A GROUNDED THEORY OF THE REQUIREMENTS ENGINEERING PROCESS

Layla Alfawzan[1] and Alphonso Bellamy[2]

[1]Eastern Michigan University, USA
[2]College of Technology, Eastern Michigan University, Ypsilanti, MI 48197

ABSTRACT

This paper explores the requirements engineering (RE) process by conducting interviews with RE professionals and applying grounded theory to determine whether a theory of RE emerges. Analysis of the interviews revealed prominent data patterns that were used to model the RE process. The model depicts the RE process as one of establishing a match through discovery and streamlining, and which utilizes corrective measures in order to manage threats to establishing a match. The process involves many entities but is mainly conducted by RE professionals whose experience plays a major role in extracting complete requirements and detecting occasions of mismatch between customer needs and the software requirements, which represent their main concern during the process.

This paper contributes to the empirical analysis of RE by presenting evidence of the RE process in its basic form as carried out in industry, which may form as a building block for further RE research.

KEYWORDS

Requirements engineering, Grounded theory & Empirical software engineering

1. INTRODUCTION

Software development is a costly and risky endeavor. Cost overruns, project delays, stakeholder frustration, and defective products plague the industry. Brooks [1] argued that the single hardest part of conceptually defining a software system is the task of delineating the software requirements. He stressed, "no other part of the work so cripples the resulting system if done wrong" and "no other part is more difficult to rectify later" [1, p. 17]. The challenge is compounded by the fact that most customers are unsure of what they want and, quite often, do not even detail the problem appropriately to extract full and accurate requirements. Moreover, most software engineering practices assume that it is possible for software requirements to be fully specified in advance, which Brooks contended was fundamentally false. In light of these difficulties and given how significant the industry is, it is therefore of utmost importance to better manage the process of defining and maintaining software requirements, commonly known as the process of requirements engineering (RE). Yet, RE is a highly complicated and error-prone process. While recommendations for best practices may exist in the literature, there seems to be a wide discrepancy between what is published on RE and its applications in the industry. Therefore, the present study attempts to address this gap by conducting a qualitative study in order to determine if a theory of RE process emerges, which may prove highly useful if put forward towards improving the existing RE practices.

DOI: 10.5121/ijsea.2019.10501

1.1 Problem Statement and Research Questions

The process dynamics of RE in terms of the activities involved and the associated complications, in addition to how these dynamics affect project outcomes, are insufficiently described in empirical research. In order to address this problem using qualitative methods, our research was guided with the general question "How is the process of RE carried out?"

1.2 Objective

The objective of the study is to generate a theory of the RE process, and we therefore gathered information pertaining to the current RE practice. This information constituted descriptions of the activities involved in the RE process, including—but not limited to—elicitation, analysis, specification, validation, and management. Additional information regarding the issues that arise during this process and how these process dynamics affect project outcomes was also collected. Interviews were conducted with 11 RE professionals in order to gather the data.

1.3 Contribution

This paper contributes the following:

1. Empirical evidence to describe the RE process in its basic form as carried out in industry, which may serve as a building block for further RE research.
2. Empirical evidence of RE challenges and ways to overcome them.
3. A grounded theory of the RE process based on empirical evidence, which adds to theory building research.
4. A summary of the experience and challenges of professionals carrying out an effective RE process with numerous lessons that can benefit practitioners.

2. BACKGROUND

2.1 Requirements Engineering

A software system is assessed in terms of the extent to which it satisfies the purpose for which it was intended. Nuseibeh and Easterbrook [2] viewed the process of RE as one of discovering that purpose, and which faces a number of challenges inherent in distributed stakeholders with varying needs and unclear or implicit goals. Zave [3] defined RE as:

The branch of software engineering concerned with the real-world goals for functions of and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families. (p. 315)

Contrary to common misconception, RE is not a discrete front-end exercise, but rather an ongoing activity that is initiated at the start of a software development project and which continues to evolve along the lifecycle of the project [4]. According to Kotonya and Sommerville [5], the focus of the RE process starts at comprehending the requirements and shifts towards the production of requirements and system modeling. They defined the standard process of RE as consisting of four steps: requirements elicitation, analysis and negotiation of the requirements, requirements specification, and requirements validation. The SWEBKO software requirements

knowledge area adds a fifth requirement activity to Kotonya and Sommerville's four steps, namely requirements management.

2.2 RE Literature

As is the case with the bulk of engineering research, the conventional paradigm of RE research has followed the philosophical school of positivism, which regards knowledge as being accumulated through the systematic observation of natural phenomena. This position has been the center of criticism [6], where papers like Hinds [7] viewed the positivist approach as being detrimental, at best, and contradictory to the process of RE. As a result, social factors have not received notable attention until recently when the RE community realized that, unlike the broad process of software engineering, the challenges posed to the RE process are human-centered.

In order to classify the empirical efforts that have been thus far dedicated to the study of RE, we mirror Robey, Ross, and Boudreau's [8] approach, which utilizes Mohr's [9] distinction between variance and process research. While variance research attempts to relate specific effects to possible antecedents, process research does the opposite by exploring a series of events over time as they affect the final outcomes. We found empirical RE to exhibit an orientation towards variance research. In addition, efforts of both approaches have been mostly descriptive, while little attention has been dedicated to developing a strong theoretical foundation for the RE process. Therefore, we attempted to address this shortage by employing a process theory approach to exploring RE. In order to generate a concrete theory of RE, we adopt a grounded theory [10] orientation, which is of merit to numerous software engineering and RE endeavors [11][12][13][14][15][16][17][6][18][19]. As with all grounded theory efforts, the study does not employ an a priori theoretical perspective but rather allows the data to dictate the theory.

Glaser [20], one of the founders of grounded theory, argues that an extensive search of the literature could preclude the process of developing a grounded theory by forcing existing theoretical concepts on the collection and analysis of data. However, this argument cannot invalidate the vitality of framing a research problem and identifying its significance within the context of extant literature, especially since the current study was conducted as a PhD research effort. Therefore, we opted for a review that is wide in the topics and the amount of literature efforts it covers, yet general in that it doen't closely examine the research efforts in detail. This particular approach is followed when covering studies that are similar to ours in terms of topic, research design, and methods so as to avoid being influenced by their results. The results of this review are summarized in Table 1.

.

Table 1—Selected RE empirical literature and its classification

	Variance	Process
Approaches and techniques	• Maiden & Sutcliffe (1992) • Maiden & Rugg (1996) • Fields, Harrison, & Wright (1997) • Sawyer, Sommerville, & Viller (1998) • Sutcliffe & Maiden (1998) • Sutcliffe & Ryan (1998) • Pocock, Harrison, Wright, & Johnson (2001) • Gizikis & Robertson (2004) • Ramos, Berry, &Carvalho's (2005) • Zowghi & Coulin (2005) • Carrizo, Dieste, & Juristo (2008) • Jiang, Eberlein, & Far (2008) • Karlsen, Maiden, & Kerne (2009) • Colomo-Palacios, Casado-Lumbreras, Soto-Acosta, and Garcia-Crespo (2011) • Pilat &Kaindl (2011) • Proynova, Paech, Koch, Wicht, &Wetter (2011) • de Gea, Nicolás, Alemán, Toval, Ebert, & Vizcaíno (2012) • Halaweh (2012) • Miller, Pedell, Sterling, Vetere, & Howard (2012) • Lopes & Forster (2013) • Lopes & Forster (2013)Burnay, Jureta, & Faulkner (2014) • Fernandez &Penzenstadler (2014) • Olmos & Rodas (2014) • Bhowmik, Niu, Savolainen, & Mahmoud (2015) • Khanom, Heimbürger, & Kärkkäinen (2015) • Miller, Pedell, Lopez-Lorca, Mendoza, Sterling, & Keirnan (2015) • Horkoff & Yu (2016) • Würfel, Lutz, &Dieh (2016) • Oriol, M., Stade, M., Fotrousi, F., Nadal, S., Varga, J., …& Schmidt, O. (2018) • Sánchez, E., & Macías, J. A. (2019)	
Challenges	• Nidumolu (1996) • Moynihan (2000) • Grunbacher & Briggs (2001) • Coughlan, Lycett, & Macredie (2003) • Solemon, Sahibuddin, & Abd Ghani (2009) • Stoiber & Glinz (2009) • Ferreira, Shunk, Collofello, Mackulak, & Dueck (2011) • Huysegoms, Snoeck, Dedene, Goderis, and Stumpe (2014) • Pena & Valerdi (2015) • Chari, K., & Agrawal, M. (2018)	• Al-Rawas & Easterbrook (1996) • Hall, Beecham, & Rainer (2002) • Hansen & Lyytinen (2010) • Nidhra, Yanamadalaa, Afzalb, & Torkara (2013) • Jantunen, & Gause (2014)
RE Process	• Lauesen & Vinter (2001) • Damian, Zowghi, Vaidyanathasamy, & Pal (2004) • Verner & Bleistein (2006) • Fernandez & Wagner (2017) • Wagner, S., Fernández, D. M., Felderer, M., Vetrò, A., Kalinowski, M., Wieringa, R., …& Lassenius, C. (2019)	• Chakraborty, Sarker, & Sarker (2010) • Jarke, Loucopoulos, Lyytinen, Mylopoulos, & Robinson (2011)

3. RESEARCH METHODOLOGY

The study adopts a qualitative research design with a case study approach and a grounded theory orientation in order to fulfill its purpose of generating a theory of the RE process. Additional details are presented in the following sections.

