doi: 10.20532/cit.2020.1004868

Investigating Agile Requirements Engineering Practices in the South African Software Development Market

Nalindren Naicker[1] and Manoj S. Maharaj[2]

[1]ICT and Society Research Group, Durban University of Technology, Durban, South Africa
[2]University of Kwa-Zulu Natal, Durban, South Africa

The aim of this research study was to assess Agile RE practices in the South African software development industry and investigate secure Agile RE initiatives towards developing secure products. This qualitative research study was contextualized in seventeen South African software development companies. The researchers used structured interviews and document reviews as the primary data collection instruments. Qualitative data was analyzed inductively using content analysis. Emanating from the research were recommendations to guide a regular software developer on good Agile RE practices. The study concluded that although Agile Software Development is practiced in the South African software industry, there needs to be stricter adherences to the Agile Manifesto and Agile Security Manifesto in requirements engineering.

ACM CCS (2012) Classification: Software and its engineering → Software creation and management → Designing software → Requirements analysis

Software and its engineering → Software creation and management → Software development process management → Software development methods → Agile software development

Keywords: agile software development, customer collaboration, requirements, engineering, requirements prioritization, security requirements

1. Introduction

In requirements engineering (RE), information regarding the features of the system is collected from the customer. These features are in the form of functional and non-functional requirements. In traditional software development practice, RE takes place at the beginning of the software development process and is a once off process [1]. In order to meet the rapid demands for web and mobile applications, many software development practitioners are implementing Agile Software Development (ASD) [2]. Requirements engineering (RE), the earliest stage of software engineering, is the focus of this research study. Rigorous processes during this stage can ensure the development of high quality products that are safe to use in the software applications market.

In ASD, Agile RE is an iterative process, as this development methodology builds the system incrementally within a rapidly changing business environment [3]. Therefore, in Agile RE, requirements cannot be frozen [4]. Changes in requirements are created as a result of competitive threats on the market, stakeholders changing their preferences, changes in the development technology and short time required for the product to get to the market [5]. Agile RE is therefore dynamic and complicated.

Agile RE is also constrained by a number of factors. A combination of project factors such as schedule, risk, costs and human resources are constraining RE processes. In addition, the nature of the requirements is another constraining factor, as requirements have complexity, dependencies, importance, business value and volatility [4].

Owing to the dynamic nature of ASD and the challenges posed by constraints in the software development environment described above, developers tend to focus less on RE activities that take time, for example the identification of security requirements or elicitation by prototyping, which is important for quality system development [6, 7]. It was, therefore, important to assess secure Agile RE practices.

The aim of this study was to assess Agile RE practices, including security, in order to contribute to the good professional practice of regular software developer in Agile RE. Therefore, the study will attempt to answer the following question: *What are the requirements engineering practices in Agile RE in the South African software development landscape?*

Researchers agree that there is a lack of empirical studies on how companies are conducting RE in Agile Software Development and much more research is required in this critical area [4, 8, 9, 10]. In order to close the research gap, there is a need to conduct an in-depth analysis of Agile RE practices to assess their impact on the development of high quality software apps that are safe to use.

2. Literature Review

To assess RE processes, in line with the research question, we review the extent of literature from the existing body of knowledge.

The four main values that underpin ASD outlined in the ASD Manifesto [11] to promote quality products are, as follows.

1. *Individuals and interactions* over processes and tools.
2. *Working software* over comprehensive documentation.
3. *Customer collaboration* over contract negotiation.
4. *Responding to change* over following a plan.

Software developers transitioning to ASD must make a switch from traditional RE to Agile RE, with cumbersome traditional processes being permitted greater flexibility. Agile RE differs from traditional RE because it is ongoing or iterative [12]. Inayat and Salim in [13] proposed the term Agile RE to describe a process derived from flexibility in approaching traditional RE activities. Regardless of the RE approach, researchers are in agreement that RE plays an important role in ASD.

Unlike traditional RE, Agile RE allows for changes in user requirements. By accommodating changes in user requirements, ASD ensures that the outcome of the process are high quality requirements and customer satisfaction [1]. Therefore, ASD literature emphasizes that in gathering requirements, it is important to focus on some key aspects. In this regard, Inayat and Salim in [13] advise that, owing to the high volatility of requirements, constant collaboration among stakeholders is important. They cited communication and awareness as the most relevant socio-technical aspects of such collaboration. Wurfel *et al.* in [14] also suggested that requirements engineers must be mindful of the vagueness in customers' requirements, and must remain open. Openness ensures that requirements engineers are not influenced by existing software applications that they may have developed for another customer.

According to Schön in [2] in Agile, RE processes such as inception stage, elicitation stage, elaboration stage, negotiation stage, specification stage and validation stage are not clearly separated. Kassab in [15] and Sebega and Mnkandla in [10] were in agreement with this finding. Classical activities are merged and conducted iteratively in the South African Agile RE practice [10]. In Agile, RE is conducted just-in-time with little design upfront. Only the most important RE activities are elaborated.

Requirements elicitation, the first stage, comprises of problem solving, negotiation and specification [10]. Requirements are elicited through focus group sessions, interviews, brainstorming, questionnaires, discourse analysis and ethnography. Prototyping is another well-known strategy used to identify and determine the requirements from the customer perspective. Requirements consistency, variability, list size and complexity derived from the elicitation stage can hinder Agile RE [10].

Requirements analysis occurs immediately after elicitation. In the analysis, requirements are examined for complexity, completeness, ambiguity and contradictions [3, 16] with the objective of minimizing future changes. Issues in

Table 1. Criteria for requirements prioritization.

Criteria	Description	Author
Ease of use	Simple enough to implement.	[18]
Effort	Time taken to implement the feature.	[18]
Cost	Monetary cost of implementing the requirement.	[18]
Size (user stories)	Based on user stories, features or story points.	[17]
Business value	The return on investment that can be generated if the feature is implemented.	[17]
Negative value	How much it would detract from the products value if the feature is not implemented.	[17]
Complexity	Degree of difficulty to implement.	[4]
Dependence	Implementation of a requirement is dependent on another requirement being implemented first.	[4]
Volatility	Degree to which changes can be expected in the requirement. The stability of the requirement is considered.	[4]
Risk	Assess the risk for the project if the requirement is implemented.	[4]

this stage are resolved through meetings with stakeholders, modeling and prioritization [3]. The industry standard for Agile RE for analysis of requirements are JAD sessions [10]. Conflicts are resolved through workshop sessions where all stakeholders get involved to negotiate and resolve conflicts. A communication gap between IT specialists and customers leads to incomplete or incorrect requirements. Sebega and Mnkandla in [10] state that requirements negotiation, requirements prioritization and scalability problems are some of the issues that impair the analysis of requirements. Once the conflicts are resolved, the lists of requirements are ready to be prioritized.

In ASD requirements, prioritization occurs during the initial planning stage and then at the time of inter-iteration. Prioritization and reprioritization are based on business value. Continuous requirements prioritization is thus a core activity of Agile RE [4]. This ensures that the highly ranked requirements important for the project get developed first. This will bring the highest business value to the customer and lower the project risk by ensuring deliverables to the customer [17].

However, practitioners are confronted with difficulties in making decisions about which requirements should be considered at the inter-iteration time. Although developers are very skilled and are the most influential stakeholders, it is the client who should make the final decision on the priority of a requirement. What takes place in practice may not necessarily correspond with what is recommended in the Agile Manifesto [11]. It is difficult to reach consensus when there are multiple stakeholders and multiple decision making criteria [4]. Clearly, both developers and clients have a role in the decision making and their input on the value of the requirement in real world practice is significant.

There are various criteria identified in literature for requirements prioritization. A pertinent example is a criterion called 'negative value' introduced by Racheva in [17]. Table 1, synthesized from ASD literature, provides typical prioritization criteria that can be applied using an appropriate prioritization technique.

Appropriate prioritization technique is important as erroneous prioritization may increase the cost of development and lead to system and project failure [4]. Therefore, a proper requirements prioritization technique is critical to ensure that the requirements engineering process is efficient [19].

