The Discovery of Grounded Theory Practices for Software Engineering Research

Rozilawati Razali, Mashal Kasem Alqudah and Dzulaiha Aryanee Putri Zainal
Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia
rozilawati@ukm.edu.my
alqudahma@ipa.edu.sa
dzulaiha.aryanee@gmail.com

DOI: 10.34190/JBRM.18.2.007

Abstract: Software engineering (SE) research addresses not only technical issues but also human behaviour. SE is considered as an immature discipline because many technical and social issues concerning software development and management have yet to be specified. SE in general is inclined towards quantitative approaches. Nevertheless, qualitative methods are still appropriate for SE research as the methods encourage deep understanding of subject matter. Grounded Theory (GT) is regarded as one of the potential qualitative methods that is applicable to SE research. The method is able to transform less and unknown SE phenomena into cohesive theories through systematic discovery of empirical data from the ground. This paper shares some encounters of using GT in SE research based on the reflection made on several SE research projects covering various phases of software development life cycle. The encounters are then transformed into adaptations and classified as GT practices for SE research, as an effort to inspire the spirit of using GT in SE particularly among novices. The practices embrace aspects concerning formulating research questions, handling preconceptions, utilising software tools, getting access to data and presenting theory and its development process. To illustrate on how the practices were derived, a case study is presented. The proposed GT practices could act as the starting point of adopting GT in SE research. They shall be refined and improved in future to possibly become best practices when more and more experience of using GT in SE are obtained.

Keywords: software engineering, grounded theory, qualitative methods, agile methods

1. Introduction

Software Engineering (SE) encompasses every facet of software systems development and management. It handles different aspects of software life cycle covering both technical as well as human concerns. SE research is likely to use qualitative methods when involving humans, as the methods are able to assist researchers in understanding individuals and the context in which their actions and decisions are made. A variety of qualitative methods are now available for use in SE. Grounded Theory (GT) (Glaser and Strauss, 1967; Corbin and Strauss, 2015; Charmaz, 2014), for instance, appears to be notably applied by researchers when analysing human behaviour and experience.

Originally introduced by Glaser and Strauss (1967), GT is a systematic qualitative method for social science research. It develops the idea that researchers should focus on theoretical development rather than merely theory testing to make the contributions useful to the field of study. GT has been embraced to some degree by a variety of disciplines. It was firstly applied in SE research for investigating the use of Computer-aided Software Engineering (CASE) (Orlikowski, 1993). Some SE research that used GT include topics concerning requirements engineering, distributed development and agile development (Badreddin, 2013).

There are numerous critical SE phenomena that have yet to be understood and GT is seen viable to support SE research in many ways. In addition, not many valid and cohesive theories are available to be tested in SE. For SE discipline to mature, it is important to establish more grounded theories by employing GT in SE research. GT is however not that easy to master by researchers in the science and technology fields, especially amongst the novices. This has resulted GT studies in SE research not being properly carried out and reported (Stol et al., 2016). As the GT adopters themselves, the authors aimed to share some encounters when using GT in SE research from the lens of novices and non-social science researchers. The encounters were gathered through the reflections made on four research projects carried out by the authors to resolve various SE concerns.

Previous studies have reported some perspectives of using GT, as a reference for potential GT adopters in SE research (Adolph et al., 2011; Coleman and O'Connor, 2008; Hoda et al., 2011). This paper augments those studies by proposing several GT practices for SE research. The practices were formulated by transforming the

Reference this paper: Razali, R., Alqudah, M. K., and Zainal, D. A. P., 2020. The Discovery of Grounded Theory Practices for Software Engineering Research. *The Electronic Journal of Business Research Methods*, 18(2), pp. 153-165, available online at www.ejbrm.com

encounters experienced by the authors into adaptations. To articulate the intention, the paper is structured as follows: Section 2 provides some background information about SE and GT; Section 3 explains the four research projects that acted as the basis of the reflections; it also elaborates the methodology used in one of the projects; Section 4 discusses the encounters and adaptations; Section 5 presents the practices through one of the projects as a case study to illustrate how the practices were derived; lastly, Section 6 concludes the paper.

2. Related work

SE is a technical discipline that covers various facets of software development and management. There are fifteen knowledge areas in SE, namely software requirements, software design, software construction, software testing, software maintenance, software configuration management, SE management, SE process, SE models and methods, software quality, SE professional practice, SE economics, computing foundations, mathematical foundations and engineering foundations, as stated in The Guide to Software Engineering Body of Knowledge-SWEBOK (Bourque and Fairley, 2014). In addition, SE also recognises other related disciplines as its auxiliary knowledge areas such as computing engineering, systems engineering, project management, quality management, general management, computer science and mathematics. From these knowledge areas, it can be clearly seen that SE encompasses both technical and social aspects.

As an engineering-oriented discipline, software engineers adopt a systematic, disciplined, quantifiable and organised approach in developing and maintaining software products. Throughout the software development life-cycle (SDLC), software engineers employ specific process models. The process models contain the detailed steps and activities that should be undertaken from the beginning until the end of a software development project. There are many types of process models in SE such as waterfall model, incremental or evolutionary development and reuse-oriented development or component-based SE (Sommerville, 2016). Agile methods (Alliance, 2016) appear as the most current emerging process model, which is a variant of incremental, iterative and evolutionary development.

A survey has shown that 97% respondents have used agile methods in their software development projects, as compared to 84% in the same survey that was conducted 11 years ago (Hoda, Salleh and Grundy, 2018). This indicates that agile methods are currently well accepted and widely adopted by software practitioners in the industry. There are many instances of agile methods such as SCRUM, Extreme Programming (XP), Feature Driven Development (FDD) and Kanban to name a few. Each of them has different practices, however all of them adhere to the basic principles and values outlined in Agile Manifesto (Beck et al., 2001). With regard to agile methods, one of the risky challenges faced by practitioners is how to select the most suitable agile method to be used in a particular software development project (Sharma and Bawa, 2016). The understanding is undeniably important to the practitioners, as the effect of not selecting the right one would jeopardise the success of the project. This phenomenon motivated the authors to discover the critical factors involved in the agile methods selection process. GT was therefore adopted in the research project to build the understanding and thus, the theory for agile methods selection. Section 3 and 5 have the details about the research project.

