Key Challenges with Software Product Requirements

Metropolia University of Applied Sciences

Master of Engineering in Business Informatics

Master's thesis

Date: 25 May 2020

Thesis Author Thesis Title Number of Pages	Vengalarao Sadhineni Key Challenges with Software Product Requirements 50 pages
Degree	Master of Engineering
Degree Programme	Business Informatics
Thesis Advisor	James Collins, Senior Lecturer

This thesis focuses on the current challenges with the requirements process in the software product development. The current trends indicate that the organisations are focusing towards the most popular Agile and Dev-Ops software development methods and still there are some organizations following traditional waterfall and V-Model. This thesis mainly focuses on identifying the key challenges with current requirements process and propose the best learnings and practices while using agile software development method.

An applied research method was applied starting with identification of the business problem, defining the objective of the study, conducting current state analysis with a survey to gather information and ideas on the current software development methods, literature study with most relevant topics related to the business problem and building a proposal with the existing knowledge.

The common business problem in the current SDLC phases for both the development and testing teams are the requirements, the study discusses the current processes and challenges through many of the industry experienced people, took their valuable suggestions to draw some conclusions based on the similarities from various experts.

The outcome of the study is a set of best practices and improvements from the current industry and the literature to effectively manage product requirements for achieving quality product with increased business value and timely delivery.

Keywords	Agile Manifesto, Requirements Engineering, Elicitation techniques, Software Development, Requirements Validation, Collaboration

Acknowledgements

The dream of perceiving my master's degree right through my childhood is achieved with this final piece of my work on this thesis document. This was a hard journey with lot of emotions both personally and professionally but at the end gives me lot of satisfaction. I believe in continuous learning and apply all the learned thoughts into our daily activities.

Firstly, I am extremely grateful to my thesis advisor who had given me motivation and relentless support throughout my journey with thesis. His advice and kind words are invaluable. I would also like to extend my sincere thanks to all my course instructors and support staff at Metropolia University of Applied Sciences who helped me to perceive my master's degree in Business Informatics.

Secondly, I would take this opportunity to express my gratitude to Mr. Viswanath Tadepalli, Project Manager at Capgemini for taking his time to validate the thesis proposal by providing his valuable feedback and suggestions.

Further, I would like to appreciate and special thanks to the fellow students for the co-learning and knowledge sharing. Sincere thanks to my ex-colleagues and friends who spent some time in providing valuable insights to my thesis work. I would also like to express my gratitude at all the authors of journals, books and other sources of literature study which helped me to acquire knowledge for the purpose of the study.

Finally, my heartfelt thanks and deeply indebted to my family members who encouraged and gave emotional support during the toughest times of this journey.

Vengalarao Sadhineni
Helsinki
25 May 2020

List of Figures:

List of Tables:

Contents

1 Introduction

This chapter is intended to provide a basic overview of the thesis topic.

Section 1.1. provides the background of requirements engineering
Section 1.2 states the problem statement of the thesis topic.
Section 1.3 describes the research objective
Section 1.4 details the research outcome

1.1 Background

The purpose of this thesis is to highlight the key challenges of the requirements engineering in the most popular agile software development. Many often feel that the requirement engineering is just a collection of information and documenting for the purpose of sign-off to meet the project milestones. This is not true and there are quite many challenges especially due to the dynamic nature of the requirements over time. Th requirements should be documented in a systematic way with the right tools, processes and stakeholders. This will give more confidence and increase the success rate of the projects.

"We believe that agility could also be used in multiple ways— in everything we do. In fact, the world is changing very quickly around us, so much so that we cannot afford anymore to have projects taking two to five years to deliver, because, during this time, the initial requirements have changed."

- ***PHILLIPPE HUSSER*** *Senior Partner, Progress Direction Michelin*

This topic of requirements engineering is chosen due to the whole product delivery depends on the quality of requirements. Generally, the requirements are done in the very first phase of the software development life cycle. Most of the projects are terminated during the initial phases itself due to the inefficiency of requirements analysis. Most of design failures and project delays are found due to the ambiguity and changing requirements. This leads to the affect a lot on both the customer as well as IT organisations leading budget.