3.1. Subjects

The organization under study, thereafter referred to as X-Solutions, is a technology solution and service provider with more than twenty years of experience in the field. It offers a wide variety of solutions and services that include software development, mobile applications, e-commerce, infrastructure, and customer relationship management (CRM). It is based in North Africa and operates through offices in several countries over the Middle East, Europe, and North America. It employs more than 350 certified professionals that include architects, designers, developers, quality engineers, and project managers. X-Solutions' clients include governments, Fortune 500 companies, and multinational businesses of varying industries covering telecommunications, financial, healthcare, media and entertainment, real estate, and retail. The company is ISO 9001:2000 certified and CMMI level 3 accredited.

Subjects consisted of 11 RE professionals with experience ranging from one to twelve years in the field. They were of varying job titles that included systems analyst, business application specialist, business application consultant, and presales technical consultant. However, they were all responsible for carrying out the process of RE. The group was diverse in terms of the industries they served (e.g. government, financial, and pharmaceuticals) and the products they delivered, which included custom software solutions, enterprise resource planning (ERP), and CRM.

3.2.Data Collection

The data was collected using a total of 11 semi-structured phone interviews, each with a different subject. Interview durations ranged from 30 minutes to over an hour and were audio recorded. The interviews were later transcribed, at which point the audio recordings were destroyed. Admittedly, interview recording is something that Glaser advises against. However, as a novice qualitative researcher, attempting to rely on note taking and personal memory in order to reconstruct interviews represented not only a challenge to the main author but also a risk to the quality of the study. Moreover, Adolph et al. [21]insist that the most efficient way of obtaining interview data is to have them recorded and transcribed, and hence we followed their advice.

During the interviews, subjects were asked to describe the dynamics of the RE process in their own words. Where subjects did not volunteer to discuss topics of interest that are extracted from the problem of the study, an effort was made to guide the discussion to such topics. In order to make sure that the interview protocol excluded leading questions, a list of potential questions was prepared beforehand to cover these topics and guide the researcher during the interview. The questions were classified into two levels: Level 1 primary questions that were covered during each interview, and Level 2 secondary questions that were only utilized when the discussion headed towards their topic or when the subject was being very brief. For instance, all the subjects were asked “How is the process of defining software requirements carried out?” while only those who spoke of requirements analysis were asked “How are the requirements analyzed?” In order to enhance the reliability of the protocol, a practitioner of software development was asked to assess the questions in terms of clarity and understandability and amendments were made where needed.

The interviews were spread apart so as to give room for early analysis that informed further collection of data. A few questions seemed relevant after analyzing some of the data and were therefore added. An example of such questions was “What is your main concern throughout the process of defining software requirements?” which addressed the core of grounded theory in developing a theory to explain a problem from the subjects’ perspectives. Subjects of earlier interviews were contacted via e-mail in order to seek their answers for the additional questions.

3.3. Data Analysis

The literature contains at least three distinct methods that claim the name of grounded theory [10][22][23]. We adopted the classical Glaserian approach[10] using Adolph et al.’s[21] recommendations for applying grounded theory in the context of software engineering. Accordingly, we went through three phases of coding: open coding, selective coding, and theoretical coding. During open coding, interview transcripts were carefully inspected line-by-line for incidents that could be suggested by a word, a line, or a whole paragraph. In addition to fracturing the data, the line-by-line coding helped achieve a grounding of the categories and their properties away from preconceived impressions [20][24]. As incidents were identified, they were labeled with in-vivo codes and rigorously compared to other incidents so that concepts could be generated as suggested by groups of analogous codes. The resulting concepts represented patterns of behavior that were abstractions of time, place, and people [25], and which were later bundled

into categories that formed the building blocks for the theory. As indicators within incoming data were compared to previous ones, concepts were sharpened to achieve a best fit with the data and theoretical properties of a category were discovered. This process is depicted in Figure 1.

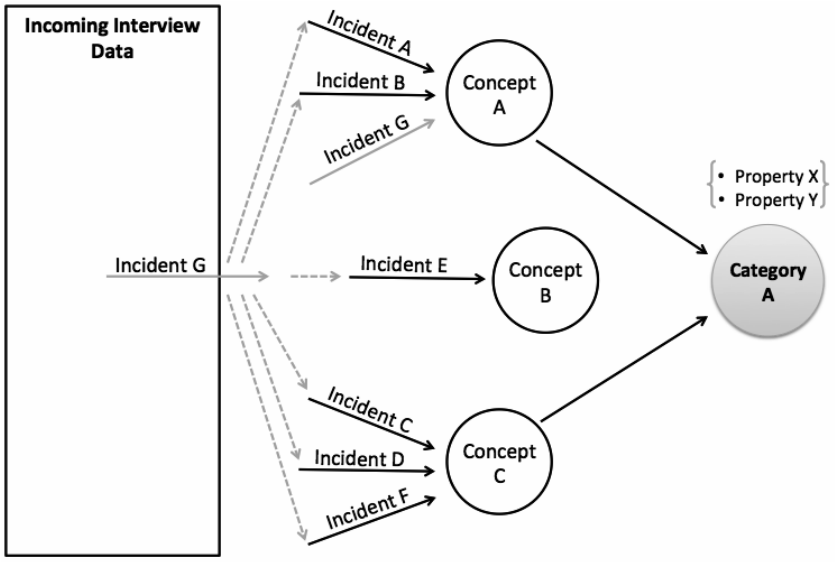

Figure 1 —Classification process of data into concepts and categories.

Memos were heavily utilized in order to capture our ideas for concepts and categories as the interviews were being coded, which is central to the process of theory generation [20]. When it was difficult for us to code certain parts of a transcript, we wrote a summary of the subject's statements, which usually helped us abstract the content and in turn code it. This activity led to the development of separate memos, each summarizing the contents of an interview, which ended up being of great help during cross-interview comparison. Additional memos were created in order to document the analysis process, code definitions, and comments about the interview protocol and required amendments.

Coding the first two interviews produced a large number of concepts that quickly started to shape as prominent or weak patterns by the time the fourth interview was coded. At this point, our memos and concepts of stable patterns were used to inform the process of theoretical sampling by which further data collection was guided. The process continued until the incidents became interchangeable and no additional properties were being introduced to existing categories, at which stage the categories become saturated.

Once all interviews were coded, we reviewed all the concepts and categories in terms of the strength of the patterns they represented, the appropriateness of what was coded under them, and the way they were hierarchically organized. This was done in NVivo software, where all the coding and memo taking had been done so far. Categories of weak patterns or single indicators were dropped out, and statements that seemed improperly coded were amended by either finding them a better fitting category or entirely un-coding them. The concepts were also reorganized into what seemed as a better hierarchy at that point, but which changed after further analysis took place.

Next, all interview summary memos were aggregated into a single document and organized into topics, which facilitated identification of similarities and differences across the interviews. Concept and category contents were reviewed again for additional notes to be added to the

aggregated memos, during which some inappropriate coding instances were further identified and resolved. Reflections upon the categories and how they may relate to formulate a theory took place after that. However, it was not until after we completed writing up the descriptive results that the theory was finalized. We found that summarizing the overall results was the final step required to produce a tight-knit theory that was grounded in the data.