GT has several significant variants, namely Regenerate GT or Situational Analysis (Clarke, 2005), Critical GT (Kempster and Parry, 2011; 2014; Lee, 2016; Oliver, 2012), Constructivist GT (Charmaz, 2014) and Evolved GT (Corbin and Strauss, 2015). All these variants are the extensions of the first and original GT, called Classic GT (Glaser and Strauss, 1967). There have been extensive discussions on the evolution of different GT variants (Bryant and Charmaz, 2007), GT in a contemporary way (Birks and Mills, 2015) and the differences between Classic GT and Evolved GT (Boychuk, 2004; Heath and Cowley, 2004; Kelle, 2005). The variants still advocate the same vital GT principles, even though they are different slightly in philosophical standpoints. The key principles include theoretical sampling, theoretical sensitivity and theoretical saturation, which are achieved through coding, memoing and continuous comparison (Glaser, 1978; 1992; 1998; Strauss, 1987). By treating everything openly as data, a cohesive theory is achieved through continuous data analysis and theoretical coding.

GT is highly reliant on the data grounded in the field, gathered mostly through interviews with selected informants. Therefore, it utilises the inductive paradigm with a slight flavour of deductive and abductive model (Glaser, 1992; 1998). Coding is executed as soon as data is gathered and the procedure is constantly carried out. There are three common coding layers in GT, amid different terminology used by the variants (Glaser and Strauss, 1967; Corbin and Strauss, 2015; Charmaz, 2014). The first layer coding includes the analysis and classification of data as codes, while the second layer groups the codes into more detailed categories. Finally, the conceptual relations between codes are established by relating major categories to the principal category. It is this interrelation between categories, which is determined through theoretical coding, becomes the cohesive theory. Besides the similarities, several literature discuss the dissimilarities between GT coding procedures (LaRossa, 2005; Heath and Cowley, 2004; Bryant, 2017).

GT discourages extensive literature review prior to the inquiry due to its grounded data oriented nature. This discouragement is intended to avoid researchers from performing theory testing, instead of theory development (Glaser, 1992; Corbin and Strauss, 2015). In GT, everything is considered as data which not only include qualitative data, but also quantitative data and graphical data; visually and verbally as well as written (Glaser, 1998; Glass et al., 2002). GT requires researchers to dwell into the data and identify relevant sources via theoretical sampling in order to discover the theories (Glaser and Strauss, 1967; Charmaz, 2014). Theoretical sampling is a special form of purposive sampling by which the sample is based on the concepts that emerge from the data. This allows researchers to better understand the properties and dimensions of the gathered data. The concepts are evolved by theoretical sampling and gradually become narrowed, focused and eventually saturated. This is the stage at which the process has entered the theoretical saturation phase where no major new findings emerge and thus researchers shall stop collecting and analysing (Glaser, 1978). Researchers are required to possess theoretical sensitivity in order to understand and extract the main concepts to be considered in the theory (Glaser and Strauss, 1967; Glaser, 1978). By having theoretical sensitivity, the researchers should be able to conceptualise and create relationships between concepts.

When employing GT, researchers code and interpret data from the moment that data is collected. The data collection and analysis in GT therefore is done iteratively, incrementally and interleaved. The theoretical sampling is driven by the preliminary understanding established at the initial phase. With the arrival of new data, the analysis continuously compares them with the preliminary data (Birks and Mills, 2015). Eventually, the theoretical saturation is achieved after several cycles of constant comparison (Glaser, 1978). In addition to coding, GT also enforces memoing (Glaser, 1978). It is the act of writing the fundamental elaboration of concepts as they appear. It also includes researchers' reflection on the analytical and methodological steps that they take (Charmaz, 2014). Memos are equally as important as the codes because researchers engage with the data through them. The combination of codes and memos become the elements that generate the concepts and ultimately the theory. Due to its substantial role, memoing should begin at the very beginning of the inquiry.

As a discipline that bridges the technical and social worlds, SE have to deal with complex people-and-people, people-and-process and people-and-technology interactions (Tomayko and Hazaan, 2004). Such multifaceted interactions could not be handled by quantitative approaches alone. In addition, SE theories are still so few, rudimentary and fragmented. More concrete and cohesive SE theories should therefore be developed. Moreover, SE typically applies a structured approach to doing things. Due to these SE characteristics, GT is applicable to SE research because the method offers the most structured way for coding and interpreting data with regard to theory building (Creswell, 2013). Apparently, not much guidance on how to use GT in SE is available. Previous references on empirical SE research methods have not covered GT (Wohlin et al., 2003; Easterbrook et al., 2008). It is thus not obvious to SE researchers, novices in particular, on how to adopt and adapt GT in their research (Badreddin, 2013). This inadequacy motivates the sharing in this paper.

3. Methodology

Four SE research projects were used as the basis to derive GT practices for SE research proposed in this paper. The first project was about the methods for specifying requirements by means of graphical formal models (Razali and Garratt, 2009). The second project was about the systematic requirements prioritisation process in agile development (Al-Ta'ani and Razali, 2016). The third project explored approaches for selecting agile methods (Alqudah and Razali, 2016; 2017; 2018; Alqudah et al., 2019) and the fourth project identified the contributing factors for successful software systems integration (Othman and Razali, 2017). These projects mainly adopted Evolved GT (Corbin & Strauss, 2015).

The third project is described in detail in Section 5 as a case study to illustrate how the practices were derived. The following paragraphs briefly explain the methodology used in the project.

3.1 Objective and sampling

The project aimed to identify the critical factors that should be considered when selecting the most suitable agile methods for software development projects. Two main qualitative techniques were employed, namely interview and participant observation. A total of twenty-three interviews with agile experts were conducted. The agile experts were chosen from various organisations originated from different countries. The experts were selected due to their experiences in practising various agile methods while working in different organisations. For the participant observation, a multinational organisation that is based in Malaysia was selected. The team comprised twenty-eight members from several countries and with experiences in various agile methods during their working periods in different organisations.

3.2 Instrument and protocol

A set of interview questions (Appendix A) and observation notes were prepared as the instruments. The instruments were peer-reviewed before each session to ensure their suitability. Prior to the interview sessions, the appointments with the informants were set, who were identified through the theoretical sampling. After receiving consent from the informants, formal invitations were emailed together with the interview questions. For the observation sessions, the respective permissions were acquired before attending. Before each interview or observation, the informants or participants were briefed about the purpose of the sessions and they were assured about the confidentiality and anonymity. Each session lasted for 60 minutes in average and was recorded with written approvals from the informants or participants.