Sebega and Mnkandla in [10] state that there is a lack of focus from practitioners in handling non-functional requirements such as security and this constitutes a major problem in Agile RE. Security literature abounds with Security Requirements Engineering (SRE) frameworks or methodologies that guide secure software development activities for requirement engineering. It is a systematic way to elicit security requirements to protect the system from potential attacks [8]. These frameworks vary in how security requirements are derived. For example, the *misuse case* and *abuse case* frameworks deal with security from the viewpoint that users misuse the functionality of the system [20], while *risk-based* frameworks extract security requirements to mitigate the risks [21]. Haley *et al.* in [22] proposed a SRE framework which comprised the following stages: functional requirements identification, identification of security goals, identification of security requirements and finally the construction of satisfaction requirements.

In Agile RE, it is not quite clear how SRE takes place in a rapid release software development approach. The Agile Manifesto created in 2001 provided firm guidelines for software developers in order to improve traditional practices [11]. Fifteen years later, in 2016, the Agile Security Manifesto was launched to combat security problems that plagued software developers. The four principles of the Agile Security Manifesto are, as follows [23].

1. *Rely on developers and testers more* than on security specialists.
2. *Secure while we work more* than after we're done.
3. *Implement features securely more* than adding on security features.
4. *Mitigate risks more* than fix bugs.

Principle 1 states that it is not possible for all companies to have security teams or specialists. The process is rapid and lightweight, therefore security of the system depends on the ASD team that owns the security. This means that they must be trained and aware of security. According to principle 2, security must be incorporated into ASD practices and should not be added as an afterthought. Principle 3 encourages the use of tried and tested security measures, for example security frameworks such as authentication and password storage, to prevent developers from being detracted from focusing on the business value. Finally, principle 4 promotes security risks assessments over an ad hoc approach to security [23].

The SRE frameworks and approaches provided by security researchers and practitioners in literature are complex and not customised for lightweight Agile RE practices. The Agile Security Manifesto states that they must be blended naturally into Agile RE activities by requirements engineers who must be trained and be knowledgeable about application security issues. Requirements engineers must have an understanding of what type of security requirements must be included in the system.

Haley *et al.* in [22] proposed how security can be factored into Agile XP requirements engineering practices. The approach focused on abuser stories. This means that it takes into consideration the actions of a hacker who intends to abuse the system and mitigates against these actions. The following Agile RE stages were suggested: identification of critical assets of the system; formulation of abuser stories; assessment of the abuser story to determine risk; negotiation between abuser and user stories; identify security user stories; definition of security-related coding standards and, lastly, cross-checking abuser stories and countermeasures against system abuse.

Adequate security education and training are essential for the development team to address the scarce security skills and tools in industry [8]. There is little support from senior management with respect to adequate funding for security [2]. Development teams are not receiving regular updates about common vulnerabilities from security knowledge sources to keep them informed of the latest security developments [20]. This research study will assess if security is considered in Agile RE practices in industry.

Kassab in [24] proposed five focus areas of Agile RE, namely: elicitation, analysis and presentation, management, effort estimation and tools. Kassab in [24] concluded from a survey on existing Agile RE practices, based on the five focus areas of 247 IT professionals from 23 countries, that RE processes within Agile RE are not very clear. Also, the main difference between RE processes and Agile RE activities is that there is a lack of documentation in Agile RE. He highlighted requirements management as an important stage in Agile RE to ensure traceability of requirements. On the positive side, he found that Agile RE is effective to get continuous feedback from the customer, thereby validating the development of the new system.

In summary, the ASD literature suggests that practitioners of software development do not clearly agree on standardized Agile RE practices. The lack of coherent and consolidated views on Agile RE makes this a valuable research area [13]. The literature has highlighted requirements elicitation, security, customer involvement, analysis and prioritization of requirements as the focal points.

A review was conducted on prior work to gauge the methodologies used to analyze software development markets. The literature revealed the lack of studies conducted in the software development market on Agile methodology in South Africa. Sebega and Mnkandla in [10] conducted a quantitative study in order to understand 'Agile processes' in the software development market in South Africa. Selecting a sampling frame was a challenge in this study, since there was only a vague idea of the target population. A list of all Agile companies in South Africa is non-existent. Researchers relied on a published list of the Top500, that is, a repository of the best managed companies in South Africa. A non-probabilistic sampling technique was used and more than 220 participants were contacted. A survey questionnaire was distributed to around 110 participants in the software development market. Only 25 participants responded to the survey. Issues in elicitation, analysis, documentation, validation, and management were analyzed broadly through Likert-type questions. This study revealed a strong penchant for principles of the Agile Manifesto. This research study follows on the Sebega and Mnkandla study in [10] with a more in-depth analysis of the South African software development market through a qualitative lens. Therefore, the qualitative methodology chosen is appropriate.

An overview of international studies revealed numerous types of studies conducted on specific industries in software development. Wagner *et al.* in [25] conducted an international study of Agile requirements engineering in practice focusing on the status quo and critical problems in the industry. A non-probabilistic sampling technique was used. The study was based on the responses from 92 organisations in Austria, Germany, Ireland, Canada, USA, Estonia, Finland, Norway, Sweden and Brazil. The data was collected from 228 responses from the companies classified as small, medium and large. The survey results showed that RE concentrates on free-text documentation of requirements elicited with a variety of techniques. The backlog is used to deal with changing requirements.

There were several qualitative studies conducted as well. Researchers in Ireland, namely, Noll *et al.* in [26], conducted a qualitative study of an application in the Open Source Software Development market. This study examined discussion forums of an open source project, namely OpenEMR, an open source medical practice management application. Through a qualitative analysis of 16 core developers they found out that end users play an important role in the development of open source software. In a qualitative study on Agile RE with 16 organisations, Cao and Ramesh in [27] sought to answer two questions: What RE practices do Agile developers follow? What benefits and challenges do these practices present? A multi-sited case study approach was used as a sampling technique and grounded theory was used to analyze the data. The study identified seven Agile RE practices in the organizations.

The literature on the methodologies used to analyze a software development market shows that both qualitative and quantitative studies were conducted. Quantitative studies focused on the surveys that involved a large number of participants for general and broad findings on the Agile software development market whilst qualitative studies through interviews, focus groups and discussion forums with a smaller number of informants sort an inquiry with greater depth. This study conducted an analysis of Agile RE activities in the South African industry to gain more insight and provide critical review of the approaches used. The impact of the approaches used on the decisions taken to elicit and implement non-functional requirements, such as the security of the system, was also of interest in this research study.

3. Setting and Methodology

In this qualitative study, the researchers sought to understand requirements engineering practices in Agile RE from the viewpoints of practitioners in the real life South African software development context. The following practices discussed in Agile RE literature were covered in the data collection: requirements elicitation, requirements analysis, requirements prioritization, the security issue, customer involvement and a general idea about Agile RE satisfaction. A literature review on methodologies of prior work in the software development revealed that both qualitative and quantitative studies were conducted with many studies opting for non-probabilistic sampling techniques.

In South Africa, the population of software development companies is unknown. In order to obtain a statistical figure, a LinkedIn search was conducted. The results of the search showed 165 software development companies. Using simple random sampling, one hundred software development companies were chosen for this research study. Simple random sampling is a technique where each member of the population has an equal chance of getting selected in the sample [28], thus making the results more applicable to general population.

Seventeen software companies responded to the researchers' invitation to participate in the research. The composition of the companies in terms of their fields was as follows: Gaming (5), Medical (3), Transport (1), Finance (2), Education (2) and Non-Specific (4). This cross-sector analysis ensured that the results of Agile RE practices were not biased towards a single sector. The composition in terms of size was as follows: 'Less than 25 staff' (10); 'Between 25 and 50 staff' (5) and '50 and more staff' (2). A non-probability, purposive sampling approach called quota sampling was used to select informants from the sample population, ensuring the inclusion and representation of all five stakeholder categories from the sample, namely: business analyst, product owner, customer/client proxy, project manager and developer [28]. Thus, while the companies were chosen randomly from the population, stakeholder categories were chosen purposively from the sample. Table 2 shows involvement of the stakeholders from different companies.