The core category was apparent all along, having a prominent pattern that addressed the subjects' main concern and which interpreted the largest variation in the data. This made the process of selective coding quite straightforward, during which a core category was selected as the center for the emergent theory. Theoretical coding, however, was a little more complicated. During this process, we worked on developing hypotheses about how the different categories related to one another so that they may be integrated into the theory. Graphical representations of the categories and their child concepts proved to be quite helpful, where we went on reorganizing the content until the final theory emerged. The process was guided by frequent reviews of subject statements, which ensured a grounding of the established relationships.

3.4. Reliability

In order to enhance the reliability of the research findings, we reviewed all the transcripts against the audio recordings in order to eliminate any errors that may have taken place during transcription. As mentioned earlier, concept and category contents were regularly reviewed and compared to one another in order to make sure that code definitions were not drifting from what was specified in the code definition memo [26]. To further ensure coding consistency, a panel of five judges were asked to assess the extent to which five randomly chosen subject statements belonged under their respective concepts or categories. The judges included both academics and practitioners of engineering backgrounds. The average score of their assessment was at 88%.

3.5. Validity

The constant comparative method of grounded theory played a key role in naming the concepts and categories by continuously attempting to fit words to the patterns they represented in order to capture their meaning to the fullest. Following exhaustive fitting of words, the name that best depicted a pattern was chosen, at which stage validity was established [25]. In addition, we relied on patterns that were stable across several subjects in order to develop the concepts and categories so that the validity of the results is enhanced. By providing a rich descriptive account of the RE process and the setting under which it took place, we aimed at providing the reader with a sense of a shared experience that in turn is supposed to boost the findings' validity [27].
In addition, an approach similar to that used for enhancing reliability was used to ensure validity, where the same panel of judges was asked to assess the accuracy with which five randomly chosen concepts and/or categories were labeled. The judges were provided with the concept/category labels and their corresponding definitions, which they all agreed to be valid.

4. DESCRIPTIVE RESULTS

This section presents the descriptive results of the study, which are classified according to the conceptual topics that emerged from the gathered interview data.

4.1 The Requirements Engineering Environment

The RE process takes place in a diverse environment that consists of a problem domain and a solution domain. The problem domain, or in other words the customer's business domain, is the

environment where the problem that requires a software solution resides and where that solution later operates. It encompasses problem owners, policies, procedures, and legacy systems. On the other hand, the solution domain represents the environment where the software solution is developed and is comprised of a process model, problem solvers, and solutions.

4.1.1 Problem Domain

X-Solutions works with a variety of problem domains that range from public sector to healthcare, financial, telecommunications, real estate, and retail. A problem domain is described based on industry, location, size of the organization, process complexity, employees, and maturity in terms of using software solutions. All of these domain characteristics factor into how RE proceeds. Participants emphasize the role of the problem domain and the importance of being familiar with it, where a deficiency in this area is seen as a challenge that must be overcome.

Problem owners and legacy systems. Problem owners are represented by the customer's business users, information technology personnel, project managers, and managers. The participants discuss dealing with problem owners that vary in their level of focus upon their needs, willingness to collaborate, and technical maturity, which all affect the process of RE especially in the number of requested requirements change.

For the purpose of this research, a legacy system is defined as an outdated software solution that is in need of replacement. Where present, legacy systems form an important source of knowledge for RE. While some participants view the lack of a legacy system as a benefit since it reduces the chances of user resistance to a new system, others describe having a legacy system as a prerequisite to technical maturity.

4.1.2 Solution Domain

X-Solutions is a highly dynamic domain that typically handles many large projects simultaneously. The participants characterize it as a mature environment that is built on a strong foundation of effective procedures, expertise, and internal knowledge.

Process model. The participants describe having a waterfall process model for software development that includes RE. The RE process begins with scope analysis, which results in a project proposal. When a contract is awarded, the proposal is handed over to the business analysis team, where a RE professional starts preparing for the process, and then proceeds through iterations of requirements elicitation, analysis, documentation, and validation until all requirements are gathered. The focus then shifts towards analysis of the requirements as a whole along with specification, followed by internal review and software requirements specification (SRS) validation. Once the SRS is approved, the RE process continues in a state of requirements and scope management until the final solution is delivered. The process is depicted in Figure 2.

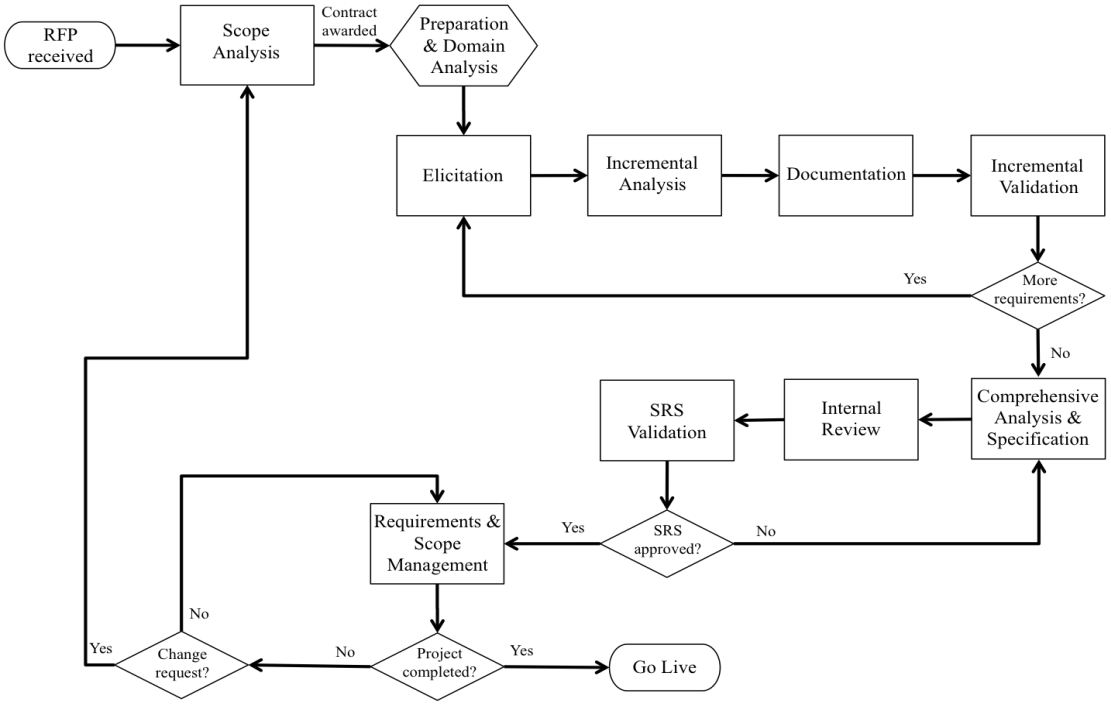

Figure 2 —RE process model.

A distinct aspect of this process is that it iterates over analysis and validation at regular points instead of having them as standalone activities that only take place once during the process. The ongoing effort of analysis and validation ensures that the RE professional is always on the right track and that errors are promptly detected and resolved. The participants label the part of the process that contains elicitation, incremental analysis, documentation, and incremental validation as the "analysis phase" and tend to refer to the activities that it encompasses collectively as "analysis." While the process is well defined in terms of activities and outcomes, the participants describe having room for flexibility to accommodate the different project circumstances.

Problem solvers. Problem solvers are individuals with roles that affect the software development process. For the purpose of RE, problem solvers include RE professionals, project managers, and members of the presales, quality control, and development teams. The presales team is responsible for defining the project scope, developing the project proposal, and securing the customer's approval. This is succeeded by RE professionals who carry out all RE activities leading up to the SRS handover to the development team, all the while managing the project scope. Project managers then take over the requirements and scope management after obtaining the SRS sign-off. The quality control team reviews the SRS to confirm compliance with their quality standards, and the development team does the same to confirm technical feasibility.

According to the participants, having a technical background is essential for developing a system's perspective—that is to understand the boundaries of the solution being implemented, and hence better analyze and negotiate requirements. Moreover, the participants find it vital for them to have good interpersonal skills in order to manage customer discussions and elicit full requirements. When asked about causes for insufficient requirements, the participants mostly pointed to shortcomings on the part of the RE professional rather than the customers. RE professionals seem to carry the burden of defining complete and accurate requirements under any circumstances and willingly take the blame otherwise.