3.3 Data collection and analysis

The data were collected via interview and observation sessions. The interviews sessions, which were done either via face to face, email, Skype or phone call, were recorded. The interviews involved one close-ended and nine open-ended questions. The close-ended question was designed to acquire information about the agile experts' experiences. The open-ended questions were towards how the experts select agile methods in projects. The observation on the other hand was conducted to triangulate the data obtained from the interview sessions. The observation was conducted in one multinational organisation located in Malaysia, which has applied various agile methods in their projects. A qualitative analytical software, namely NVivo, was used during the analysis. The software is a large-sized textual data qualitative analysis tool that is intended to manage data transcriptions. The process started with transcribing the recorded interview data and observation notes, which were then keyed into NVivo. The three GT coding techniques, namely open coding, axial coding and selective coding, were used to qualitatively analyse the collected data. After codes were created and assigned, they were then categorised based on their relations and connections. The general categories were grouped under higher ordered headings. By combining similar and dissimilar categories into general ones, the categories were compacted. Later, the core category that related the categories together was identified. The core category acted as the hub for the rest of identified categories.

4. Encounters during GT adoption in SE research projects

When adopting GT in the four research projects, the authors experienced some encounters which were then adjusted to become adaptations. The following paragraphs elaborate the encounters and the corresponding adaptations made to suit the nature of SE research.

4.1 Encounter 1: Necessity of having research questions early

Classical GT (Glaser 1992; 2002) strongly stated that research questions should not be defined, whereby they should emerge from the research, because defining a research question upfront is considered as forcing. This principle seems difficult to be satisfied completely because the engineering nature of SE necessitates systematic work with clear objectives from the beginning. The authors realised that it was necessary to express the research questions in advance in order to guide the investigation into a particular area of concern. The research questions were also required to formulate the research design and to strategise the research effort.

Adaptation: Instead of defining specific research questions, the authors established preliminary ones. The questions were only areas of concern and grew over time during the course of the inquiry. The research questions indeed helped the authors as novices to develop theoretical sensitivity, which is deemed very important in GT. The theoretical sensitivity was established based on the understanding of the subject matter when the research questions were derived. The understanding was later needed for establishing concrete and interesting concepts when building the theories. The authors therefore adopted approaches in Evolved GT

(Corbin and Strauss, 2015) and Constructivist GT (Charmaz, 2014), which recommended research questions to be unrestricted, versatile and open to possibilities.

4.2 Encounter 2: Existence of preconceptions

To prevent researchers from having preconceptions and testing existing theory, GT opposes comprehensive literature prior to the inquiry (Glaser and Strauss, 1967). The literature review must be deferred until the research questions and theory are discovered. This is to ensure that the theory is original and genuine. To the authors, this principle seems to disprove the fact that an extensive literature review is needed for researchers to prove their familiarity with the subjects being researched. By nature, research is aimed to discover new knowledge. It is unreasonable to learn new knowledge unless the current knowledge is known, which can only be accomplished by reviewing the literature. With regard to SE, practitioners record many important information in the form of documents such as specifications, manuals, logs and archives. This practice is not the same as for social sciences, which rely greatly on direct communication with people. Besides serving as a point of reference, the documentation also contains written evidence that describes a phenomenon. In fact, SE best practices enforce practitioners to refer to certain documentation such as standards and handbooks before any activities or actions are conducted. Likewise, most SE researchers themselves are practitioners. The interest and motivation of doing research are usually enhanced by training, accumulated experience and values acquired throughout their professional years. These preconceptions in one way or another could influence at the beginning, how the inquiry is framed, who should participate and what outcomes should be expected. Such preconceptions definitely could neither be entirely removed nor overlooked. The authors concluded that the literature review and preconceptions could not be avoided during the inquiries.

Adaptation: Throughout the process, the authors performed literature review cautiously to ensure that the literature did not dominate thinking. As the phenomena investigated by the research projects were predominantly new, the preconceptions were limited because the existing literature did not discuss the subjects much. In fact, the literature review was shallow and acted only as secondary data for conducting theoretical sampling, formulating instruments and foreseeing further inquiries and more importantly developing theoretical sensitivity as novices. The authors found that it was impossible to achieve theoretical sensitivity unless the subject matters were understood. However, an open mindedness was consistently established so that new issues and dimensions grounded in the data could be recognised. In fact, the authors realised that the emergent codes and concepts grounded in the data were much easier to understand by connecting the data with the existing concepts and knowledge. Being aware of the current theoretical insights, the authors were inspired to discover more rich concepts and interrelationships, besides identifying gaps in the current understanding. Moreover, the personal knowledge and experience of the authors were also found to be useful in structuring and interpreting the unstructured data. Due to these reasons, the authors believe that GT essentially requires a balance between open-mindedness and the ability to identify emerging concepts. As long as the researchers remain open to the data, the literature review and the existing knowledge are still useful (Dunne, 2011; Hallberg, 2010; Hughes and Jones, 2003).

4.3 Encounter 3: Dynamic interactions between data collection and analysis

GT integrates data collection and continuous reflection during the whole inquiry (Corbin and Strauss, 2015; Charmaz, 2014; Bryant and Charmaz, 2007). Researchers must record their reflection in memos while collecting and coding the data (Glaser, 1978). Memos enable researchers to record preliminary attributes and associations between concepts as they arise as well as to recognise gaps in the theoretical sampling. The collected data are reviewed and re-examined multiple times to gain a sense of what is going on. This is achieved by constant comparison techniques, seeking similarity and differences patterns until the theoretical saturation is achieved. The re-examination and reflection in the memos lead to a richer understanding of the findings that shall subsequently be integrated as a cohesive theory by means of theoretical coding. In fact, memoing together with analysis, is the motivation to conduct theoretical coding. The authors learned that this nature of GT caused the border between data collection and analysis to be fuzzy, thus making the interplay between them complex. As the process progressed, information in the form of raw data, coded data and reflection memos were all around the place. The process was found to be too daunting to the authors, although SE as a discipline welcomes agility.