Appendix 2 provides further details on the profile of the companies chosen for the research. The data was collected through structured interviews and document analyses. Documents analyzed were the Business Requirements Documents and use case diagrams. According to Silverman in [29], an interview helps us to generate the data which gives authentic insight into people's experiences. Interviews permit informants to give detailed information that cannot be obtained by a survey [30]. Seventeen face-to-face interviews were conducted with open ended questions for an average duration of one hour per interview.

The interview schedule was created according to important aspects of Agile RE, such as elicitation, analysis, analysis and prioritization of requirements, documentation and security discussed in the literature review. The questions were carefully crafted to explore Agile RE practices and developed in accordance with the study aim. The issues chosen for the interviews were outlined in the literature review in studies such as [3, 4, 8, 10, 15, 18, 20, 27].

Face-to-face interviews sought detailed experiences, motives, and opinions from the perspective of experts in the field. Interviews were audio recorded. Questions on the interview schedule were enumerated, which made them easy to reference during the analysis. The researchers ensured that the recording was audible for the transcriber and sufficient time was allocated for transcribing. Transcripts were produced from audio recordings of interviews

Table 2. Involvement of stakeholders.

Stakeholder	Company	Number Interviewed
Business Analyst	Gaming (1) and Finance (2)	3
Product Owner	Gaming (2) and Medical (2)	4
Customer	Transport (1) and Gaming (2)	3
Project Manager	Education (2)	2
Developer	Non-specific (4) and Medical (1)	5

and analyzed using content analysis. Appendix 3 contains a sample of the interview data and shows the kind of responses received to each question. Qualitative data was analyzed using content analysis through the thematic analysis approach with the assistance of a computer program called Nvivo Pro [7, 20]. In order to arrive at themes, codes were used. The study employed a coding scheme that was repeatable, transparent and led to objective findings. See appendix 3, appendix 4 and appendix 5.

Coding began by choosing a small number of well-defined labels (codes) for the transcript responses. Appendix 5 contains coding definitions. This ensured that coding was straightforward and could be completed quickly and easily. The codes that were assigned were all-inclusive and mutually exclusive [31]. To explain using a concrete example from the study as a sample of how coding was processed, consider the question on "Provide a detailed explanation of how the requirements were elicited for this project?". The codes chosen for this question were 'Elicitation techniques'; 'Tools'; 'Specific teams'; 'Customer Involvement'; 'Role of Business Analyst'; 'Core RE activities'; 'Communication'; 'Text Documents'. The code dimensions chosen were clear enough so that respondents' responses in the interview transcript could not fall into two codes at the same time. The next step was to codify the labels. This meant arranging things in a systematic order by aggregating them into categories. The data was segregated or grouped or linked in order to consolidate meaning and explanation [31]. In the search for patterns in the data, similarly coded data was grouped into a single category. Intuition and tacit knowledge were the skills required in coding and categorizing [31]. Taking an example, three categories, namely: 'Elicitation of Requirements', 'Customer Involvement' and 'Output Documentation' emerged from the codes using the question in the above example. Overall, 21 categories were created in this study, as shown in appendix 4. A theme is an extended phrase or sentence that identifies a unit of the data. The theme chosen is the outcome of the coding and analytical reflection of the interviews and literature review [31]. Taking the example forward, the three categories chosen, namely: 'Elicitation of Requirements', 'Customer Involvement' and 'Output Documentation' were clustered into the theme 'Requirements Elicitation'. Appendix 4 contains the entire coding scheme for all interview questions.

Codes are used to give collected data meaning [32]. Coding was a way of indexing or categorizing the text in order to establish a framework of thematic ideas about it. Codes formed a focus for thinking about the text and its interpretation. The codes used for this research project [33] were descriptive codes such as 'elicitation techniques', categorical codes such as 'security information sources' and analytical codes such as 'Core RE activities'. A computer program Nvivo Pro was used for filing and retrieving coded information. An inductive approach was used to arrive at the final thematic areas.

Reliability and reproducibility of the data analysis were enhanced by asking an external person to redo the analysis using the same coding definitions. Notes were compared with the external source to validate the data analysis. The results showed strong correlation between the researcher and the external person. Furthermore, more than one type of qualitative data, namely interviews and document analysis were used in the analysis. This allowed for triangulation of the results. The researcher ensured sampling adequacy through saturation and replication [34]. Despite a small number of respondents in each company, the researcher was able to gauge saturation when no new insights were given by additional sources of data.

Based on the results of the content analysis and overriding themes, several findings were made.

4. Findings

Key themes within Agile RE emerged preceding the extensive literature review and the findings from the qualitative data analysis. Table 3 shows the clustering of related categories into themes.

Appendix 4 and Table 3 are the sources of the assessment of the RE practices in the South African software development market. The themes emanated corresponding to the research question were as follows: Theme 1: Requirements Elicitation; Theme 2: Security Requirements Elicitation; Theme 3: Security approach; Theme 4: Security training; Theme 5: Customer Involvement; Theme 6: Prioritisation of Requirements and Theme 7: ASD RE Satisfaction.

The study findings are discussed below, in terms of the themes that emanated following the coding.

4.1. Discussion under Themes

Theme 1: Requirements elicitation. Requirements elicitation in RE focuses on ensuring that high quality requirements are extracted from the user. The document review in this study revealed that requirements generated for the projects are highly refined, lacking complexity and ambiguity. It is important to note that across all companies, the entire software development team was not involved in the Agile RE practice of requirements elicitation. The business analyst emerged as the most important stakeholder involved in meetings with the customers, to elicit requirements. A product owner explained that the user requirements were derived by the business analysts who visited the client and obtained the specifications for the system. They brought the requirements back, did further brainstorming with the project manager and proposed the cost and estimates. Business analysts generated a Business Requirement Document with the inputs from project managers. Customers do not play an active role. When this document of requirements is completed, it gets signed by the client. Thereafter, project managers take responsibility for constructing the system in several releases. One company reported that 5 of their clients had a business analyst on their staff. These clients make the process of requirements elicitation very easy for the software developer, as the client's business analyst captures the critical customer requirements on the client side. This reduces the overall RE process time for the business analyst, since the client's business analyst submits a completed Business Requirements Document to the software development business analyst. Prototyping to elicit requirements was not widely used by companies. One developer said that, "we don't have time to build a prototype". The data analysis

Table 3. Outcome of the coding process.

CLUSTERING	THEMES	RESEARCH QUESTION
Elicitation of Requirements Customer Involvement Output Documentation	***Requirements Elicitation***	**What are the requirements engineering practices in Agile RE in the South African software development landscape?**
Elicitation of Security Requirements Security Teams Security Risk Assessment Identification Time	***Security requirements identification***	
Known Security approach Customer Involvement Security Policy Benefits of security Role of ASD Team	***Security approach***	
Security Policy Security Training Security Awareness	***Security training***	
Customer Involvement Roles in prioritisation of requirements In-house Business Analyst	***Customer Involvement***	
Prioritisation techniques Roles in prioritisation of requirements Important criteria	***Prioritisation of requirements***	
Non-functional requirements Team Dynamics Challenges in Agile RE	***Agile RE Satisfaction***	

also revealed socio-technical tension between requirements engineers and developers.

Table 4 below shows the summary of findings for theme 1.

Table 4. Summary of findings for requirements elicitation.

Noticed Agile RE practices
In 14 out of 17 companies the business analyst elicited all requirements. Team not involved.
Prototyping was used in 4 out of 17 companies as an elicitation technique.
Business Requirements Document generated by the business analyst in 14 out of 17 companies.
5 companies had a business analyst on their staff.
Business analyst works independently of the ASD team in 16 out of 17 companies.
Customers need to play a more active role in the elicitation of requirements in 16 out of 17 companies.
Business analyst establishes viewpoints in 14 out of 17 companies.
Viewpoints captured in Business Requirements Document in all companies.
Business analyst clarifies users' viewpoints when necessary in 15 out of 17 companies.