Developers are in charge of implementing the solution, which does not begin until an SRS is finalized. Unlike RE professionals, developers are not expected to be familiar with the problem domain. It is the role of a RE professional to mediate between the problem domain and the developers by eliciting complete business requirements from the customer and meticulously documenting them so that the developers can easily apply them to a solution. As opposed to the business-oriented RE professional who strives for customer satisfaction, developers are described as being technically orientated with concerns regarding the efforts that they need to put towards completing their jobs.

Solutions. The participants work on custom software solutions and off-the-shelf customer relationship management (CRM) and enterprise resource planning (ERP) software. The type of solution to be developed affects how the RE process proceeds in terms of activities and tools to be used.

4.2 The Requirements Engineering Process

4.2.1 Scope Analysis

The first phase in RE begins once a request for proposal (RFP) is received, where the presales team works on delineating the scope of the solution to be developed based on a comprehensive RFP (if available) or through meeting with the customer. The result of this step is then integrated into a demo that is presented to the customer. Once the scope is approved, a proposal is created. By signing the proposal, the customer agrees to the scope as defined in the document and acknowledges that future changes will go through a change request. Once the proposal is signed off, a handover meeting that gathers all project teams takes place.

4.2.2 Preparation and Domain Analysis

The RE professional's role starts with this activity, where they study the scope as defined in the proposal and dissect it into main sections that are used in stakeholder analysis, i.e. identifying the roles of the people required during RE and soliciting their names. The key part of preparation is domain analysis, which involves an investigation into the customer's business domain, business processes, policies, software solutions, technology, business users, and organizational chart. In the case of a custom solution, the RE professional performs benchmarking and uses the gained insights during requirements elicitation and analysis. Other exercises that may take place during this phase include preparation of tentative requirements documentation and/or prototypes.

4.2.3 Elicitation, Incremental Analysis, Documentation, and Incremental Validation

The requirements are elicited, analyzed, documented, and validated using a piecemeal approach, which closely knits the four activities in this process. The first step involves holding a meeting with all customer stakeholders in order to validate the scope and understand how processes of the different departments affect each other. The focus then shifts towards elicitation, which is conducted through series of workshops with individual departments in order to gather detailed requirements. One-on-one interviews are sometimes utilized as well, and employee observation may be conducted upon customer request. It is disfavored, however, since the approach is found to be ineffective at capturing complete requirements

The participants stress having a proper combination of business users and decision makers during the elicitation workshops so as to facilitate simultaneous access to accurate requirements and

decision making, which increases process efficiency. To further increase productivity, the RE professional may sometimes utilize prototypes or initiate elicitation by preparing questionnaires based on their analysis of the proposal and customer-provided documentation. These are shared with the customer before holding the workshops, which allows them the opportunity to think through the questions before addressing them.

Requirements analysis is an ongoing effort of eliminating discrepancies in a complete and interconnected set of requirements. This involves breaking down general business requirements into specific technical ones, establishing logical connections, identifying gaps, and making tradeoffs between the different requirements. The effort is done regularly upon acquiring new knowledge from the customer, whether through meetings or written documents, in order to clarify ambiguities early on and guide further elicitation.

The RE professional develops meeting minutes following every elicitation meeting in order to document the gathered requirements, which must be validated and approved by the customer before conducting the next meeting. In addition to ensuring swift detection and correction of errors, the approach helps establish gradual customer commitment to the requirements. Once elicitation is completed, the approved minutes are consolidated and presented to all customer stakeholders for validation.

4.2.4 Comprehensive Analysis and Specification

Once elicitation concludes and the customer approves the full set of gathered requirements, more analysis takes place along with specification. Some of the participants find that it is not until this stage, when the requirements are all put together to form a complete story, that requirements gaps are detected. Therefore, they emphasize close inspection of the requirements to make sure they are consistent and compete, which requires a strong attention to detail.

The SRS is described as a hybrid document that covers aspects of SRS and business requirements specifications, and which is documented with the help of templates that are tailored to each project. The participants are conscious of all the different perspectives that the SRS will be read from and accordingly create a document that is readable by all these entities and which satisfies all their needs. As for document quality, most of the participants view it as a reflection of the RE professional's experience and efforts, and hence take full responsibility for developing an SRS that proceeds through internal review and validation with minimal change.

4.2.5 Internal Review

The present phase puts the SRS through multiple levels of review, starting with the business analysis team peer-review, which is only utilized for larger projects or with less experienced RE professionals. Next, the quality control team reviews the document to confirm that it abides by their quality standards and that the requirements are comprehensive and within scope. The participants describe this review as highly detailed and effective in the sense that most SRSs pass through customer validation with little change and are later implemented with minimal interruptions. The development team concludes the document reviews to confirm technical feasibility of all requirements. ERP products have a less elaborate review process since they primarily rely on existing functionalities and features.

4.2.6 SRS Validation

As mentioned earlier, validation is not a discrete activity but rather an ongoing effort. The purpose of previous validations was to confirm proper understanding of the requirements as they

were gathered, which is re-established during SRS validation. Where changes are required, the updates are reviewed with the project manager and the development team to confirm they are within scope and technically feasible before being documented into the SRS. Once amended, the document is sent to the quality control team for another review before being sent back to the customer for validation.

4.2.7 Requirements and Scope Management

While requirements management does not take place until after an SRS is approved, scope management starts along with elicitation so as to avoid the occurrence of scope creep at any point in the process. Participants explain that specific project and change request circumstances dictate how requirements are managed. In general, changes are not allowed past SRS approval unless put through a change request process, which initiates a new RE cycle for the change. However, a change request is usually overlooked for minor changes that have minimal impact on the ongoing implementation, which are frequently requested once a full solution design is available. It is unclear how these changes are documented though, if ever. Changes that are within scope are always accepted no matter how complex they are. Such changes are rare, however, and so are change requests in general. The participants credit this to the thorough elicitation and to the extensive validation done throughout the process.

4.3 Challenges

4.3.1 Uncertainty

The participants speak of uncertainty during RE from various perspectives depending on the source and the impact of the issue. The first source of uncertainty is the customer, which takes many forms starting with uncertainty of their needs. In order to deal with this, the participants invert the rule of listening more than talking and rely on suggestions rather than questions in extracting requirements during elicitation. They admit, however, that it is difficult to rely on this approach when developing custom solutions to automate business processes. The challenge is further compounded when the uncertainty continues through the results, where the customer refuses to accept the final solution for no obvious reason.

Participant 6: “We did all what they wanted and we did several testing phases and they’re ok with it… a disaster, because he didn’t know what he wanted from the beginning, and at the end he didn’t know whether he got it or not.”

Customer-induced uncertainty may also materialize in incorrect or missing information. While some customers drop information unconsciously or for deeming it irrelevant, others deliberately hide troublesome aspects of existing processes in fear that it may restrict the new solution. Some RE professionals rely on intuition in detecting missing information and others are able to identify it through conflicting accounts of existing business processes provided by different users. However, they explain that figuring out the absence of information is not always a straightforward endeavor, let alone retrieving it. Continued engagement, repeated probing, and building trust are among the approaches used to battle this issue.

Uncertainty may also arise when the customer assigns inappropriate entities to the RE effort, where the RE professional develops doubts about working with the right people. Experience plays an important role in detecting improper customer stakeholders given indications such as unreasonable requirements or requirements that largely deviate from the scope as specified in the proposal.

Not being able to visualize the requirements as a software solution creates a form of uncertainty for the customer with regards to the end product, which can result in missing requirements. This is usually handled with the use of prototypes, where customers are better able to specify their needs. Uncertainty may also arise in projects with new business domains or technologies.

4.3.2 Resistance

Most of the participants report customer resistance to a new solution as one of their toughest challenges, which they attribute to one of two reasons: having a bad experience with a legacy system or fearing a solution would replace them. The resistance is exhibited in stakeholders withholding information and being uncooperative during elicitation. Having strong interpersonal skills is key in dealing with customer resistance, where trust is built through effective communication and demonstration of proper domain knowledge and successful results in early phases of a project. However, the participants explain that such resistance is never eliminated but only mitigated. When all efforts fail to resolve the situation, the RE professional tries to seek an alternative customer representative or escalate the issue. Another form of resistance may materialize when customers who are not pressed for time refuse to approve an SRS in order to gain more room for changes.

4.3.3 Conflict

Conflict among customer stakeholders may occur across different departments or between stakeholders of the same department, which complicates defining a single set of requirements to satisfy all entities. The participants emphasize gathering all customer stakeholders for an initial elicitation workshop to discuss their high-level requirements and identify conflict between them. This provides the conflicting parties a chance to explore opportunities for agreement while the RE professional tries to moderate the discussion. When negotiation fails, the issue may be escalated.