Adaptation: The authors acknowledged that it was extremely useful to use software tools during the process. The tools helped authors to code, sort and categorise the relationships between concepts. The tools also enabled data storage and retrieval. Constant comparisons could be made easily by contrasting statements on particular issues in various clusters of concepts. It was also possible to systematically trace concepts and their sources as well as the reflection memos associated with them. In addition, the tools helped to visualise concepts in the form of network views as perceptual maps as well as demonstrating possible interactions between the ideas generated. The maps promoted creative thinking and helped authors in theoretical coding with which they incorporated concepts into core categories. In essence, the perceptual maps can be regarded as software artefacts which are similar to conceptual models in SE. SE researchers including the authors are well accustomed to use diagrams in processing information, the maps indeed enhance the understanding. The ability to use software tools efficiently in managing things is also a strength of SE researchers who are mainly technology-savvy. Using tools to aid data analysis are highly encouraged by Contemporary GT (Bryant, 2017; Breuer, 2009; Goulding, 2002), although the practice is strictly contrary to Classic GT (Glaser, 1998; 2003).

4.4 Encounter 4: Uncertainties of sampling and availability

Theoretical sampling in GT asserts that theory development evolves from the time the ideas are conceived and later refined, rejected and reconceptualised (Glaser, 1978). The aim of theoretical sampling is to better understand the properties and dimensions of the emerging categories, as the process to achieve theoretical saturation. Memoing serves as the means that ties theoretical sampling to one discovery cycle, through which researchers identify additional data that are helpful to enhance current understanding and later establish the cohesive theory. While theoretical sampling is regarded as a form of purposive sampling, they are nevertheless indefinite (Charmaz, 2014). Unless the current data are reflected, the sampling is uncertain. As soon as the data suggest, researchers must gain access to the sources concerned. This requirement was found to be very challenging to the authors, since the resources and sources were limited, dispersed and restricted. The required practitioners could not be accessed immediately, especially those employed by organisations. The practitioners' availability as well as the organisation's data privacy and confidentiality often prohibit practitioners to participate (Badreddin, 2013; Timonen et al., 2018),

Adaptation: The authors had trouble accessing experts, as identified through theoretical sampling, who had experience with the subject matters under investigation. In some cases, experts were dispersed geographically and isolated. Due to this constraint, the authors ventured into other ways such as through social media and social networks. Special interest groups were found to be very helpful for obtaining expertise and reaching the right experts locally and worldwide. As social media are an informal forum for social interactions where people freely represent themselves, the authors recognised that responses from experts were genuine and unrestricted. Moreover, the existing peer-to-peer relationships in social media made the experts' inputs more meaningful and fruitful. Due to such relationships also, more experts were obtained when the purposive sampling became snowball sampling, as more and more experts were introduced. In future, the authors believe that the entries posted by the experts in social media could be analysed through sentiment analysis (Cambria and Hussain, 2015), from which their views on one specific topic are inferred. Crowdsourcing (Brabham, 2013) may also be explored to promote the public in providing expertise and professional advice. On the other hand, the authors utilised as many observation opportunities as possible to resolve experts' availability issues. Online monitoring of some tasks and some interviews were done via live streaming. Whenever technology is involved, SE researchers have the capability to explore and excel.

4.5 Encounter 5: Inexplicit theory development and presentation process

The coding of data in GT is crucial as the selected codes affect the results of the inquiry directly. To understand the phenomenon, coding alone is not enough. The codes must be summarised, sorted and cohesively conceptualised. In addition, to develop the concepts, considerable reflections should be made towards the codes. In developing a theory, researchers must go beyond the codes by finding out how the concepts work together and shape the theory. This is the creative aspect of the process that relies on the researchers' imagination and beliefs. The authors admitted that it was not that easy to articulate the process, particularly when the available GT references only explain about how data are obtained and analysed, instead of how the process shall be described and how the theory should be presented. Indeed, even among the social science research from which GT originates, there is no standard way to describe the process. Therefore, it was not easy for the authors to report the GT process step-by-step, mainly when textual data were involved. As SE practitioners, the authors are better at handling objective and quantitative data. It was thus a challenging experience to deal with subjective and qualitative data. The authors found the process to be quite formidable, especially when they were confused about what a theory actually is and how it should look like.

Adaptation: For the visualisation of concepts, the authors followed the traditional SE practices. Instead of only texts, the authors used box-and-arrow diagrams and tables to explain concepts. The graphical representations offer more readily graspable interpretations of the theories, thus self-descriptive. The objectivity by structure in graphical representations are far more intuitive. By using the software tools, the authors found that the graphical representations were not so difficult to produce. In addition, the software tools helped to outline the concepts from the beginning until reaching the theoretical coding. The tools also enabled within and across concepts tracing as well as across different coding layers. The authors realised that reporting the theory development process was quite straightforward with the help of software tools. It was simply a task of translating the visual data into text description. The authors however acknowledged the fact that the tools were merely meant to ease the process. The actual coding, conceptualising and theorising the data had still to be done by the authors themselves.

5. GT practices for SE research: a case study of agile methods selection process

From the above encounters and adaptations, the following practices were derived as a guide for GT adoption in SE research. Table 1 summarises the encounters and the corresponding practices to adapt them. These practices are not meant to be definite and complete. Therefore, future SE research that adopt GT are highly encouraged to validate and refine them further. In future, it is believed that these practices could evolve to become best practices when significant evidence about GT adoption in SE research are obtained. In the meantime, they can be the practices for novices who intend to use GT in SE research.

Table 1: Encounters and practices to overcome

Encounter	Practice
Necessity of having research questions early	Develop broad preliminary research questions and modify them as the inquiry progresses
Existence of preconceptions	Conduct the literature review cautiously and restrain previous experience to prevent biases
Dynamic interactions between data collection and analysis	Take full advantage of software tools for data management
Uncertainties of sampling and availability	Explore diverse channels of communication to gain access
Inexplicit theory development and presentation process	Utilise graphical representations to explain the process and present the cohesive theory

This section continues with the explanation on how the practices were derived through a SE research project. The project was about the development of agile methods selection framework, which is one of the four research projects mentioned in this paper.

