, 82:229–240.

Tan, F. and Hunter, M. G. (2002). The Repertory Grid Technique: A Method for the Study of Cognition in Information Systems. *MIS Quarterly*, 26(1):39–57.

Tofan, D., Galster, M., and Avgeriou, P. (2011). Capturing Tacit Archirectural Knowledge Using the Repertory Grid Technique. In *Proc. Int'l Conf. on Software Engineering (NIER Track)*, pages 916–919. ACM.

Young, S. M., Edwards, H. M., McDonald, S., and Thompson, J. B. (2005). Personality Characteristics in an XP Team: A Repertory Grid Study. In *Proc Workshop on Human and Social Factors of Software Engineering*, pages 1–7. ACM.