4.3.4 Stakeholder Availability

In addition to representing a challenge of its own, stakeholder availability tends to cause other difficulties such as uncertainty, conflict, time delays, and requirements changes. The challenge is exhibited in stakeholders that are changed during RE, unavailable for consultation beyond elicitation, or unavailable altogether when inappropriate entities substitute them. The last form tends to occur with decision makers, where mediators take their place during RE and hence the process becomes very lengthy.

4.3.5 Uncooperative Stakeholders

Completing the RE effort within time constraints given uncooperative stakeholders represents a challenge to RE professionals. It sometimes materializes during initial project handover, where presales members conduct insufficient handovers that lack the transfer of essential background information to RE professionals. The challenge may also occur when the development team is late in responding to a consultation regarding the technical feasibility of a certain requirement, which delays the progress of RE. Some customer stakeholders may likewise be uncooperative by not responding to communication attempts and/or requests, or not showing up for scheduled meetings. Patience is important for dealing with such entities, in addition to having effective communication skills to absorb conflicts and move forward.

4.4 Perceptual Measures of Effectiveness and the Significance of the Requirements Engineering Process

In order to understand how RE affects project success, it is important to learn about how project success is measured. All the participants view customer satisfaction as the ultimate indication for a successful project. While some depict customer satisfaction as the main acceptance measure of the final solution, others stress that it must meet their original needs, add value to them, and be used by end users. As for the RE process, the participants view it as the foundation for the software development endeavor, where all efforts exerted afterwards would amount to nothing if built upon poor RE. This conclusion sheds light on how critical the role of a RE professional is, where any shortcoming on their part increases the chances for project failure.

5. THEORY OF REQUIREMENTS ENGINEERING

The information obtained through this study reveals that the RE process is one of establishing a match through discovery and streamlining, and which utilizes corrective measures in order to manage threats to establishing a match. The theoretical model that emerged from the data for this process is depicted in Figure 3.

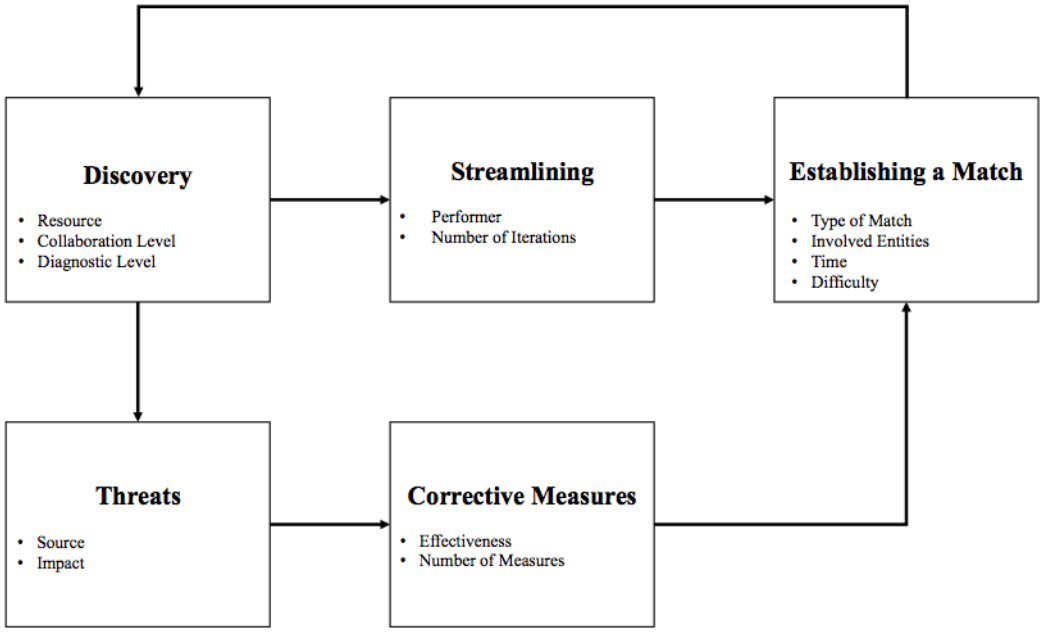

Figure 3—RE process of establishing a match.

The RE process illustrated in Figure 3 starts with a requirements discovery phase followed by a streamlining phase before a match can be established. Sometimes, mismatch threats emerge while discovering the requirements and must be handled through corrective measures in order to proceed. After establishing a match, the process is repeated until the final match between customer needs and the SRS is established. This model is based on the categories and concepts generated during data analysis which are summarized in Figure 4.

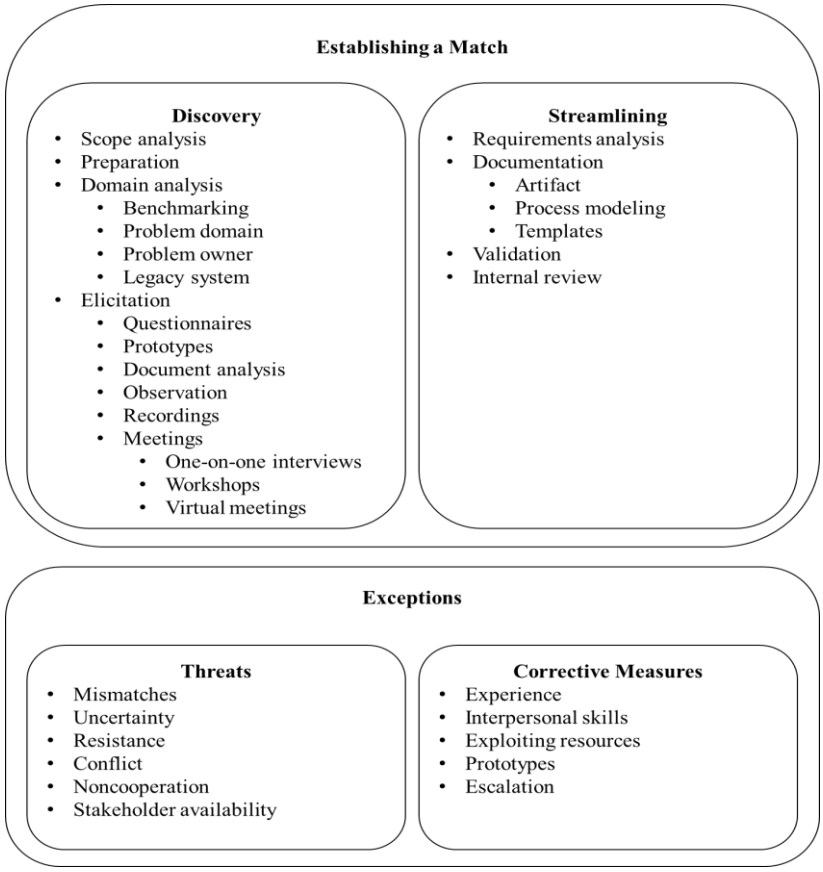

Figure 4 —Data analysis categories and concepts.

The basic social process that emerges from the data and which explains the patterns of behavior that are fundamental to RE is the process of establishing a match. At every point in time during this process, problem solvers are seeking a match between various aspects among diverse groups of problem owners and problem solvers in order to pave the road for the match between customer needs and the SRS artifact.

The process has four distinct properties that are defined as follows:

Type of match, which reflects the elements that have been matched. These include: customer needs, scope, requirements, quality standards, technical feasibility, and undocumented knowledge.

Involved entities, which represents the entities among which a match has been established, who range from presales members to RE professionals, quality control members, development members, project managers, and customer stakeholders.

Time needed to reach a match.

Difficulty of reaching a match.

When describing their main concern throughout the process of RE, the participants speak of varying concerns that in large reflect threats to establishing a match. The threats are nothing but occasions of mismatch, which constitute the core challenge that problem solvers are trying to

overcome during this process. The participants' concerns include: uncertainty with regards to how accurate the defined requirements are and their relevance to customer needs; mismatches depicted in scope creep, requirements gaps, and changes during implementation; stakeholder availability represented in identifying proper stakeholders and stakeholder changes; and resistance to approve the artifact. In turn, the participants' main goals during this process converge into establishing a match by defining accurate and comprehensive requirements to satisfy customer needs and add value to them, all the while controlling scope and streamlining in order to go live within the project's time frame.The effectiveness of the process is determined by how well of a match is established between customer needs and the SRS artifact, and eventually with the delivered solution.