Theme 2: Security requirements identification. Security requirements focus on protecting assets of the system and prevent unauthorised access and violation of system assets. Common practices in Agile RE are that large companies have dedicated security teams who have established processes to ensure that security requirements are identified. Small to medium companies rely on the discretion of the software development team, such as business analyst or team leader, for identification of security requirements during requirements engineering. Generally, in these companies, security requirements are limited in number and are generic. Security requirements are identified as system enhancements when there is a violation, or when the customer specifically requests it. Sebega & Mnkandla in [10] concur with this finding as they suggest a lack of integration in Agile RE of non-functional requirements such as security.

Table 5 below shows the summary of findings for theme 2.

Table 5. Summary of findings for security requirements identification.

Noticed Agile RE practices
Security requirements identified by the business analyst in 11 out of 17 companies
Insufficient security requirements identified in 14 out of 17 companies.
Security requirements identified at coding when there is a glaring issue in 10 out of 17 companies.
Security requirements identified are generic (not customised) in 10 out of 17 companies.
Customer requirements are only specified when the company experienced a security violation in another project and wanted to mitigate upfront in the new project. This was the case in 7 out of 17 companies.
Security requirements were identified after a system violation in 11 out of 17 companies.

Theme 3: Security approach. A security approach in requirements engineering is to deploy a systematic method to unravel and identify security requirements. In software engineering literature, this RE process is known as security requirements engineering. Interviews, Business Requirements Documents and informal discussions with developers have showed that small to medium companies have not adopted any known approaches to security. It was the duty of the business analyst (a subject matter and domain expert) through consultation with the client to specify security requirements at the requirements engineering phase. Owing to the business analyst's limited knowledge on security, very few to no security requirements were elicited. Security approaches in general may be factored into the design and development under the following circumstances:

1. developer requests for security approaches during coding;
2. customer requests security during requirements engineering;
3. business analyst includes security through consultation with the client and client security policies during requirements engineering.

However, with regard to point (3) above, the researcher noted, after viewing the requirements document, that only a few critical security requirements such as authentication and access control were factored in.

Table 6 below shows the summary of findings for theme 3.

Table 6. Summary of findings for security approach.

Noticed Agile RE practices
No formal security methodologies used in all 17 companies.
No security experts used in 15 out of 17 companies.
Ad hoc approaches on security in 15 out of 17 companies.
Developers request security approach during coding in 11 out of 17 companies.
Business analyst to ensure security requirements are elicited through consultation with the client and client security. These are restricted to authentication and access control in 10 out of 17 companies.
Customer can request security during requirements engineering. This happened in 2 out of 17 companies.

Theme 4: Security training. Security training in application development assists developers to combat known vulnerabilities.

Two distinguished practices for application security training emerged. In large companies, security training is provided for the affected employees, whilst in smaller companies security training is not prioritised. Furthermore, it was apparent that stakeholders were unaware that the latest trends in terms of application security can be sourced from security knowledge sources such as Open Web Application Security Project (OWASP) and Common Weakness Enumeration (CWE).

Table 7 below shows the summary of findings for theme 4.

Table 7. Summary of findings for security training.

Noticed Agile RE practices
Application of security training not prioritised in 16 out of 17 companies.
Poor awareness of security knowledge sources in 15 out of 17 companies.

Theme 5: Customer involvement. Customer involvement in ASD projects is an important ASD principle and in order to reap the benefits of ASD, the customer must be involved in the software development. The content analysis showed that customer involvement in ASD was poor across all organisations. The customer came to meetings at the beginning of the ASD project. They were not working with the development team on-site and were requested to attend meetings only. If they were unable to attend, the meeting was re-scheduled. They were only involved in very crucial meetings. Some customers had their own in-house team of business analysts whereby they elicited all the requirements, including security.

Table 8 below shows the summary of findings for theme 5.

Table 8. Summary of findings for customer involvement.

Noticed Agile RE practices
Poor customer involvement in 16 out of 17 companies.
5 companies out of 17 had their own teams of business analysts to elicit requirements on their behalf.

Theme 6: Prioritization of requirements. Requirements prioritization is a very important process in Agile RE. Requirements of high priority are implemented first, to add business value to the customer. Incorrect prioritization can lead to a loss of value, eventually resulting in project failure. When security requirements are lowly ranked, they don't get implemented. Prioritization is an important component of requirements analysis.

ASD literature supports the strategy that multiple stakeholders, for example the business analysts, the project manager and the customer using multiple decision making criteria, should prioritize requirements. The study indicated that, in Agile RE practice, only the project manager, using multiple criteria decided the priority of the requirement. Project managers or the team leaders consulted technical documents and business analysts when necessary and prioritized based on their vast experience in the field or reached consensus with other process owners, such as the business analyst. Six companies did not implement any known requirement prioritization technique and no software application was used to assist in the prioritization of requirements.

The project manager completed the ranking process on behalf of the customer. He ranked the customer's preference first, followed by ranking according to various other criteria. This meant that the customer was not directly involved in the process.

General ranking process at an established international software development company was summarized by a business analyst. All functional requirements were ranked. If the customer has 10 requirements, they inform the project manager what features they want first. Customers normally advise on the basis of core requirements. According to a senior developer in a company, ranking is done according to the following approach. Developers analyze the requirements, for example if feature XX is released and the development time is 4 months and if feature YY is released and the development time is 2 months. When feature XX is live, the turnover (revenue) for the customer in one month is larger than feature YY in one month. XX has higher revenue for the customer and is selected for implementation. Developers consider the development time and revenue to the customer. This is known as an opportunity cost. As a rule, developers prioritize where they can generate more profits for the customer.

Sometimes the feature is simple and the development time is long. In this case the project manager informs the customer of the time taken to release the feature. If the release of the feature has a huge impact on the turnover, this will be communicated by the customer. Despite a lengthy development time, it is chosen for implementation. Also, there can be interdependence of requirements, for example features 1 and 2 must be done before feature 5. In this case, there is no negotiation with the customer. Features 1 and 2 are both selected together for implementation. Other important criteria that developers consider are business value and weight of the customer in a multi project environment.

Non-functional requirements were not prioritized at 10 companies. There were standard ways of doing things and non-functional requirements were brought in from past projects. Developers did not rank non-functional requirements. Once the frameworks were in place for non-functional requirements, it was easy for the development team to implement at coding stage. For example, if they wanted to create a login screen, the encryption mechanism was taken from the security framework. It was considered beneficial for a security framework and a scalability framework to be added to the new project. The need to code complex repetitive functions was removed. Existing projects were often used as a source of information and this saved time, resources and money.

Table 9 below shows the summary of findings for theme 6.

Table 9. Summary of findings for analysis and prioritization of requirements.

Noticed Agile RE practices
No team involvement in requirements prioritization (project manager and/or business analyst in 15 out of 17 companies.
No proper prioritization techniques used in 16 out of 17 companies.
Lack of direct involvement of customer in 16 out of 17 companies.
No specific prioritization technique in 15 out of 17 companies.
Prioritization generally takes place through consensus in 14 out of 17 companies.
Important criteria: opportunity cost, business value and dependence in 14 out of 17 companies.

Theme 7: Agile RE satisfaction. The researcher explored team members' satisfaction with Agile RE practices. Agile RE practices in this research were used to assess the extent to which traditional RE processes were applied to Agile RE. The data analysis indicated that all team members did not explicitly express their dissatisfaction with Agile RE practices, but rather raised a number of concerns for improvement.

Common practice in Agile RE in the South African industry suggested that developers were not inclusive of customers in RE processes, as in the case of elicitation of non-functional requirements such as security. Developers did not use any form of communication with the customer to obtain input from the customer on security issues.

An observation made by the developers was that the business analyst was not sufficiently knowledgeable about the actual customer requirements. More often than not there were changes to the requirements that came from the business analyst. These changes were not about the client changing requirements, but emerged from the lack of understanding by the business analyst of what the client really wanted. This hindered the development process.