1.2 Problem Statement

For any IT organisation, it is very important to sustain with the competition in the global market. Time to Market is the key for any product delivery, unable to deliver to the market at the right time will lead to customer dissatisfaction and soon the product will be outdated with features and technology. It is essential to define a clear, precise and testable requirements for a successful timely delivery of a quality software product. To achieve this, the requirements process should be well understood by all stakeholders complimented with tools and techniques. In the current IT organisations, the continuous change and re-prioritization of the requirements are very challenging and hence many organisations adapt to the agile development approach.

With the ambiguity of the product requirements, it is hard to translate into a working software product as expected by the customer. Some of the significant implications due to unclear requirements thereby unable to meet the expectations and timelines are as follows:

- Customer needs are not met
- Causes business disruption
- Misinterpretation leads to design issues
- Delayed Time to Market delivery
- Increases re-work and budgets

1.3 Research Objective

The objective of this thesis is to identify the current challenges and propose the best practices to deal with the uncertain product requirements. The current state analysis was planned with a questionnaire distributed to people working in the IT industry to gather the current challenges, practices and ways of improvements. A literature study will be done on the same topic in addition to the feedback from the current state analysis to build a proposal with process improvements and suggestions.

1.4 Research Outcome

The research outcome of the thesis is to propose a set actions to improve the requirements process from the current industry best practices to the key challenges with uncertain product requirements.

2 Research Methodology

This chapter provides an overview about the research methodology used to define and develop the problem statement, objective, current state analysis techniques, literature review and a proposal in a phased manner.

Section 2.1 provides an overview of the research approach
Section 2.2 presents the research design and research plan
Section 2.3 explains the research analysis
Section 2.4 outlines the research proposal

2.1 Research Approach

The research method used for the study is qualitative survey approach and online interactions with some of the respondents from the survey. The survey is generally classified under quantitative research approach, but the qualitative survey used for this study consists of open-ended option to each question for the respondent to express their thoughts. With the author's rich experience with IT industry and performed various challenging roles, a set of problematics areas are listed in the key phases of the software development life cycle namely Requirements, Development & Test phases. The design of the questionnaire is explained in section 3.1 of this document.

2.2 Research Design

The research design consists of five stages as shown in below figure to carry out the thesis study in a logical way to achieve the outcome to the problem statement. The objective of the study is derived from the problem statement. The current state analysis is elaborated in the chapter 3 of this document to identify the key findings from the study and co-relate with the best practices from the literature review. The key findings from the current state analysis and the best practices from the literature study are combined to build the proposal with improvements and suggestions to the problem statement.