The process of establishing a match has two emergent phases: discovery and streamlining. During discovery, problem solvers use a mixture of activities and techniques towards exploring problem owners' needs. The effectiveness of the process relies in large on the experience of the problem solvers. The process has three properties:

Resource, which represents the source of knowledge for the process, including problem owners, problem domain, problem solvers, solution domain, the internet, and other elements of the environment.

Collaboration level, which describes whether discovery is being done on an individual level, dual levels, or multiple levels.

Diagnostic level, which represents the extent to which the process of discovery involves diagnostic analysis as opposed to descriptive analysis.

The initial phase of discovery is concerned with scope analysis, which continues until a match of the scope is established among the problem owners and problem solvers. Once the match is reflected in an artifact (i.e. proposal), discovery shifts towards domain analysis. During this activity, the problem solver accumulates a knowledgebase about the problem domain, problem owners, and the legacy system. This knowledge is further expanded by insights of benchmarking and is used to convey a sense of match between the problem solver and the problem owners in terms of domain knowledge—an element that is essential for effective discovery.

Domain analysis continues through elicitation, during which the problem solver explores detailed problem owner needs using several techniques that are selected based on the characteristics of the problem owner and the solution to be developed. These techniques include: meetings, questionnaires, prototypes, document analysis, recordings, and observation. Elicitation makes up the bulk of discovery, where iterations of elicitation-based discovery continue until a match is established between problem owners' needs and the SRS artifact.

In addition to experience, problem solvers rely on exploiting resources in order to conduct effective discovery. Instances of exploiting resources include seeking problem domain or solution knowledge from fellow problem solvers and previously developed solutions and artifacts in addition to utilizing solutions as prototypes in order to facilitate elicitation.

Occasionally, threats materialize during discovery, which may hinder the progress towards establishing a match unless a corrective measure is taken. As mentioned earlier, threats represent occasions for a mismatch that include uncertainty, conflict, resistance, noncooperation, stakeholder availability, and mismatches. The category has two properties:

Source, which represents the origin of the threat being one of the following: problems owners, problem domain, or problem solvers.
Impact, which describe how difficult it is to proceed with discovery under the threat.

The majority of threats are socially constructed, and therefore corrective measures are rooted in a problem solver's interpersonal skills that have been polished through experience. Some instances are further managed using prototypes or by exploiting resources, such as those involving uncertainty. When all measures fail at mitigating a threat, the problem solver usually resorts to escalation. Properties of corrective measures are:

Effectiveness, which describes the extent to which a threat is eliminated by a measure.

Number of measures taken towards eliminating a threat.

Streamlining utilizes a combination of analysis, validation, and review in order to eliminate mismatches as they appear so as to minimize their impact on establishing a match, where the longer they endure the higher the impact becomes. The process has two properties:

Performers, which represents the problem solvers in charge of a given phase.

Number of iterations required to reach establishing a match.

Once a discovery iteration is completed, the problem solver uses analysis in order identify mismatches in the content discovered and resolutions for eliminating them. Next, the output of analysis goes through validation, where a problem owner reviews the content discovered and provides their recommendations for updates where needed. Iterations of streamlining continue until the problem owner confirms the content, which marks the completion of another "establishing a match" iteration, during which the approval is reflected in an artifact.

After the last discovery iteration, streamlining utilizes review before validation during the final phase, which puts the artifact that combines all the content discovered under heavy scrutiny by problem solvers. This activity ensures that the contents of the artifact abide by the solution domain's standards and the solution's boundaries. After that, the artifact goes through validation, and where updates are necessary, the updated artifact goes back to review before proceeding to validation once again. As described above, streamlining iterates until a problem owner confirms the contents of this artifact, which marks the arrival at establishing a match between problem owner needs and the SRS artifact. Requirements management and scope management are utilized beyond this in order to maintain this match until a final solution is delivered.

6. DISCUSSION

When comparing our account of the RE process model (see Figure 2) to what is published in the literature [4][28], we find that it includes all the basic activities of requirements elicitation, analysis, documentation, validation, and management. However, our results indicate that these activities have an iterative nature, which is an aspect that is not emphasized in the literature. Additionally, whereas requirements validation is described as occurring after an SRS has been developed, we find that it starts much earlier in the process so that a gradual validation of the requirements takes place. This approach seems to be one of the strengths of this process model, where it helps establish a cumulative agreement to the requirements, all the while detecting requirements errors and resolving them as they occur. Furthermore, our results introduce three new activities to the existing RE process model: scope analysis, domain analysis, and internal review.

Our results reveal a special emphasis on domain analysis, where an understanding of the problem domain is viewed as a prerequisite to effective RE. This notion is consistent with much of what has been published in the literature about the vitality of accounting for the problem domain in RE [29][30]. Therefore, we think that the RE community could indeed find it valuable to focus more on the topic of domain analysis [31] and the techniques by which such problem domain knowledge may be extracted [30].

When it comes to the selection of RE techniques, we find that RE professionals rely on a set of project characteristics in order to select proper elicitation techniques, which include the customer's technical maturity. Earlier efforts to explore the selection of RE techniques, such as Jiang et al. [32], considered various project properties that excluded this element. Our findings are consistent with the work of Hadad, Doorn, and Ledesma[33],who encourage taking such situational factors into account while planning RE activities at different milestones. Our results reveal that workshops, interviews, and prototypes are among the most commonly used elicitation techniques, which supports the findings of Wagner et al. [34]. It is also in line with Pacheco, García, & Reyes' work [35], which reports these techniques as demonstrating effectiveness across many studies.

Regarding the challenges that face RE professionals, our results don't show any challenges that are described as drivers for project failure. Rather, they are viewed as normal work challenges that are generally manageable. On the other hand, we find deficient RE to represent a potential cause for unsuccessful projects. Interestingly, our findings suggest that the bulk (if not all) of challenges facing RE professionals are socially constructed and that classic difficulties such as project complexity and timeframe are not prominent. While such an observation is specific to the context of the study and is not generalizable, it still counts as support to the literature arguing for the need to better address the human aspect of the RE process[36]. Based on our findings, the RE community may indeed find it valuable to collaborate with scholars from different disciplines as suggested by Cheng and Atlee [37] and Lenberg et al. [38] so as to provide guidance to RE professionals on how to overcome these social challenges. Furthermore, our account of RE seems to be effective despite relying on nothing but classic RE methods. On the contrary, sources of process effectiveness are rooted in an outlined process that has room for flexibility; clear roles and responsibilities; and seamless teamwork. Again, while this remark may not be generalizable, it nonetheless provides an opportunity for reflection on the merit of developing new RE techniques to add to the rich literature on RE methods.

Our account of requirements uncertainty is closest to Davis' [39] concept of overall requirements process uncertainty. Instances of uncertainty caused by missing or incorrect requirements, customers being unsure of their needs, and customers' inability to imagine requirements as implemented in a software solution mirror Davis's measures of users' ability to specify requirements and the existence/stability of a set of usable requirements. Moreover, our observation of uncertainty with regards to a business domain or a technology is consistent with Davis's third measure of analysts' ability to elicit and evaluate requirements. We also find that RE professionals rely on their experience in extracting tacit knowledge using a process of continuous probing and revision. As part of their effective attitude towards their role, they take full responsibility for defining and specifying complete requirements under any circumstance. This finding is inline with the results of Fernández et al. [40], which hold that RE professionals' weak qualification and lack of experience are among the causes for incomplete and/or hidden requirements. Also, it highlights the importance of Burnay et al.'s [41] urges to prepare elicitation interview questions so that implicit requirements may be better captured. While some authors argue that such requirements are best identified using ethnographic methods [42][43][44], our results reveal that observation is rather time-consuming and ineffective at gathering complete

requirements. This comes as a surprise since these results are primarily based on projects of automating business processes that depend in large on extracting information pertaining to how people conduct their jobs – an element that is claimed to constitute a tacit component. This may suggest that ethnographic methods are not of universal value in extracting tacit knowledge, but rather are suitable for certain types of projects such as the development of highly complex industrial software.