The business analyst should liaise with the software development team for technical specifications before finalizing the technical specifications document. However, the business analyst did not get input from the team members when generating technical specifications document, hence compromising the quality of the requirements. A developer explained: "The business analyst always misses out things. There are always loopholes and we seldom get a perfect requirements document".

The document review showed that requirements are mainly in the form of user stories. These requirements may be a task, or further broken down into many tasks, that are managed using project management software such as Pivotal Tracker or Jira. Many developers misunderstood the idea of user stories and were of the opinion that user stories do not give the broad picture of the system under development as compared to a UML diagram. When a developer gets to implement a requirement, they need to have a high level view of the system. Developers believed that they need to view UML diagrams to have a high level view of user requirements. According to [20], user stories must be discussed and extended before implementation.

Many of the teams felt that the personality of requirements engineers can impact the success of the project. Personnel involved in collecting requirements must have a good temperament to deal with queries and share information with the development team. Other than the customer, they have the best understanding of the system.

Developers believed that they were not taken seriously due to certain biases of senior stakeholders. A business analyst explained that the pecking order of the team was very important. He said that, "if you are not senior in the team ranking, nobody listens to you, even if you make a valuable contribution". Senior team members don't take juniors' views into account and, as a result, it becomes very difficult and frustrating to reason with higher authority. A junior developer confirmed that, "In scrum meetings, difference in opinion can lead to serious personality clashes which can be detrimental to the project and to your job security."

Altogether 10 developers felt that a requirements engineer is an important stakeholder in the software development process as they generally have product knowledge, not the developer. The concern was that software development is a competitive industry and should the requirements engineer leave, it creates lots of issues for the programmers. The business analyst has expert knowledge in the field and supports all stakeholders. A developer explained that the biggest problem encountered is when staff responsible for gathering the requirements leave. He was working on a financial application and was struggling with understanding the requirements from an incomplete requirements document. He could not consult anyone. He explained that, "lots of information is lost when the staff member is lost to the company".

Table 10 below shows the summary of findings for theme 7.

Table 10. Summary of findings for Agile RE satisfaction.

Agile RE practices
More customer involvement, especially for non-functional requirements in all companies.
Competence of the business analyst in question in 14 out of 17 companies.
No high level view of new system by developers in 14 out of 17 companies.
Team dynamics an issue in 12 out of 17 companies.
Staff turnover a problem – no staff retention in 16 out of 17 companies.
High workload in 15 out of 17 companies.
High paced ASD environment in all companies.

4.2. Recommendations to improve Agile RE practices

Based on the findings in the previous section, guidelines are provided for improvement in professional practice.

Table 11. Suggestions for improvement of poor Agile RE practices.

Poor Agile RE practices	Suggestions for improvement
Requirements elicitation ● Prototyping is not a widely used elicitation technique. ● Business analyst elicited all requirements. Team not involved. ● Business analyst works independently of the ASD team. ● Customers need to play a more active role in the elicitation of requirements.	● A prototype is used to implement customer requirements giving them an opportunity to experiment with the system thereby revealing any omissions or errors [7]. Prototypes will ensure that project goals are met and they can be used to verify that the right system is built. ● Increase Joint Application Development sessions (JAD) during the Agile RE phase. JAD sessions increase customer involvement and resolve conflicting requirements [3]. ● ASD companies must ensure that team composition strategies consider that the personalities of requirements engineers must be aligned with the principles and values of the ASD outlined in the manifesto namely, "Projects are built around motivated individuals who should be trusted" and "self-organising teams" [11, 30]. ● ASD methodologies recommend at least a single person who is an expert in the domain, to represent all the customers [35, 36].
Security ● Insufficient security requirements identified. ● Security requirements identified at coding when there is a glaring issue. ● Security requirements identified are generic (not customised). ● Security requirements are only specified when the company experienced a security violation in another project and wants to mitigate upfront in the new project. ● Security requirements are identified after a system violation. ● Application security training not prioritised. ● Poor awareness of security knowledge sources. ● No security approach. ● Small companies do not have budget for security.	● In Agile RE, it is recommended that customers have meetings with project managers to discuss security [8]. ● The best time to involve all stakeholders in security requirements elicitation is in the security risk analysis activity where the risk factors of requirements are assessed [23,34]. ● Requirements engineers must be kept abreast of the latest trends in application security, by means of attending annual trainings and regular consultations with information from security knowledge sources such as OWASP and CERT, to improve their knowledge in application security [6, 8]. ● A lightweight security approach over more complex security approaches is recommended [35].

Poor Agile RE practices	Suggestions for Improvement
Analysis and prioritization of requirements ● No team involvement in requirements prioritization (project manager and/or business analyst). ● Lack of direct involvement by customer. ● No specific prioritization technique. ● Prioritization generally takes place through consensus.	● There must be a more inclusive approach to ensure best practice. The prioritization of requirements must occur immediately after the analysis process [3].
Agile RE satisfaction ● More customer involvement, especially for non-functional requirements. ● Competence of the business analyst in question. ● No high level view of new system by developers. ● Team dynamics an issue. ● Staff turnover a problem – no staff retention.	● Business analysts must have a specialist knowledge of the product [36]. ● One example is to provide annual vacation trips for top performers. This once off incentive will motivate staff to perform well and does not come at great cost to the company [37]. ● Poor team dynamics wastes time, money and resources. Study [38] recommends diagnosis of the problem, as this will differ from team to team. Once the problem is identified, an appropriate strategy can be put in place to improve team dynamics.

5. Conclusion

RE is a critical factor in software development and anecdotal evidence still suggests poor practices in the industry. The study findings reveal that customers should get more involved in Agile RE, especially in requirements prioritization and security. Furthermore, in South African companies, staff retention and specialist knowledge in vital RE processes, such as security, is lacking.

A basic checklist for requirements engineers, synthesized from the findings of the research study to guide a regular Agile software developer on requirements engineering practices ensures that secure, high quality ASD applications are produced but are also safe to use. While there is a strong desire to align with Agile Manifesto, there must be stricter adherence to the Agile Security Manifesto to reap full benefits of the methodology. Future research could consider distributed and global software development teams. There is a need for more research in Agile RE on distributed projects. The challenges and best practices from research with distributed ASD teams will allow the Agile Software Development community to gain better insight. Researchers steer away from this kind of research owing to difficulties in acquiring data. The outsourcing of features for the systems development may prove to be a convenient approach for incorporating security requirements and security features. However, this needs to be assessed empirically on the ground.

References

[1] Pressman, R. S., & Maxim, B. R. (2015). Software Engineering: A Practitioners Approach. Singapore: McGraw Hill Education.

[2] E.-M. Schön *et al.*, "Agile Requirements Engineering: A Systematic Literature Review", *Computer Standards & Interfaces*, vol. 49, pp. 79–91, 2017. https://doi.org/10.1016/j.csi.2016.08.011

[3] A. De Lucia and A. Qusef, "Requirements Engineering in Agile Software Development", *Journal of Emerging Technologies in Web Intelligence*, vol. 2, no. 3, pp. 212–220, 2010. http://dx.doi.org/10.4304/jetwi.2.3.212-220

[4] R. H. AL-Ta'ani and R. Razali, "Prioritizing Requirements in Agile Development: A Conceptual Framework", in *Proc. of the 4th Interntional Conference on Electrical Engineering and Informatics*, Selangor, 2013.

[5] B. Ramesh *et al.*, "Agile Requirements Engineering Practices and Challenges: An Empirical Study", *Information Systems Journal*, vol. 20, no. 5, pp. 449–480, 2010. https://doi.org/10.1111/j.1365-2575.2007.00259.x

[6] P. Salini *et al.*, "A Model-Based Security Requirements Engineering Framework Applied for Online Trading System", in *Proc. of the IEEE International Conference on Recent Trends in Information Technology*, Chennai, 2011, pp. 1195–1202.

[7] A. Sutcliffe and P. Sawyer, "Requirements Elicitation: Towards the Unknown Unknowns", in *Proc. of the 21st IEEE International Requirements Engineering Conference (RE)*, 2013.