5.1 Practice 1: Develop broad preliminary research questions and modify them as the inquiry progresses

The research began by defining preliminary research questions, which assisted the researchers to understand the context of agile methods selection. The research intended to investigate the critical factors that should be considered when selecting agile methods for software development projects. The preliminary research questions led to searching of relevant information in the literature. Through the literature, the researchers developed a better understanding of the topic and became more sensitive towards the issues about the topic. The preliminary questions were then modified based on the emerging insights gained throughout the inquiry. For instance, “How can the appropriate agile methods be determined?” was one of the preliminary research questions that later become “How can the appropriate agile methods’ practices be identified?” and “What are the critical factors that should be considered when selecting the appropriate agile methods together with the respective practices? How could these factors become a pragmatic framework for selecting the appropriate agile methods and practices?”. The questions evolved based on the data obtained from a series of interviews with agile experts. The researchers realised from the interviews that merely identifying the agile methods and how they are determined was not enough without knowing the practices that the methods contain. As there are many interrelated factors that influence the selection, the framework has to be conceptualised in a pragmatic way to ease the understanding.

The research employed in-depth interviews with agile experts, with the aim to examine and understand the way of how the experts select agile methods for various projects. Therefore, the research adopted semi-structured interviews with open-ended and probing questions to help the researchers to obtain the views, experiences, perceptions and viewpoints of the experts on agile methods selection. By starting with a broad brush tentative research questions, the researchers were able to construct tentative interview questions, which assisted them to drive the inquiry. As the interviews progressed, more questions were then added, altered and evolved as and when needed. For instance, when the experts provided insights with regards to a novel agile method, the researchers were prepared to ask follow-up questions to understand how the selection of the method was done. Similarly, when the experts were asked about the process of selecting agile methods, there were occasions where the experts highlighted factors that were not mentioned by other experts in previous interviews. Hence, follow-up and probing questions were asked accordingly to ensure the new factors were considered and clearly understood. It was thus necessary to develop tentative questions that were unrestricted and flexible so that they could be modified based on the data obtained from the field.

5.2 Practice 2: Conduct the literature review cautiously and restrain previous experience to prevent biases

In reality, there are a number of agile methods that are available in the industry. However, the researchers were not aware and familiar with all of them at first. In fact, each agile method has its own practices and in fact, the practices vary a lot. Hence, the researchers had to conduct literature reviews in order to understand various agile methods and their practices. By reviewing the literature, the researchers developed theoretical sensitivity especially in understanding the differences between the available agile methods, their characteristics and practices as well as the possible ways of adopting them. The researchers considered the process as 'building oneself understanding' rather than 'building the theory'. This means the focus of the analysis was still towards the inputs obtained from the experts as well as the observation notes collected from the field. Similarly, although the researchers possessed some experience of using agile methods and had been involved in software development projects for quite some time, the experience was kept from over influencing the inquiry. The literature review and previous experience however were found to be very helpful especially when the researchers started recording the memos and relating them to the codes.

5.3 Practice 3: Take full advantage of software tools for data management

The research adopted GT to develop the theory of agile methods selection. The gathered data were analysed through open coding, axial coding and selective coding by using a software tool, namely NVivo. The tool was invaluable in tackling considerable amount of research materials, notes and related theories. After all, NVivo is intended to assist in analysing data that empowers researchers to connect, search and sort data (Niedbalski and Ślęzak, 2017). NVivo also enables the management of interview transcripts, the creation of codes and quotations as well as memos storage. Despite its outstanding capability in managing and organising data, the interpretation processes were still manually done by the researchers, who remained in total control of the intellectual process. The researchers admitted that the assignment, creation and sorting of codes without using NVivo would be very difficult especially when considering tons of codes under each selection factor. In other words, the categorisation and the identification of relationships among the factors would not be properly done without the assistance of the tool. Another benefit of using NVivo was the generation of the tree map that demonstrates the categories including the factors that affect the selection of agile methods. The tree map enhanced the researchers' understanding of the selection factors, which later assisted the researchers to shed light on the process of selecting agile methods.

5.4 Practice 4: Explore diverse channels of communication to gain access.

Agile Malaysia is a Facebook group that was launched by agile practitioners in Malaysia, within which articles, experiences and issues faced by the members of the group are aired. The group shares the dates and times of the monthly meetings conducted in different software development organisations in the city of Kuala Lumpur. The researchers fully utilised the opportunities concerning the monthly meetings to search for agile experts in different projects and using different agile methods. In the meetings, the researchers would solicit the organisers to introduce them to agile experts and to obtain the emails of experts who have skills in using more than a single agile method. In addition, the researchers also joined social media groups like Linkedin to search for practitioners in the field who have experience in using more than one agile method – and fortunately, after the experts were contacted for their participation, most of them replied promptly. Twenty-three experts from different organisations in various countries participated, among which 7 were from Malaysia, 2 each from Australia, Singapore, Palestine and United States respectively, 1 person each from organisations in the United Kingdom, Canada, Kuwait, Netherlands, New Zealand, Egypt, India and Japan respectively.

The researchers used face-to-face interviews with some, while others were interviewed through Skype or telephone calls due to the distance entailed. Two participants answered the questions through email and others through live interview sessions. To understand fully the selection of agile methods, observation sessions were also conducted to complement the interviews. The main objective was to recognise the way of how agile team members choose the suitable agile methods to use in a project. Hence, the researchers joined the agile team members in their meetings to be able to observe them directly. The researchers observed the members by attending their meetings at the beginning of a project and during their retrospective meetings, in order to comprehend the selection process. By employing participant observation, the sessions allowed the researchers to ask questions for clarification and to engage in informal discussions with the agile experts. During the observation sessions, the researchers asked questions to the agile experts in order to reflect and support the interview data. The feedback from the team members were written down in a notebook. The researchers also took photos of the meeting sessions, including the whiteboards where the members scribbled information during the discussion of selecting the agile methods. One team member acted as a scribe, drew columns of several agile methods as the headers. Whilst discussing which method to select, the scribe noted down everything on the board. The researchers conducted the observation sessions in one Malaysian organisation, which has applied different agile methods in their projects. Although the organisation is located in Malaysia, there were twenty-eight team members who were scattered around various countries. The company has used XP, SCRUM, Lean Startup, Lean Development, SCRUM of SCRUM and Kanban in executing their previous software development projects.