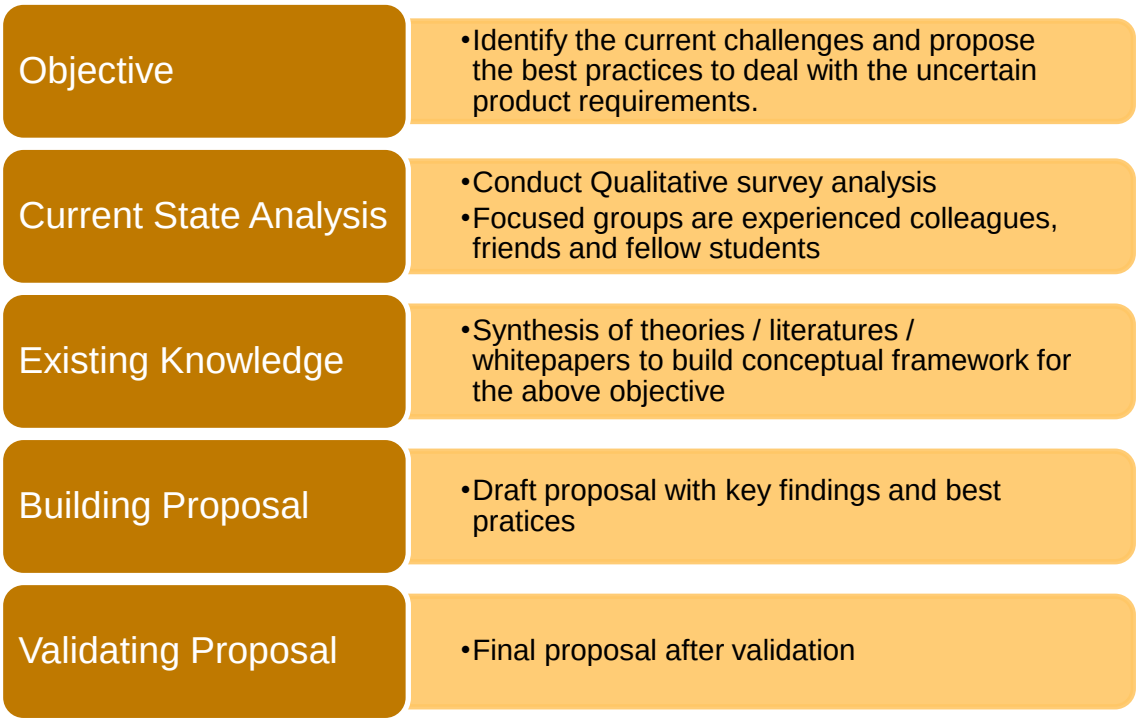


Figure 1: Research Design

2.3 Research Analysis

The data collection and analysis plan were prepared with all the stakeholders and sources involved as shown in the table below.

Input data to	Focus	Data Type	Source	Outcome
Current State Analysis	Identify the key challenges with software product requirements	Own experience Literature	Author Internet	Survey questionnaire
Building Proposal	Identify the key challenges with software product requirements	Survey responses Literature	IT professionals Internet	Initial proposal with solutions to the key findings from CSA
Validating Proposal	Review of the Initial proposal	Feedback	Project Manager	Final Proposal

Table 1: Data Collection and Analysis Plan

The research analysis was carried out with the qualitative survey and this survey do not focus on the numerical data. The survey was distributed to the industry experienced professionals and the responses were sought. The detailed response analysis is done for the survey questionnaire in section 3.2 and the graphical view of the responses can be found from Appendix C of this document. The analysis of the data is then interpreted to find some key patterns and the valuable insights from the qualitative data collected. The responses are then classified into key findings as detailed in section 3.3 of this document.

2.4 Research Proposal

The key findings from the current state analysis is the basis for the literature study. A detailed retrospective analysis is carried out from the current industry best practices and learnings. After the current state analysis is completed, the responses are analysed to group into common areas and identify the key findings. The key findings and the proposal are elaborated in section 3.3 and chapter 5 of this document.

3 Current State Analysis

The current state analysis for the thesis problem statement had been carried out using google forms survey and the responses were collected from the focussed groups and network of experienced professionals. The questionnaire was sent to the individuals using LinkedIn, WhatsApp and Facebook Messenger and followed up to get the responses on time. The main objective of this survey was to understand the current challenges faced with the software product requirements during the software development cycle.

This section outlines the design of the questionnaire, analysis of the responses and key findings in the sub-sections 3.1, 3.2 and 3.3 respectively.

3.1 Design of the questionnaire

The questionnaire was designed based on the work experience of the author and some references to the literature study on the current industry challenges with respect to the most popular agile development method. A check list of the current and past incidents was prepared by visualising the problems faced by various stakeholders in the whole software development and delivery process. There is no anonymity of the respondents for the survey and the email ID was collected so that clarifications or discussions can be initiated further based on the insights. After the questionnaire was designed, it was sent to couple of my ex-colleagues to test the survey and their feedback was updated before it was distributed to larger audience.

The following types of questions were designed to get insights for a qualitative analysis and all these questions had an option for additional comments from the respondents. Hence there were no close ended questions designed for the survey.

- Single Select options are the questions where the respondents can choose any one of the choices provided. These are basically a close ended question but for this survey an additional short free-text option was provided to get the respondents any other feedback.
- Multi-Select Options are one of the best choices to make respondents happy while providing feedback. This type of questions requires more in-depth analysis and thinking while designing the question. All possible options must be provided, and this also sets the expectation for the response analysis in the direction of the study.

- Likert Scale helps to get the perception of the respondents and many times helps to understand the emotions of the respondents due to their current challenges. This is also helping to identify the areas for improvements depending on the satisfaction levels of the respondents.
- The Open-ended questions opens the discussion and the respondent are free to express the opinion in a long text. This is one of the best types for qualitative analysis but at the same time the respondent may not be interested if there are too many such questions.

The questionnaire had 25 questions for qualitative analysis where each question had multiple options given and had a last option for the respondents to express their own views to the question. The questionnaire is also aimed to save the time for respondents with appropriate options already given for 24 questions and this also developed quick thinking while answering the questions. The last question is open ended, and the respondent can provide his views on the current challenges and improvements needed on this thesis topic.