[8] H. El-Hadary and S. El-Kassas, "Capturing Security Requirements for Software Systems", *Journal of Advanced Research*, vol. 5, no. 4, pp. 463–472. https://doi.org/10.1016/j.jare.2014.03.001

[9] I. Inayat *et al.*, "A Systematic Literature Review on Agile Requirements Engineering Practices and Challenges", *Computers in Human Behavior*, vol. 1, no. 1, pp. 915–929, 2015. http://dx.doi.org/10.1016/j.chb.2014.10.046

[10] Y. Sebega and E. Mnkandla, "Exploring Issues in Agile Requirements Engineering in the South African Software Industry", *The Electronic Journal of Information Systems in Developing Countries*, vol. 81, no. 1, pp. 1–18, 2017. https://doi.org/10.1002/j.1681-4835.2017.tb00597.x

[11] K. Beck *et al.*, "Manifesto for Agile Software Development", 2001.

[12] R. M. Fontana *et al.*, "Processes Versus People: How Should Agile Sofware Development Maturity Be Defined?", *The Journal of Systems and Software*, vol. 97, no. 1, pp. 140–155, 2014. https://doi.org/10.1016/j.jss.2014.07.030

[13] I. Inayat and S. S. Salim, "A framework to Study Requirements-Driven Collaboration Among Agile Teams: Findings from Two Case Studies", *Computers in Human Behavior*, vol. 51, no. 1, pp. 1367–1379, 2015. https://doi.org/10.1016/j.chb.2014.10.040

[14] D. Wurfel *et al.*, "Grounded Requirements Engineering: An Approach to Use Case Driven Requirements Engineering", *The Journal of Systems and Software*, vol. 1, no. 1, pp. 1–13, 2016. https://doi.org/10.1016/j.jss.2015.10.024

[15] M. Kassab, "The Changing Landscape of Requirements Engineering Practices over the Past Decade", in *Proc. of the IEEE 5th international workshop on empirical requirements engineering (EmpiRE)*, 2015, pp. 1–8.

[16] A. L. Z. de Souza Bermejo *et al.*, "Agile Principles and Achievement of Success in Software Development: A Quantitative Study in Brazilian Organisations", *Procedia Technology*, vol. 16, no. 1, pp. 718–727, 2014. http://dx.doi.org/10.1016/j.protcy.2014.10.021

[17] Z. Racheva *et al.*, "Do We Know Enough about Requirements Prioritisation in Agile Projects: Insights from a Case Study", in *Proc. of the 18th IEEE International Requirements Engineering Conference*, Sydney, 2010, pp. 147–156. https://doi.org/10.1109/RE.2010.27

[18] M. S. Hasan *et al.*, "An Evaluation of Software Requirement Prioritisation Techniques", *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 8, no. 9, 2010.

[19] P. Achimugu *et al.*, "A Systematic Literature Review of Software Requirements Prioritisation Research", *Information and Software Technology*, vol. 56, no. 1, pp. 568–585, 2014. http://dx.doi.org/10.1016/j.infsof.2014.02.001

[20] P. Salini and S. Kanmani, "Application of Model Oriented Security Requirements Engineering Framework for Secure E-Voting", in *Proc. of the 6th International Conference on Software Engineering (CONSEG)*, 2012, pp. 1–6. https://doi.org/10.1109/CONSEG.2012.6349489

[21] A. Souag, "Towards a New Generation of Security Requirements Definition Methodology Using Ontologies", in *Proc. of the 24th International Conference on Advanced Information Systems Engineering (CAiSE'12)*, Gdańsk, Poland, 2012, pp. 25–29.

[22] C. B. Haley *et al.*, "A Framework for Security Requirements Engineering", in *Proc. of the International Workshop on Software Engineering for Secure Systems*, 2006, pp. 35–42.

[23] Info Security Group, "Agile Security Manifesto", [Online]. Available: https://www.infosecurity-magazine.com/white-papers/agile-security-manifesto/ (Accessed 29 Feb 2020).

[24] M. Kassab, "An Empirical Study on the Requirements Engineering Practices for Agile Software Developement", in *Proc. of the 40th Euromicro Conference on Software Engineering and Advanced Applications*, Verona, 2014, pp. 254–261. https://doi.org/10.1109/SEAA.2014.77

[25] S. Wagner *et al.*, "Agile Requirements Engineering in Practice: Status Quo and Critical Problems", *CLEI Electronic Journal*, vol. 21, no. 1, p. 15, 2018.
https://doi.org/10.19153/cleiej.21.1.6

[26] J. Noll *et al.*, "A Qualitative Study of Open Source Software Development: The Open EMR Project", in *Proc. of the IEEE International Symposium on Empirical Software Engineering and Measurement*, 2011, pp. 30–39.
https://doi.org/10.1109/ESEM.2011.11

[27] L. Cao and B. Ramesh, "Agile Requirements Engineering Practices: An Empirical Study", *IEEE Software*, vol. 25, no. 1, pp. 60–67, 2008.
https://doi.org/10.1109/MS.2008.1

[28] I. Sommerville, "Software Engineering", 10th ed., London: Pearson, 2016.

[29] D. Silverman, "Qualitative Research: Theory Method and Practice", London: Sage Publications Ltd, 1998.

[30] J. F. Gubrium and J. A. Holstein, "Handbook of Interview Research: Context and Method", SAGE Publications, 2002.

[31] J. Saldaña, "The Coding Manual for Qualitative Researchers", Sage, 2015.

[32] N. Walliman, "Your Research Project", London: Sage Publications Ltd, 2004.

[33] G. Gibbs, "Analyzing Qualitative Data", London: Sage, 2013.

[34] J. M. Morse *et al.*, "Verification Strategies for Establishing Reliability and Validity in Qualitative Research", *International Journal of Qualitative Methods*, vol. 1, no. 2, pp. 13–22, 2002.
https://doi.org/10.1177/160940690200100202

[35] I. A. Tondel *et al.*, "Security Requirements for the Rest of Us: A Survey", *IEEE Software*, vol. 25, no. 1, 2008.
https://doi.org/10.1109/MS.2008.19

[36] E.-M. Schön *et al.*, "Agile Values and Their Implementation in Practice", *IJIMAI*, vol. 3, no. 5, pp. 61–66, 2015.
http://dx.doi.org/10.9781/ijimai.2015.358

[37] R. Volper, "7 Compensation Tactics to Help Retain Employees", 2012, [Online]. Available:
https://www.cnbc.com/id/46045960

[38] Team Technology, "Strategies to Improve Team Dynamics", 2017, [Online]. Available:
http://www.teamtechnology.co.uk/team/dynamics/strategies/

Appendix 1

Interview Schedule for Secure Requirements Engineering in Agile

The purpose of this questionnaire is to:

- Evaluate the extent to which secure RE approaches are implemented in Agile RE practices in the industry;
- Establish how software engineers manage client security requirements.

INTERVIEW QUESTIONS: Guided questions for stakeholders of the Agile Project

1. Researcher: What is your role in the Agile Software Development project?

   Respondent: ______________________________

2. Researcher: Provide a detailed explanation of how the requirements were elicited for this project?

   Respondent: ______________________________

3. Researcher: How did you ensure that all viewpoints were catered for during requirements engineering of this project?

   Respondent: ______________________________

4. Researcher: At what stage in the RE process do you elicit security requirements?

   Respondent: ______________________________

5. Researcher: What are the other non-functional requirements identified and when were they identified within requirements engineering processes?

   Respondent: ______________________________

6. Researcher: What security knowledge sources were referenced for this project?

   Respondent: ______________________________

7. Researcher: There are several known Security Requirements Engineering (SRE) approaches in security literature, such as Misuse Cases, CLASP, Secure TROPOS, Anti-Models, Abuser stories, *etc.* Have you implemented any structured SRE approach in this project?

   Respondent: ______________________________

8. Researcher: Who was responsible for identifying security requirements in this project?

   Respondent: ______________________________

9. Researcher: How knowledgeable are your software engineers experts with regard to secure software development? What level of training do software engineers receive when working on this project?