Coughlan et al.'s [45] framework on communication problems dedicates one of its four dimensions to stakeholder selection and participation. It suggests that RE professionals are at a loss when it comes to identifying and accessing ideal stakeholders in the absence of a clear guiding process. The issue is further compounded when such ideal candidates are unavailable, which stagnates the RE process. Our findings are consistent with these observations, where stakeholder identification, access, and availability are reported as challenges to effective RE.
Differences within and between the problem and the solution domain are the source of diversity in the environment of software development, and while this has long been cited as a challenge to effective communication during RE [46][47], we find X-Solutions to rather embrace it and use it to their advantage. During requirements elicitation, an emphasis is put on having a diverse group of customer stakeholders in order to ensure gathering complete requirements. The same strategy is used towards requirements validation with the customer, in addition to having the multiple levels of internal review with diverse teams. All our participants speak of having SRS documents that are readable and understandable by all entities involved, pointing to the effectiveness of the RE professional and the quality control team in accomplishing this. On the level of the RE professional, our participants acknowledge the diversity among the people they deal with and carry the burden of working through such an environment instead of throwing the blame on others when part of their job goes wrong. Given that X-Solutions is an international company with extensive operations outside the country where their main headquarters are, it is important to note that their approach to diversity seems to work even across varying cultures. This suggests that it may be possible for organizations to deal with diversity by acknowledging its presence, viewing it as an opportunity rather than a challenge, and having a RE professional that is flexible enough to deal with it.

Among the most important success criteria that our findings propose is having a dedicated role for carrying out the activities of RE and selecting appropriate employees to fulfill this role, which sheds light on the merit of exploring competencies that are needed by RE professionals [48]. Our results suggest that ideal candidates would have excellent interpersonal skills that are supported by a sufficient level of experience and an orientation towards details and customer satisfaction. As argued by many authors and supported by our results, software engineers have a natural inclination towards being overly technical, which often results in ineffective communication with customers [49]. Therefore, the model of separating RE responsibilities from those of the developers' seems to allow for better communication, which is essential for defining accurate software requirements. Additionally, it permits both RE professionals and developers to concentrate on their areas of competency, whereby they are able to produce quality outcomes. These findings are consistent with Klendauer et al.[50], who noted significant levels of project success when a RE professional role was introduced to organizations. However, the success was contingent upon the RE professional having a delicate balance of analytical and interpersonal skills, where otherwise the role could be obstructive. The authors further discovered that advanced RE techniques were not of significance to their interviewees, which is something that we also note in our study.

In terms of the emergent theory of RE, we are happily surprised to find that it carries similar themes to Adolph et al.'s [11] grounded theory of the software development process. The authors discovered that people use a process of reconciling perspectives in order to manage the process of

software development and eliminate perspective mismatch, which impeded getting the job done if not resolved. They described the process of reconciling perspectives as having the two stages of converging and validating, during which people use negotiation in order to establish a shared consensual perspective that is later validated. We find the concept of reconciling perspectives to be consistent with establishing a match, where both are used to resolve the problem of mismatch. We acknowledge, however, that the scope of the mismatch may vary across the theories. Moreover, the reconciling perspectives' stages of converging and validating seem to have a similar goal to our stage of streamlining, which aims at identifying and resolving mismatches in order to reach an agreement and later validate it. As is the case with the process of establishing a match, reconciling perspectives is described as an iterative process that continues to repeat until a perspective mismatch is fully reconciled.

Our account of the people's main concern during the process is different from Adolph et al.'s, however. The main concern of those involved in the process of software development was identified as getting the job done by delivering a working product that appeases the customer. A reduced concern about customer appeasement was found in teams that had no direct interaction with the customers, where getting the job done was viewed in terms of technical craftsmanship. RE professionals, on the other hand, are primarily concerned about occasions of a mismatch that may not necessarily exhibit an impact until after their duties are completed. While Adolph et al.'s findings are in line with what we observe about software developers, we find RE professionals to be highly concerned about customer satisfaction that may be impeded by occasions of mismatch. This distinction sheds the light once more on the merit of having a dedicated role for RE that is fulfilled by individuals with a customer satisfaction priority that supersedes getting their jobs done.

When viewed within the context of the RE process being part of the software development process, consistencies between our theory and Adolp et al.'s provide strength to our results and to the merit of using grounded theory to study topics of software engineering, which is yet to be exploited.

Our results provide several interesting insights into the practice of RE. However, these findings should be reviewed through the lens of their limitation, which confines them to the context of the study. Nevertheless, they provide the RE community with several thought-provoking observations that could represent a starting point for those seeking to further understand the RE process. We encourage researchers to test these results under various settings and to compare them to the results of similar qualitative endeavors.

7. SUMMARY

This paper uses grounded theory methods to devise a theory of RE. The theory is based on the experience of practicing professionals and aims to bridge the gap between the academic literature and commercial practices. As such, the presented RE theory has value in both academia and industry.

To build on the academic literature, this paper provides researchers an account of how RE is carried out in industry with details on the involved activities, transition between process stages, and work iterations. It provides the concerns of professional practitioners and their methods in resolving difficulties. This paper presents practices that have not been emphasized in previous literature and can be built upon by researchers interested in RE. Although grounded theory is not popular in the software engineering discipline, this paper demonstrates its merit as certain insights could not have been discovered otherwise.

Practitioners in the industry often rely on professional training and experience to perform their jobs. This paper summarizes the experience and challenges of professionals carrying out an effective RE process with numerous lessons that can benefit practitioners. Some of the key takeaways for practitioners from this paper include the implications of establishing a match for the RE process, concerns about mismatches between customer needs and defined requirements, and the importance of acquiring domain knowledge. This paper also touches on elaborative processes that are close to commercial organizations such as internal review, which may also be of interest to practitioners.

REFERENCES

[1] Brooks, FP (1987) "No silver bullet: Essence and accidents of software engineering", *Computers*, Vol. 20, No. 4, pp10-19. https://doi.org/10.1109/MC.1987.1663532

[2] Nuseibeh B & Easterbrook S (2000) "Requirements engineering: A roadmap", *Conference on the Future of Software Engineering*,pp35-46. https://doi.org/10.1145/336512.336523

[3] Zave P (1997) "Classification of research efforts in requirements engineering", *ACM Computing Surveys*, Vol. 29, No. 4, pp315-321. https://doi.org/10.1145/267580.267581

[4] Institute of Electrical and Electronics Engineers (IEEE) (2014) *Guide to Software Engineering Body of Knowledge (SWEBOK)*, version 3.0. http://www.swebok.org

[5] Kotonya G & Sommerville I (1998) *Requirements engineering: Processes and technique*, John Wiley & Sons, Inc, Hoboken, NJ.

[6] Jantunen S & Gause DC (2014) "Using a grounded theory approach for exploring software product management challenge",*Journal of Systems and Software*,Vol. 95,pp32-51. https://doi.org/10.1016/j.jss.2014.03.050

[7] Hinds C (2008) "The case against a positivist philosophy of requirements engineering", *Requirements Engineering*, Vol. 13, No. 4, pp315-328. http://dx.doi.org/10.1007/s00766-008-0061-9

[8] Robey D, Ross JW, Boudreau MC (2002) Learning to implement enterprise systems: An exploratory study of the dialectics of change. Journal of Management Information Systems 19(1): 17-46. Retrieved from http://www.coba.unr.edu/faculty/kuechler/788/learningtoimplementerpchangemetaanalysis.pdf

[9] Mohr LB (1982) *Explaining organizational behavior: The limits and possibilities of theory and research*. Jossey-Bass, San Francisco, CA.

[10] Glaser BG& Strauss A (1967) *The discovery of grounded theory: Strategies for qualitative research*. Aldine, Chicago, IL.

[11] Adolph S, Kruchten P, &Hall W (2012) "Reconciling perspectives: A grounded theory of how people manage the process of software development", *Journal of Systems and Software*, Vol. 85, No. 6,pp1269-1286. http://dx.doi.org/ 10.1016/j.jss.2012.01.059

[12] Chakraborty S, Sarker S, & Sarker S (2010) "An exploration into the process of requirements elicitation: A grounded approach", *Journal of the Association for Information Systems*, Vol. 11, No.4,pp1. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.187.4915&rep=rep1&type=pdf

[13] Coleman GO'Connor R (2007) "Using grounded theory to understand software process improvement: A study of Irish software product companies", *Information and Software Technology*, Vol. 49, No. 6, pp 654-667. http://dx.doi.org/10.1016/j.infsof.2007.02.011

[14] Crabtree CA, Seaman CB & Norcio AF (2009, October) "Exploring language in software process elicitation: A grounded theory approach", In *3rd International Symposium on Empirical Software Engineering and Measurement*, pp. 324-335. IEEE.