All the 25 questions were made mandatory and the respondents had an option to add their own input for all the questions. These 25 questions are classified in to three sections as in the below table.

Sections Name	**Questions Range**	**Single Select**	**Multi-Select**	**Likert scale**	**Open ended (Long Text)**
General	Q1 – Q9	4	4	1	0
Knowledge, Skillset, Tools	Q10 – Q14	3	1	1	0
Quality and Process	Q15 – Q25	3	1	6	1
Total	**Q1 – Q25**	**10**	**6**	**8**	**1**

Table 2: Questionnaire Classification

The survey questionnaire can be found in Appendix B of this document.

3.2 Analysis of the responses

The survey had responses from thirty experienced software professionals who were in different countries and working in multiple domains. Each respondent had answered to all the thirty questions as there were no optional questions to skip. The response analysis of each question can be found in the below table. The graphical representation of the responses can be found in section 6.3 of the Appendix.

No.#	Survey Question	Response Analysis
Q1	Your primary areas of work or experience?	This question was designed to understand the focus of business sector for the study. As expected, 50% of them work in Banking & Finance, 23.3% in Health care, 20% in Insurance and the remaining 6.7% worked in other sectors.
Q2	Choose your experience level in the software/product industry	This question recorded the experience levels of the respondents. From the below responses, 93.3% of the respondents has more than 5 years of experience which is interpreted as the quality responses who already experienced the challenges with the requirements. • More than 15 years – 30% • Between 10-15 years – 40% • Between 5-10 years – 23.3% • Less than 5 Year
Q3	Do you think that the requirements process can be same for all type of software products irrespective of its size and complexity?	63.3% responded that the requirements process depends on the size and complexity of the product while only 16.7 opined on the contrary. 20 % of them felt that the same process may be used sometimes.
Q4	You had worked or currently working with	This question was designed to understand if there were some respondents who worked on multiple teams and the responses indicate that many of them do work. Majority of them work in the Requirements (50%), Project Management (53%), Development (63%) and Testing (60%) areas.

		This gives confidence that the responses were coming from the main streams who are mainly dealing with requirements management.
Q5	Software development model followed in your organization/project?	About 67% of the respondents were dealing with agile development while only 10% each on Waterfall, Dev-Ops and V-Model. These statistics are evident that the current focus of improvements for the study may be confined to the Agile development.
Q6	Do you think that the end user needs are translated into quality requirements based on your experience with delivery of software product?	70% say that the end-user requirements are translated into quality requirements while 10% don't agree with this statement at all and interestingly 20% say may be when focused attention given with quality measures. There is one additional comment received that creating the fantastic customer journeys made it possible to achieve the objective of quality requirements.
Q7	Do you agree that the requirements must be clearly documented with examples, illustrations and user stories etc.?	While majority of the respondents with 93.4% (76.7% strongly Agree and 16.7% Agree) in the Likert scale of 1-5 that the requirements must be clearly documented, only 6.6% of them strongly disagree to the statement. This is a clear contradiction to the previous responses to Q5 and the reason may be due to the fact that many teams in the organization adopted some of the agile methods like scrum and sprint planning but in terms of dealing with the requirements still follow the traditional methods of documentation.
Q8	What are the best ways of conducting requirements elicitation and analysis phase according to you?	The respondents have almost voted to all the multiple choices options given, brainstorming (90%) and workshops (86.7%) scored the highest while other options online meetings, experiments and observations also score about 50%. There was one additional response from the respondent saying customer testing and focused groups which is a very valid in