5.5 Practice 5: Utilise graphical representations to explain the process and present the cohesive theory

The research employed a graphical representation to present the model or framework for selecting agile methods in software development projects. The framework, as shown in Figure 1, illustrates the factors that have been found to affect the selection of agile methods. The agile methods whose efficacy is affected by these factors are SCRUM, Extreme Programming (XP), Test Driven Development (TDD), Agile Modeling (AM), Crystal Clear (CC), DevOps/Continuous Delivery, Dynamic Systems Development Method (DSDM), Adaptive Software Development (ASD), Feature Driven Development (FDD), Lean Development (LD), Lean Startup (LS), and Kanban, and the scaled agile methods, which are Crystal Red (CR), Disciplined Agile Delivery (DAD), Scaling Agile Framework (SAFe), Large Scale Scrum (LeSS), SCRUM of SCRUMS, Nexus, Nexus+ and Spotify. Each method comprises different practices and can be used for different projects in different situations. The factors were grouped into five categories namely customer involvement, organizational culture, project constraints, nature of project and development team proficiency. The framework indicates that the agile team members should conduct a meeting before embarking into any project that opts to use agile methods. In the meeting, the members shall assess the degree of which each factor exists in the project. The combination of those degrees would later determine the most suitable agile method to be employed in the project. Alqudah et al. (2019) has the details of the framework. The framework is indeed the cohesive theory produced by the research.

In explaining the theory development and presentation process through graphical representations, the researchers employed SE's conservative modelling styles, similar to Data Flow Diagram (DFD) and Activity Diagram (AD). DFD and AD outline the processes entailed and the data that flow from one process to the other. For the framework itself, the representation used is comparable to combining Context Diagram (CD) and Entity-Relationship Diagram (ERD) together, with some modifications to suit the interests of the research. The diagrams depict the entities involved and their respective attributes as well as their relations to one and another. The drawing of these diagrams was accomplished by using software tools.

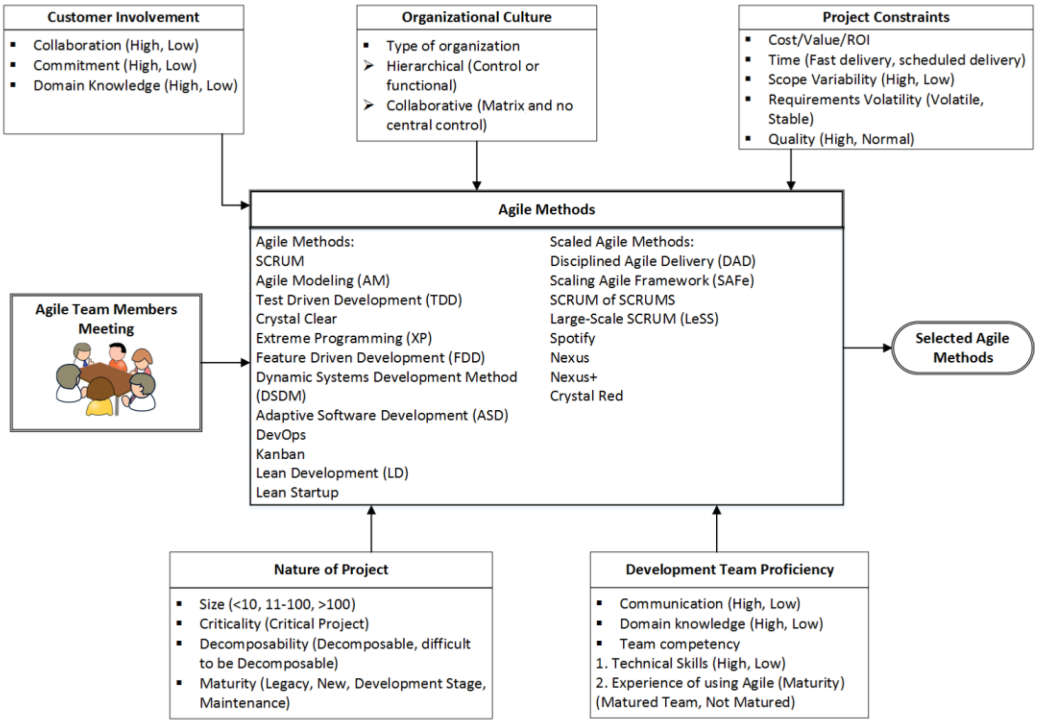
Customer Involvement
Collaboration (High, Low)
Commitment (High, Low)
Domain Knowledge (High, Low)
Organizational Culture
Type of organization
Hierarchical (Control or functional)
Collaborative (Matrix and no central control)
Project Constraints
Cost/Value/ROI
Time (Fast delivery, scheduled delivery)
Scope Variability (High, Low)
Requirements Volatility (Volatile, Stable)
Quality (High, Normal)
Agile Team Members Meeting
Agile Methods
Agile Methods:
SCRUM
Agile Modeling (AM)
Test Driven Development (TDD)
Crystal Clear
Extreme Programming (XP)
Feature Driven Development (FDD)
Dynamic Systems Development Method (DSDM)
Adaptive Software Development (ASD)
DevOps
Kanban
Lean Development (LD)
Lean Startup
Scaled Agile Methods:
Disciplined Agile Delivery (DAD)
Scaling Agile Framework (SAFe)
SCRUM of SCRUMS
Large-Scale SCRUM (LeSS)
Spotify
Nexus
Nexus+
Crystal Red
Selected Agile Methods
Nature of Project
Size (<10, 11-100, >100)
Criticality (Critical Project)
Decomposability (Decomposable, difficult to be Decomposable)
Maturity (Legacy, New, Development Stage, Maintenance)
Development Team Proficiency
Communication (High, Low)
Domain knowledge (High, Low)
Team competency
1. Technical Skills (High, Low)
2. Experience of using Agile (Maturity) (Matured Team, Not Matured)

Figure 1: Framework for selecting agile methods in software development projects

6. Conclusion

In general, GT trains SE researchers to code, relate and merge different grounded concepts into an integrated and cohesive theory. Among other qualitative methods, GT provides SE researchers with a systematic approach to overcome the subjective nature of qualitative research. As a method, GT entails certain principles that must be complied by its adopters throughout the inquiry. SE on the other hand possesses specific characteristics that pose some challenges to GT adoption. To ease the adoption of GT in SE research especially among novices, some GT principles therefore need to be adjusted to suit the nature of SE. This paper outlined some encounters of using GT in SE research by novices. The encounters were turned into adaptations, which were then translated as GT practices for SE research. The practices enhance the strength of SE researchers while ensuring the GT principles are still conformed. GT has bright future in SE research in years to come. This paper could therefore become a reference for GT novices to start using the method in SE research.