[15] Daneva M, Van Der Veen E, Amrit C, Ghaisas S, Sikkel K, Kumar R, Ajmeri N, Ramteerthkar U &Wieringa R (2013) "Agile requirements prioritization in large-scale outsourced system projects: An empirical study", *Journal of Systems and Software*, Vol. 86, No. 5, pp1333-1353. http://dx.doi.org/10.1016/j.jss.2012.12.046

[16] Hansen BH& Kautz K (2005) "Grounded theory applied-studying information systems development methodologies in practice", The 38th*Annual Hawaii International Conference on System Sciences*, pp264b-264b. http://dx.doi.org/10.1109/HICSS.2005.289

[17] Hoda R, Noble J& Marshall S (2012) "Developing a grounded theory to explain the practices of self-organizing agile teams", *Empirical Software Engineering*, Vol. 17, No. 6, pp609-639. http://dx.doi.org/10.1007/s10664-011-9161-0

[18] Sorrentino M& Virili F (2005) "Web services system development: A grounded theory study", *BLED 2005 Proceedings*. Retrieved from http://aisel.aisnet.org/bled2005/51

[19] Würfel D, Lutz R& Diehl S (2016) "Grounded requirements engineering: An approach to use case driven requirements engineering", *Journal of Systems and Software*, Vol. 117,pp645–657. http://doi.org/10.1016/j.jss.2015.10.024

[20] Glaser BG (1978) *Theoretical sensitivity: Advances in the methodology of grounded theory*. Sociology Press, Mill Valley, CA.

[21] Adolph S& Hall W, Kruchten P (2011) "Using grounded theory to study the experience of software development", Empirical Software Engineering, Vol. 16, No. 4,pp487-513. http://dx.doi.org/10.1007/s10664-010-9152-6

[22] Strauss A& Corbin JM (1990) *Basics of qualitative research: Grounded theory procedures and techniques*. Sage Publications, London.

[23] Charmaz K (2006) *Constructing grounded theory: A practical guide through qualitative analysis*. Sage Publications, London.

[24] Glaser BG (1992) *Basics of grounded theory analysis*. Sociology Press, Mill Valley, CA.

[25] Glaser BG (2001) *The grounded theory perspective: Conceptualization contrasted with description*. Sociology Press, Mill Valley, CA.

[26] Gibbs G (2007) *Analyzing Qualitative Data*. In U Flick (Ed.) The Sage qualitative research kit. Sage, London.

[27] Creswell JW (2009) *Research design: Qualitative, quantitative, and mixed methods approaches (3rd Ed.)*. Sage, Thousand Oaks, CA.

[28] Laplante, P (2018) *Requirements engineering for software and systems*, CRC Press, Boca Raton, FL.

[29] International Organization for Standardization (ISO), International Electro technical Commission (IEC), Institute of Electrical and Electronics Engineers (IEEE) (2010*) ISO/IEC/IEEE 24765: Systems and Software Engineering – Vocabulary*.

[30] Singh, M & Chopra, A (2019). Requirements engineering as science in the small. In Proceedings of the *41st International Conference on Software Engineering: New Ideas and Emerging Results*, pp45-48.

[31] Lengyel L, Mészáros T, Asztalos M, Boros P, Máté A, Madács G, Hudák P, Kovács K, Tresch A & Charaf H (2015) "Quality assured model-driven requirements engineering and software development", Computer Journal, Vol. 58, No. 11,pp3171–3186. http://doi.org/10.1093/comjnl/bxv051

[32] Jiang L, Eberlein A& Far, BH (2008) "A case study validation of a knowledge-based approach for the selection of requirements engineering techniques", *Requirements Engineering*, Vol. 13, No. 2,pp117-146. https://doi.org/10.1007/s00766-007-0060-2

[33] Hadad, G, Doorn, J & Ledesma, V (2019). Dynamic Situational Adaptation of a Requirements Engineering Process. In *Advanced Methodologies and Technologies in Network Architecture, Mobile Computing, and Data Analytics*, pp1386-1399.

[34] Wagner, S, Fernández, D, Felderer, M, Vetrò, A, Kalinowski, M, Wieringa, R, ... & Lassenius, C (2019). Status quo in requirements engineering: A theory and a global family of surveys. *ACM Transactions on Software Engineering and Methodology*, Vol. 28, No. 2, pp9.

[35] Pacheco, C, García, I & Reyes, M. (2018). Requirements elicitation techniques: a systematic literature review based on the maturity of the techniques. *IET Software*, Vol. 12, No. 4, pp365-378.

[36] Sánchez, E & Macías, J (2019). A set of prescribed activities for enhancing requirements engineering in the development of usable e-Government applications. *Requirements Engineering*, Vol.24, No. 2, pp181-203.

[37] Cheng, BH& Atlee, JM (2007)"Research directions in requirements engineering", Proceeding of *2007 Future of Software Engineering (FOSE '07)*,pp285–303. http://doi.org/10.1109/FOSE.2007.17

[38] Lenberg P, Feldt R& Wallgren LG (2015) "Behavioral software engineering: A definition and systematic literature review", *Journal of Systems and Software*, Vol. 107,pp15–37. http://doi.org/10.1016/j.jss.2015.04.084

[39] Davis GB (1982) "Strategies for information requirements determination", *IBM Systems Journal*,Vol. 21, No. 1,pp4-30. http://dx.doi.org/10.1147/sj.211.0004

[40] Fernández DM, Wagner S, Kalinowski M, Felderer M, Mafra P, Vetrò A, Conte T, Christiansson MT, Greer D, Lassenius C & Männistö T (2017) "Naming the pain in requirements engineering", *Empirical software engineering*, Vol. 22, No. 5,pp2298-338. http://dx.doi.org/10.1007/s10664-016-9451-7

[41] Burnay C, Jureta IJ & Faulkner S (2014) "What stakeholders will or will not say: A theoretical and empirical study of topic importance in requirements engineering elicitation interviews", *Information Systems*, Vol. 46,pp61-81. http://dx.doi.org/ 10.1016/j.is.2014.05.006

[42] Goguen JA (1996) Formality and informality in requirements engineering. *Second International Conference on Requirements Engineering*,pp102–108. http://doi.ieeecomputersociety.org/10.1109/ICRE.1996.10005

[43] Halaweh M (2012) "Using grounded theory as a method for system requirements analysis", *Journal of Information Systems and Technology Management*,Vol. 9, No. 1,pp23-38. http://dx.doi.org/10.4301/S1807-17752012000100002

[44] Martin D & Sommerville I (2004) "Patterns of cooperative interaction: Linking ethno methodology and design", *ACM Transactions on Computer-Human Interaction*, Vol. 11, No. 1,pp59-89. https://doi.org/10.1145/986253.986260

[45] Coughlan J, Lycett M & Macredie RD (2003) "Communication issues in requirements elicitation: A content analysis of stakeholder experiences" ,*Information and Software Technology*, Vol. 45, No. 8,pp525-537. http://dx.doi.org/10.1016/S0950-5849(03)00032-6

[46] Khanom S, Heimbürger A & Kärkkäinen T (2015) "Can icons enhance requirements engineering work?",*Journal of Visual Languages & Computing*, Vol. 28,pp147–162. http://doi.org/10.1016/j.jvlc.2014.12.011

[47] Alsanoosy, T, Spichkova, M & Harland, J (2019). Cultural influence on requirements engineering activities: a systematic literature review and analysis. *Requirements Engineering*, pp1-24.

[48] Sedelmaier Y& Landes D (2017, April) "How can we find out what makes a good requirements engineer in the age of digitalization?", In *Global Engineering Education Conference (EDUCON)*, 2017 IEEE, pp. 230-238. http://doi.org/10.1109/EDUCON.2017.7942853

[49] Al-Rawas A & Easterbrook S (1996) "Communication problems in requirements engineering: A field study", Proceedings of the *First Westminster Conference on Professional Awareness in Software Engineering*, London. Retrieved from https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19970005658.pdf

[50] Klendauer R, Berkovich M, Gelvin R, Leimeister JM& Krcmar H (2012) "Towards a competency model for requirements analysts", *Information Systems Journal*, Vol. 22, No. 6,pp475-503. https://doi.org/10.1111/j.1365-2575.2011.00395.x