Q9	What do you think, the collaboration between the requirements, development & testing teams be during the whole product life cycle?	Majority of the respondents felt that Continuous walk-through sessions with Dev & Testing teams (90%), Testers feedback & impact (73.3%), Requirement support (60%) and Feasibility check (56.7%) are very essential collaboration ways for an efficient product delivery. The responses to this question embark the need for the continuous collaboration between requirements, development and testing teams.
Q10	Are you satisfied with your requirements team knowledge and competence in defining quality product requirements?	On a Likert scale of 1-5, only 16.7% strongly agree and about 30% of them stayed neutral to this question. Only 50% of them felt that the requirement team possess required knowledge and competence, and this is clearly below par. This is clearly an area of improvement for the requirements team to build strong knowledge base and competent levels otherwise the whole product delivery suffers due to misconception or incomplete handling of requirements.
Q11	Do the requirements team make efforts to understand the end user needs and estimate the product feasibility and design constraints well in advance?	50% of the respondents felt that the requirements team make proactive efforts while 33.3% felt may be sometimes. Only 16.7% felt that no proactive efforts done by the requirements team.
Q12	Is your requirements team able to communicate the end user expectations to the development and testing teams clearly?	60% of the respondents felt that the communication between the teams are clear while 33.3% felt may be sometimes. Only 2% are not satisfied with the communication.
Q13	Have you used any of the tools for maintaining product requirements?	Only 56.7% of the respondents are using the tools for product requirements while 43.3% of them don't use any tools apart from documents. In this digital era, this is far behind the number of tools available and this is a major disadvantage

		that the traceability maintenance for the requirements and test coverage becomes complex and in-accurate due to manual traceability.
Q14	What are the benefits and values in using tools for requirements?	More than 73% of them felt that requirement maintenance, traceability & coverage, Efficient reviews and version control are the benefits and values in adopting tools for the requirements definition. About 27% of them also felt that the e-signing also can be handy but low score for this may be because not all the organizations still don't approve individual requirements, but they are done in bulk.
Q15	What are your current challenges with respective to your requirements analysis phase which impacts both development and testing?	The current challenges for the development and testing teams with the requirements according to the respondents are rated as follows: • Misunderstanding or Unclear requirements (73.3%) • Integration challenges (53.3%) • Requirements volatility (50%) • Conflict requirements (46.7%) • Lack of knowledge sharing (43.3%) • No Acceptance criteria (30%) These responses are a clear indication of multiple root-causes that were discussed in Q10 and Q11.
Q16	Do you think the requirements need to be reviewed by all the stakeholders before they are signed-off? Stakeholders include architects, developers and testers.	In a Likert scale of 1-5, the respondents strongly agree (56.7%) and agree (26.6%) that there is clear emphasis of the requirements review by the stakeholders and this would benefit in multiple ways e.g. design constraints and impact assessments gets done at a high level in parallel to the requirements phase. This would eliminate lot of rework at later stages of the project.
Q17	Do you have a standard template for capturing product requirements?	73.3% of the respondents agree that they have a standard template while 23.3% say that they don't have a template. There was 1 additional re-

		sponse – depends on the client. It is recommended to have a standard template customized depending on client and project with clearly defined acceptance criteria. This would be very handy for the developers and testers to validate if they have built the right product.
Q18	Would it be helpful to classify the requirements as mandatory, conditional, future, nice to have etc. so that they can be prioritized for delivery planning?	On a Likert scale of 1-5, 86.7 % of them Strongly agree or Agree that the requirements classification helps to prioritize the scope of delivery while 13.4% of them stayed neutral or less agree, may be due to their project size and the adopted delivery model.
Q19	How often the requirements change even after the design phase and during final testing phases?	The responses for this question not unanimous as 66.7% of them felt that the requirements are changed often while only 33.3% felt that the requirements are not volatile during the later phases of SDLC.
Q20	Do you think that the developers and testers should be given an opportunity to participate in the requirements discussions and interact with customers/end users?	According the responses, it was clearly needed to have regular or iteration wise interactions with the end users to understand the functionality there by enabling them to do efficient planning of their activities. The responses are as follows: • Occasionally (10%) • Conflict resolution (3.3%) • Every iteration (26.7%) • At regular intervals (50%) • Never (10%)
Q21	What is your opinion on the project plan, does it give enough importance to elicitation and analysis of requirements?	Almost all respondents agree that enough importance should be given to requirements phase in the project planning. This is an early indicator for other teams to estimate their scope of work and better understand the functional requirements earlier to their phase start.
Q22	Do you agree to this statement "Poor requirements	This is another unanimous response that almost all (96.6%) the respondents strongly agree or agree to this statement. Hence there is already