Acknowledgements

One of the research projects included in the paper is funded by the University Grant Scheme (GUP-2018-005).

References

Adolph, S., Hall, W. and Kruchten, P. 2011. Using grounded theory to study the experience of software development. *Empirical Software Engineering*, 16(4), 487-513.

Alliance, A. 2016. *What is agile?*. Agile Alliance. https://www.agilealliance.org/agile101.

Alqudah, M., and Razali, R., 2016. A review of scaling agile methods in large software development. *International Journal on Advanced Science, Engineering and Information Technology*, 6(6), 828-837

Alqudah, M. K., and Razali, R., 2017. Key factors for selecting an agile method: a systematic literature review. *International Journal on Advanced Science, Engineering and Information Technology*, 7(2), 526-537.

Alqudah, M., and Razali, R., 2017. A comparison of Scrum and Kanban for identifying their selection factors. *In 2017 6th International Conference on Electrical Engineering and Informatics (ICEEI),* (pp. 1-6).

Alqudah, M. and Razali, R., 2018. An empirical study of Scrumban formation based on the selection of Scrum and Kanban Practices. *International Journal of Advanced Science, Engineering and Information Technology*, 8 (6), 2315-2322.

Alqudah, M., Razali, R. and Alqudah, M., 2019. Agile methods selection model: A Grounded Theory Study. *International Journal of Advanced Computer Science and Applications*, 10(7), 357-366.

Al-Ta'ani, R. H. and Razali, R., 2016. A framework for requirements prioritisation process in an agile software development environment: empirical study. *International Journal of Advanced Science, Engineering and Information Technology*, 6 (6), 846-856.

Badreddin, O., 2013. Thematic review and analysis of grounded theory application in Software Engineering. *Advances in Software Engineering*.

Beck, K. et al., Manifesto for agile software development. <https://agilemanifesto.org/>. [Accessed 28 September 2020]

Birks M and Mills J., 2015. *Grounded Theory: A Practical Guide*. 2nd ed. London: SAGE.

Bourque, P. and Fairley, R.E., eds., 2014. *Guide to the Software Engineering Body of Knowledge*-SWEBOK. Version 3.0, IEEE Computer Society. http://www.swebok.org.

Boychuk Duchscher, J. E., 2004. Grounded theory: reflections on the emergence vs. forcing debate. *Journal of Advanced Nursing*, 48(6), 605-612.

Brabham, D. C., 2013. *Crowdsourcing*. Cambridge, Massachusetts; London, England: The MIT Press.

Breuer, F. 2009. *Reflexive grounded theory: an introduction for research practice*. Wiesbaden: VS Verlag Bryant, A. and Charmaz, K., 2007. *The SAGE Handbook of Grounded Theory*, Sage.

Bryant, A., 2017. *Grounded theory and grounded theorizing: pragmatism in research practice*. New York, NY: Oxford University Press.

Cambria, E. and Hussain, A., 2015. *Sentic computing: a common-sense-based framework for concept-level sentiment analysis*. Springer.

Charmaz, K., 2014. *Constructing grounded theory.* 2nd ed. London, England: Sage.

Clarke, A. E., 2005. *Situational analysis: grounded theory after the postmodern turn*. Thousand Oaks, CA: Sage.

Coleman, G. and O'Connor, R., 2008. Investigating software process in practice: a grounded theory perspective. *Journal of Systems and Software*, 81, 772-784.

Corbin, J., and Strauss, A., 2015. *Basics of qualitative research: techniques and procedures for developing grounded theory*. 4th ed. Thousand Oaks, CA: Sage.

Creswell, J.W., 2013. *Qualitative inquiry & research design: choosing among five approaches*. 3rd ed. Sage.

Dunne, C., 2011. The place of the literature review in grounded theory research. *International Journal of Social Research Methodology*, 14, 111–124.

Easterbrook, S., Singer, J., Storey, M.A. and Damian, D., 2008. Selecting empirical methods for software engineering research. In: *F. Shull, J. Singer and D. I. K. Sjøberg* (Eds.) *Guide to Advanced Software Engineering*. Springer.

Glaser, B. G. and Strauss, A. L., 1967. *The discovery of grounded theory: strategies for qualitative research*. Aldine de Gruyter, New York.

Glaser, B. G., 1978. *Theoretical sensitivity*. Sociology Press.

Glaser, B. G., 1992. *Basics of grounded theory analysis: emergence vs forcing*. Sociology Press.

Glaser, B. G., 1998. *Doing grounded theory: issues and discussions*. Sociology Press.

Glaser, B. G., 2002. Constructivist grounded theory?. *Forum: Qualitative Social Research*, 3, 3, Art. 12.

Glaser, B. G., 2003. *The grounded theory perspective ii: description's remodeling of grounded theory methodology.* Mill Valley, CA: Sociology Press.

Glass, R. L., Vessey, I. and Ramesh, V., 2002. Research in software engineering: an analysis of the literature. *Information Software Technology*, 44(8), 491-506.

Goulding, C., 2002. *Grounded theory: a practical guide for management, business and market researchers*. London: Sage.

Hallberg, L. R. M., 2010. Some thoughts about the literature review in grounded theory studies. *International Journal of Qualitative Studies on Health and Well-Being*, 5(3), 10.3402/qhw.v5i3.5387.

Heath, H. and Cowley, S., 2004. Developing a grounded theory approach: a comparison of Glaser and Strauss. *International Journal of Nursing Studies*, 41, 141-150.

Hoda, R., Noble, J. and Marshall, S., 2011. Grounded theory for geeks. *In proc. 18th conference on pattern languages of programs*.

Hoda, R., Salleh, N. and Grundy, J., 2018. The rise and evolution of agile software development. *IEEE Software*, 35 (5), pp. 58-63. https://doi.org/10.1109/MS.2018.290111318.