   Respondent: ______________________________

10. Researcher: If the software engineers are non-security experts, explain some challenges experienced by them when factoring security into the system.

Respondent: ____

11. Researcher: Discuss your SRE approach.

Respondent: ____

12. Researcher: Comment on customer involvement in SRE and at what stages in RE did they get involved.

Respondent: ____

13. Researcher: What was the main focus of your security requirements engineering approach for this project and can you remember some of the security requirements identified?

Respondent: ____

14. Researcher: List the detailed steps involved to elicit and analyze the security requirements for this project. In your answer discuss identification of assets and risk assessment.

Respondent: ____

15. Researcher: How were threats/vulnerabilities identified, rated and prioritized in this project?

Respondent: ____

16. Researcher: What role did the client's security policy play in your SRE approach?

Respondent: ____

17. Researcher: What would you say are the benefits of your SRE approach?

Respondent: ____

18. Researcher: Who ranked the priority of requirements in this project?

Respondent: ____

19. Researcher: Describe the process involved in ranking requirements and the role of security requirements in this process of the project.

Respondent: ____

20. Researcher: Are you satisfied with Agile RE practices employed for this project in general? Motivate.

Respondent: ____

21. Researcher: Do you have any suggestions for improvement of security requirements in Agile RE practices in this project?

Respondent: ____

THE END

Appendix 2

Company Profile

No.	Size	Revenue	Location	Target Market
1	Less than 25 staff	R20 + million	KwaZulu-Natal	Transport
2	Less than 25 staff	R50 + million	Eastern Cape	Finance
3	Less than 25 staff	R30 + million	KwaZulu-Natal	Finance
4	Less than 25 staff	R15 + million	KwaZulu-Natal	Education
5	Less than 25 staff	R10 + million	Eastern Cape	Manufacturing
6	Less than 25 staff	Not disclosed	Western Cape	Call Centre
7	Less than 25 staff	R60 + million	Gauteng	Medical
8	Less than 25 staff	R70 + million	Gauteng	Medical
9	Less than 25 staff	R40 + million	KwaZulu-Natal	Gaming
10	Less than 25 staff	R15 + million	KwaZulu-Natal	Education
11	Bet 25 and 50 staff	Not disclosed	Gauteng	Government
12	Bet 25 and 50 staff	R100 + million	Gauteng	Gaming
13	Bet 25 and 50 staff	R60 + million	Western Cape	Gaming
14	Bet 25 and 50 staff	R50 + million	Western Cape	Wholesale/Retail
15	Bet 25 and 50 staff	R40 + million	Gauteng	Gaming
16	50 and more staff	R100 + million	KwaZulu-Natal	Gaming
17	50 and more staff	R100 + million	Gauteng	Medical

Appendix 3

Sample of Interview Responses

Question	Company	Response
1	2	I am a team member and Junior software developer. I develop web apps in Agile
2	6	You mean for the current project right now? We had a meeting with the client over two skypes. They were in Jhb and and we in Dbn. We talked about what they wanted to achieve and the goals for the system. They wanted to build a Content Management System (CMS) to manage the data that is generated by the app. We discussed what is most important for this CMS. As the client was speaking, we were writing it down on the white board. Besides interviews, the client also sends follow up emails.
3	4	It depends on the type of clients. If it is an institution client, then they would put their viewpoints to the Business Analysis. If it is an end user client, then it would be relationship managers that would visit clients on an ongoing basis (monthly). Also, clients speak to support people. The client can also e-mail us. So, their viewpoints come to us in different ways. For security, we have an information security team. A group of 3 or 4 individuals. They analyze security of the organization. They deal with security from biometrics on the door, making sure the office is locked. We get feedback from them.
4	11	Security is critical and important security checks are discussed early in requirements engineering. We do login authentication. We also can track everything a user does. So if a user is doing something he is not supposed to do then we can track that user via the history. Security requirements required for the project are in the BRD. Their BA must inform the client of security issues. The client must agree to put it into the system. However, security requirements are kept to a min. If there is a security breach, we must know who did it and why.
5	14	Non-functional requirements are identified when there is a need. We focus more on functional requirements. When we do take our first iteration to the customer, we get feedback. They tell us if the app is not as accessible as they want it to be or possibly it is taking too long to load or it's not giving the feedback we want. Based on the feedback, we add more user stories for non-functional requirements.
6	14	No security knowledge sources were referenced for this project. We are expected to keep up with new trends. This is our own initiative. Security is not the focus. Security is more on the back burner. Security is more implied but not much is done. It is not something that we are required to know. There are lots of mumblings about it but it never gets full attention in a development. No, I haven't heard of Open Web Application Security Project (OWASP) and Common Weakness Enumeration (CWE),
7	17	No known approaches in security. The security team is aware of the latest trends. The security team must develop a framework for security.
8	11	The Business Analyst.
9	2	No – software engineers are not security experts. They do not receive training on a regular basis. I previously worked for a larger company that had a security team with specialist knowledge.

Question	Company	Response
10	16	If developers look at security, they can use the templates available for security. There are so many templates available. Those templates are already prescribed before and the developers follow. There are templates for security, as well. No time is wasted checking for security breaches when there are already prescribed templates for the developers to follow. There is a checklist. The big challenge in companies is that security knowledge is based on an individual's personal knowledge.
11	16	One approach is to use security frameworks. For example, if you are working on a particular module like login and you can use it for all projects. You don't spend the time developing security as you can use the framework in your projects like libraries. You leverage from it instead of worrying about security for each project. I know about pre-written templates for IIS with a plug in. It will check for SQL injection and malicious scripts. Another approach is to outsource security. We also develop software for different countries. The software must meet the regulations in each of these countries before it is deployed to those countries. The software must fulfil security requirements. For example, for online payment there is a security company called e-cobra (number one company) that checks if security standards are met. They will check that, for example, if you are hashing e-mail addresses. They must not just be stored in the database, they must be hashed. These things are checked and certified before deployment.
12	4	The customer gives the functional requirements. No role for customers at this stage. The BA identifies how secure it should be and the level of security that should be there. The customer may only request role based authentication (login and password), *etc.* but those things are already in. Not much coming from the customers in terms of security, because security is a non-functional requirement.
13	6	Requirements were drawn up based on the specifications. No security requirements was specified other than login passwords.
14	16	No formal risk assessment. When we get requirements from the customer, these are high level requirements. They don't break it down. After that meeting is ended, we now have a team meeting. At the team meeting, we break everything down in terms of all the technical aspects. Then each person is assigned tasks. You look at all the concerns for your task. The team member does his own risk assessment, for example, this feature must not be accessible, *etc.*, *etc.* The team member decides what vulnerability he wants to mitigate against. Risk assessment is not done collectively. It is done in isolation with the team lead and the developer who is assigned that feature. If cross site scripting is a concern of the team member, he takes it to the team lead. The team lead makes all the technical decisions. He tells us what to worry about. All security decisions are at the team lead's discretion and the customers' requirements.
15	3	No, not done by us developers.
16	3	They have their own authentication servers which will house the website. Yes, they do secure these servers. I suppose their security needs from their policy are somehow factored into the product.
17	5	Features are developed based on what the customer asked for. We look at functionality. The top priority is what feature the customer wants. Security comes with a standard framework that developers use. The benefit of our approach is that it saves time.