	always leads to rework and budget overruns"?	good experience and awareness of the consequences of poorly handled requirements.
Q23	Do you think that breaking down to small/medium requirements helps to deliver a quality product on time?	83.3% of the respondents strongly agree that the breakdown of requirements helps to deliver a quality product while only 6.7% disagree with the statement. 10% of the respondents express that this may be on need basis and the breakdown level.
Q24	Do you think it's important to measure quality characteristics for the product requirements?	On a Likert scale of 1-5, 46.7% strongly agree and 36.6% Agree that the quality metrics are extremely important while 13.3% stayed neutral. Only 3.3% opinions as not important.
Q25	Do you have some suggestions/improvements to make the requirements process more efficient and simpler?	This is an open-ended question to get the respondents thoughts on the current requirements challenges and improvements for efficient handling of requirements in timely delivery of quality product. Brief summary of the areas for improvements by the respondents are grouped as follows: • Tools and trainings • Processes and standards • Collaboration and Involvement • Project Management • Iterative development • Quality of requirements

Table 3: Questionnaire Response Analysis

3.3 Key Findings

The current state analysis responses reveal that the agile approach being followed by many organizations either in full scale or adopting some of the agile values partially. The main goal for the IT organizations is to adapt the changes and deliver the business value continuously to keep the customers engaged. Most of the respondents also expressed that the requirements are documented clearly but still there is some level of misunderstanding to the development teams due to lack of proper communication.

The common challenges faced by most of the respondents are integration and conflicting requirements. These are good examples of missing stakeholder collaboration and incomplete feasibility study during the requirement analysis phase. The survey finds that there is a definite need to improve and enhance the skillset of the requirements team to clearly document the complex requirements in to simple and understandable way. The consolidated view from the survey outcome is that many organizations except startups have their defined requirements processes but they may not be suited for the volatile business needs.

It is recommended to customize their way of working according to their needs and their challenges overcoming challenge by challenge through retrospective actions at regular intervals. The break-down of the requirements in to small and deliverable items according to the business goals is also perceived as an important quality characteristic of a good requirement.

The responses of 25 questions from the CSA survey are analyzed in detail. The top challenges and improvement areas are listed below in table 3. The top 10 responses are then grouped in to six findings and the mapping with the questionnaire reference is provided in the below table. These findings are explained in more detail in chapter 5 of this document.

Finding#	Finding description	QRef#
Finding 1	Clear illustration of requirements – User stories, customer journeys	Q7, Q15, Q22
Finding 2	Use on elicitation techniques – Brainstorming, Workshops	Q8
Finding 3	Improve stakeholder collaboration with walk-throughs and involvement for impacts & feasibility assessment	Q9, Q20
Finding 4	Reusability and efficient requirements management with appropriate usage of tools, templates	Q14
Finding 5	Welcoming requirements change and prioritization to achieve business value	Q18, Q19
Finding 6	Break-down of requirements - Epics, features, backlogs	Q23

Table 4: CSA Key Findings

4 Literature Review

This section discusses about the current industry trends and best practices from the existing literature regarding the software product requirements in an agile world.

Section 4.1 gives a brief overview of Requirements Engineering process
Section 4.2 explains the popular trending Agile Manifesto
Section 4.3 about the requirements practices and techniques
Section 4.4 on the tools and collaboration
Section 4.5 presents the iterative development
Section 4.6 summarises the literature review

4.1 Requirements Engineering

Requirement engineering is a systematic approach to understand and capture the user needs with traceability to each requirement which helps to build a right product that satisfies the user needs and increases the business value. Like Steve Jobs said, *"users don't know what they want until you show it to them"*. No matter how much granularity of requirements there is always a gap the way the end users expect it to work and the way the product is being translated. So, it is not an easy job to describe the requirements for any complex product unless the software engineer understands precisely what to build and the software tester knows how to test.

The requirements phase is a very significant phase and must be done with utmost care which otherwise leads to adverse effects on incorrect design and inaccurate testing of the product. Even a small change at a later stage of the project leads to a big chaos on the re-design and re-testing of the product leading to over budget and missing market delivery timelines. Many start-up companies or small projects don't often have a requirement document in their initial stages but when these grow into larger business groups or projects then the same groups realize the potential need for having the requirements and processes documented in order mitigate the risks and for compliance reporting. (Swebok, 2015)

The requirements engineering is so important today because the pace of the product development drastically increased, and the vibrant customers are expecting the new versions atleast twice a year. Requirements engineering plays a vital role in regulated industries like health care where there is a potential injury or death. (Berenbach et al., 2009).