Hughes, J and Jones, S., 2003. Reflections on the use of grounded theory in interpretive information systems research. *Electronic Journal of Information Systems Evaluation*, 6(1).

Kelle, U., 2005. “Emergence" vs. "forcing" of empirical data? A crucial problem of "grounded theory" reconsidered. *Forum: Qualitative Social Research*, 6(2).

Kempster, S. and Parry, K., 2011. Grounded theory and leadership research: a critical realist perspective. *The Leadership Quarterly*, 22, 106–120.

Kempster, S. and Parry, K., 2014. Critical realism and grounded theory. *In P. K. Edwards, J. O’Mahoney, and S. Vincent (Eds.), Studying Organizations using Critical Realism: A Practical Guide*. Oxford, England: Oxford University Press.

LaRossa, R., 2005. Grounded theory methods and qualitative family research. *Journal of Marriage and Family*, 67, 837–857.

Lee, F. S., 2016. Critical realism, method of grounded theory, and theory construction. *In F. S. Lee and B. Cronin (Eds.), Handbook of Research Methods and Applications in Heterodox Economics*. Cheltenham, England: Edward Elgar.

Niedbalski J., Ślęzak I., 2017. Computer assisted qualitative data analysis software. Using the NVivo and Atlas.ti in the research projects based on the methodology of grounded theory. *In: Costa A., Reis L., Neri de Sousa F., Moreira A., Lamas D. (eds) Computer Supported Qualitative Research. Studies in Systems, Decision and Control*, 71. Springer, Cham. https://doi.org/10.1007/978-3-319-43271-7_8

Oliver, C., 2012. Critical realist grounded theory: a new approach for social work research. *British Journal of Social Work*, 42, 371–387.

Orlikowski, W. J., 1993. CASE tools as organizational change: investigating incremental and radical changes in systems development. *MIS Quarterly*, 17(3), 309–340.

Othman, M. H. and Razali, R., 2017. Electronic government systems interoperability model. *Journal of Telecommunication, Electronic and Communication Engineering*, 9(3), 1-9.

Razali, R., Garratt, P., 2009. A usability profile for graphical formal modelling methods. *Frontiers in Artificial Intelligence and Applications*, 199, 443–462.

Sharma, A. & Bawa, R. K., 2016. A Framework for Agile Development Method Selection using Modified PROMETHEE with Analytic Hierarchy Process. *International Journal of Computer Science and Information Security*, 14(8), 846.

Sommerville, I., 2016. *Software engineering global edition*. 10th ed. Pearson.

Stol, K., Ralph, P. and Fitzgerald, B., 2016. Grounded theory in software engineering research: a critical review and guidelines. *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*, Austin, TX, 2016, 120-131.

Strauss, A. L., 1987. *Qualitative analysis for social scientists*. Cambridge University Press.

Timonen, V., Foley, G., and Conlon, C., 2018. Challenges when using grounded theory: a pragmatic introduction to doing GT research. *International Journal of Qualitative Methods*.

Tomayko, J. E and Hazaan, J. E. T. O., 2004. *Human aspects of software engineering*. Charles River Media, Massachusetts, Mass, USA.

Wohlin, C., Höst, M. and Henningsson, K., 2003. Empirical research methods in software engineering. In: ESERNET, volume LNCS 2765. Springer.

Appendix A

Interview Protocol & Instrument

Time of Interview:
Date:
Place:
Interviewee ID:

Introduction

We are conducting a research in Agile software development specifically in the selection of Agile methods for software projects. Currently, we are investigating the process of selecting Agile methods based on the experiences of Agile project managers and other decision makers who are involved in the process. The main purpose of this research is to identify the critical factors that affect the selection of Agile methods. You have been selected as an interviewee because you have been involved in selecting Agile methods for software projects in your organisation. You shall participate by sharing your experience in the selection process as well as highlighting the contributing factors that influence the selection. The interview is scheduled to last approximately one hour. The main focus is on the questions that are related to Agile methods selection. The interview is recorded for data analysis purposes.

Interview Question for Agile methods' selection:

1. Which Agile methods have been selected and used at your organisation across different projects? (Note: Please select as many methods that have been used at your organisation)

1. Crystal Clear
2. Crystal Red
3. Crystal Orange
4. Crystal Yellow
5. XP
6. Agile Modeling
7. Adaptive Software Development
8. Feature Driven Development (FDD)
9. Dynamic Systems Development Method (DSDM)
10. Disciplined Agile Delivery (DAD)
11. SCRUM
12. DevOps/continuous Development
13. SAFs
14. Cynefin
15. Lean Development
16. Kanban
17. Internet Speed Development (ISD)
18. Lean Startup
19. Others……………………………………………………………………………………

2. How long have you been practising the selected methods? How was the experience?
3. What are the problems that you intend to solve when selecting different Agile methods and what were the outcomes?
4. Why are there many Agile methods? How do they differ from one to another? How about the practices? How do the practices relate to the methods?
5. Describe the process of selecting the appropriate Agile methods at your organisation.
6. What was/were the Agile method(s) selected for your last project? How was/were it/they selected?
7. The factors below are considered as among the crucial factors when selecting the appropriate Agile methods and/or Agile method practices. Do you agree?
From your experience, to which extent these factors are important when making decisions to select the appropriate Agile methods for software projects.

Selection Factors	Interviewee's Views
Nature of Project	
- **Project Size (number of people involved)**	
- **Project Criticality**	
- **Project Decomposability**	
Stakeholder Skill	
- **Communication**	
- **Knowledge of Domain**	
- **Team Competency**	
Organizational culture	
- **Type of organization (Hierarchical, Collaborative and others)**	
▪ **Hierarchical is the same as control or Functional**	
▪ **Collaborative is the same as matrix or democratic**	
Project Constraints	
- **Cost/value/ROI**	
- **Time**	
- **Scope**	
- **Quality**	
Customer Involvement	
- **Collaboration**	
- **Commitment**	
- **Knowledge of domain**	

8. Are there any other factors that could affect the selection of Agile methods and/or Agile method practices?
9. Who are involved in making decisions when selecting the appropriate Agile methods in your team? Why are they being involved? What are their roles? What information do they contribute?
10. Do you know any practitioners that have the same role/experience as you? Do you mind introducing them to us?