Question	Company	Response
18	17	Management decides on the rank of a requirement based on revenue clients deadlines. That determines our priority. Consensus in a group (management team). Not just one person. We use TFS. We log in requirements there. I'm not sure of any requirements prioritization software used for this process. Security requirements are non-functional and always have low priority.
19	3	If we have a timeframe to 'go live' then the project managers must be able to prioritize what they can do. We use smart sheets to manage the project. Everything is stored in the cloud. This also serves as a tool to rank requirements. If you worked with google docs then its functionality is similar. Everyone can edit the same document. They create columns to rank the priority of the requirements. Some requirements can't be done because of time constraints. These requirements are pushed into support after 'go live' and come in as an enhancement. The project manager ranks the requirements. He is assisted by the product manager and the implementation executive. All these in-house people make all the decisions. The client is not involved.
20	2	We should identify core requirements. We should also inform clients of security vulnerabilities. Developers must be more open and educate clients on security issues. Clients must be able to specify in the requirements engineering phase what security requirements should be in the system. Developers must now work with these security requirements. Maybe developers should work on a lightweight security approach.
21	17	In my experience as a software developer I have worked for eight different companies. What I found out is that software development very rarely starts from scratch. We are always working on existing systems and security is already taken care of. The software is already there we are just adding on features. We make a profit if we can sell the same product to many clients. One should also avoid using a SQL server. We use a hierarchical database. We use an object oriented database with chronological files. Also building lots of prototypes is important, because only when the client views the prototype can he make changes to the system. During requirements engineering it is important to talk about the hardware being used. Having a protocol where the database administrator goes in and physically changes the password every 12 hours. When someone leaves the company they cannot access the database. Amongst other advantages, cloud based solutions have additional security. Software is stored on a server on the cloud. Credit card information is stored offline. Your site will divert payment to an outsourced company. This company will have security in place. Also, it protects your company from lawsuits. You don't want your developers having access to other people's credit card information. Archived records are encrypted. If someone steals the database they cannot access the records. Also, it will be great to provide developers with more security training and access to security knowledge sources.

Appendix 4

Coding Scheme

Interview Questions	Codes and Attributes Associated with Specific Questions	Sub-group or Category
1. What is your role in the Agile Software Development Project?	*Developer*	n/a

2. Provide a detailed explanation of how the requirements were elicited for this project?	*Elicitation techniques tools;*
	Specific teams;
	Customer Involvement;
	Role of Business Analyst;
	Core RE activities;
	Communication;
	Text Documents
		Elicitation of Requirements
		Customer Involvement
		Output Documentation

3. How did you ensure that all viewpoints were catered for during requirements engineering of this project?	*Viewpoints;*
	Customer Involvement;
	Meetings;
	e-mail;
	Tools and techniques;
	Role of development
	Specific teams
		Customer Involvement
		Role of Business Analyst
		Communication
		Output Documentation

4. At what stage in the RE process do you elicit security requirements.	*Stage of Identification* *Core RE activities* *Security Policy*	Elicitation of Requirements Security Requirements

5. What are the other non-functional requirements identified and when were they identified within requirements engineering processes?	*Types of non-functional requirements;*
	Customer feedback;
	Scalability;
	Performance
		Non-functional Requirements
		Customer Involvement
		Elicitation of Requirements

6. What security knowledge sources were referenced for this project?	*Types of knowledge sources; Latest security trends; Other Teams*	Security Awareness
7. There are several known Security Requirements Engineering (SRE) approaches in security literature such as Misuse Cases, CLASP, Secure TROPOS, Anti-Models, Abuser stories, *etc.* Have you implemented any structured SRE approach in this project?	*Known SRE Approaches;* *Security frameworks* *Security Policy*	Security Approaches

8. Who was responsible for identifying security requirements in this project?	*Customer involvement;*
	Role of Business Analyst;
	Team Lead;
	Tools and techniques
		Security Awareness

9. How knowledgeable are your Software Engineers experts with regard to secure software development? What level of training do Software Engineers receive when working on this project?	*Security training;* *Security knowledge*	Security Training

Interview Questions	**Codes and Attributes Associated with Specific Questions**	**Sub-group or category**
10. If the Software Engineers are non-security experts, explain some challenges experienced by them when factoring security into the system.	*Security knowledge; New frameworks; Methodologies; Security Information sources*	Security Awareness
11. Discuss your SRE approach.	*Customer involvement; Onsite customer*	Customer Involvement
12. Comment on customer involvement in SRE and at what stages in RE did they get involved.	*Security methodologies; Customer Involvement*	Security approach
13. What was the main focus of your security requirements engineering approach for this project and can you remember some of the security requirements identified?	*Security training; No approach Security requirements*	Security Training
14. List the detailed steps involved to elicit and analyse the security requirements for this project. In your answer discuss identification of assets and risk assessment.	*Customer involvement; Technical aspects; Diagrams; Methods risk assessment*	Security requirements; Security Risk Assessment
15. How were threats/vulnerabilities identified, rated and prioritised in this project?	*Testing; Tools*	Security Tools Security Testing
16. What role did the client's security policy play in your SRE approach?	*Security policy*	Security Policy
17. What would you say are the benefits of your SRE approach?	*Benefits*	Benefits of security
18. Who ranked the priority of requirements in this project?	*Customer involvement; Team effort*	Roles in prioritisation of requirements
19. Describe the process involved in ranking requirements and the role of security requirements in this process of the project.	*Prioritisation techniques; Important criteria; Security requirements; Role of security requirements*	Prioritisation techniques Important criteria
20. Are you satisfied with Agile RE practices employed for this project in general? Motivate	*Satisfied RE practice; New technology; Changing requirements*	Agile RE satisfaction Challenges
21. Do you have any suggestions for improvement of security requirements in Agile RE practices on this project?	*Security engineering; Best practices; Inform clients of security; Fatigue and burnout; Vendor products*	Best Practices Challenges Team Dynamics

Appendix 5

Extract of Code Definitions

Code	Meaning
Elicitation Techniques	Methods used to extract requirements from customers. These include techniques such as interviews, focus groups, team meetings and brainstorming.
Specific Teams	These are teams within a software development company such as security team, testing team, integration team and exploration team.
Customer Involvement	The role of the customer in the Agile Software Development Team.
Role of Business Analyst	The tasks that a business analyst undertakes to get customer requirements.
Core RE Activities	These activities are elicitation, analysis and prioritization of requirements, identification of security requirements, requirements management and text documentation.
Communication	The methods used to communicate with customers to elicit requirements.
Text Documents	Documents that are the output of requirements engineering.
Viewpoints	Capturing the views of different users of the system.
Meetings	All types of meetings for developing the software.
Tools and Techniques	The different tools and techniques used in requirements engineering.
Role of development Team	Tasks undertaken by software developers.
Stage of Identification	The stage at which security requirements are identified.
Security Policy	The approved policy of the company on the security of its assets.
Non-Functional Requirements	These requirements are not stipulated by the customer but are needed to build high quality products. They include performance, security, number of lines of code, *etc.*
Customer Feedback	After an iteration the customer is required to provide a review of the work carried out.
Scalability	Ability of a software product to perform well when more users are added to the system.
Performance	Non-functional requirement that measures the effectiveness of the system.
Known SRE approaches	Known Security Requirements Engineering approaches that are found in the literature.
Security knowledge	Knowledge of the software development team on building secure products.
Security Risk Assessment	Risk assessment conducted to determine vulnerabilities of the system.

Received: September 2019
Revised: April 2020
Accepted: May 2020

Contact addresses:

Nalindren Naicker
Durban University of Technology
Durban
South Africa
e-mail: naickern@dut.ac.za

Manoj S. Maharaj
University of Kwa-Zulu Natal
Durban
South Africa
e-mail: maharajms@ukzn.ac.za

Nalindren Naicker was born in South Africa in the city of Durban. Education background is as follows: PhD [information systems & technology]; MSc [information systems]; Hons BSc (computer science); BSc (computer science). He currently serves as head of the Information Systems Department at the Durban University of Technology in South Africa. His last publication was a journal paper titled "Change-Point Analysis: An Effective Technique for Detecting Abrupt Change in the Homicide Trends in a Democratic South", which was published in the Scientific World Journal in April 2020. He is currently involved with the supervision of PhD and Masters students at the Department of Information Systems. He is a member of the ICT and Society Research Group for the Faculty of Accounting and Informatics.

Manoj S. Maharaj is a Professor of discipline of information systems & technology in the School of Management, Information Technology and Governance at the University of KwaZulu Natal. Prof. Maharaj is an academic of more than 30 years. He has extensive experience in the management of higher education and training institutions. This includes legislation policies and procedures related to state-owned enterprises (SOEs). He has a deep knowledge of the ICT sector and has researched and published extensively in this domain. He has also contributed significantly to academic capacity development.