The process of requirements capturing and finalisation for software development described below on a high level. It is also important that the requirements teams should be aware of the available techniques and best suited process to their development models and the product teams.

Requirements engineering is the first phase in the software development life cycle. It was noticed from my own experience that not all the stakeholders are involved in the requirements discussions, often limited to the requirements team. So, it would benefit for both development and testing teams informed on the regular basis. The concept of requirements engineering constitutes of four stages, namely elicitation, analysis, specification document and validation as shown in figure below.

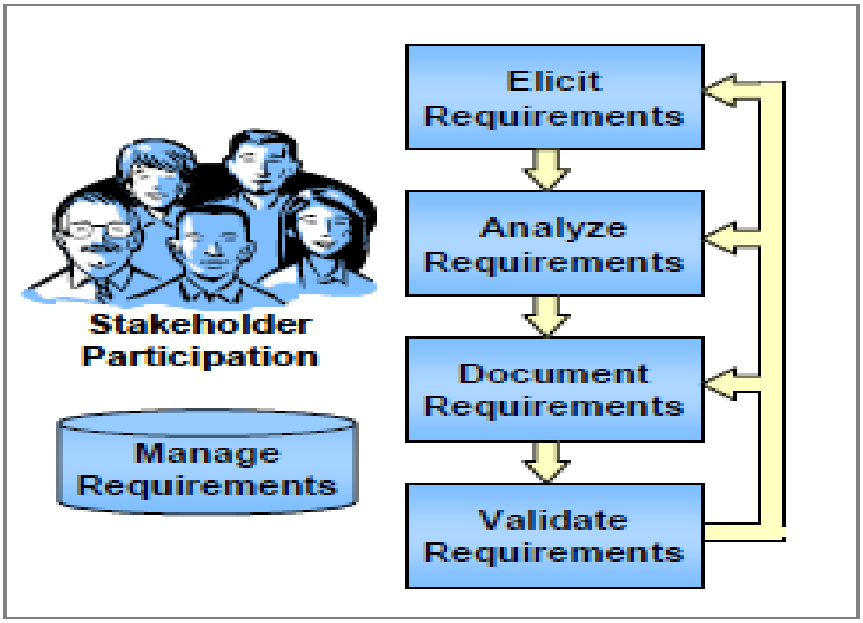


Figure 2: FHWA - Requirement Engineering Activities

The first stage of the requirement phase is the elicitation of the requirements. This is a very important stage where the requirements team needs to understand the end user expectations to transform into an acceptable quality product. “One of the fundamental principles of a good requirements elicitation process is that of effective communication between the various stake- holders” (Swebok, 2015). There are various elicitation techniques, the most popular being Brainstorming, focus group methods as shown in the below figure.

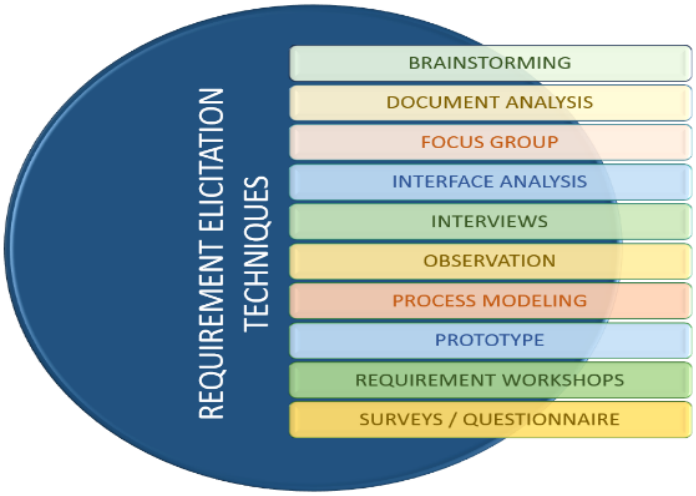


Figure 3: Anarsolutions - Elicitation techniques

Requirement analysis is the second stage in the requirements phase which defines and analyses the feature or problem in depth, develops and evaluate the alternative options to select the best solution for implementation. This stage is an extension of the requirements elicitation and complement each other. The objective of the requirements analysis is to elaborate the system requirements to derive software requirements, identify and find a solution to the conflicting requirements (Swebok, 2015) after a thorough impact assessment with the product engineering SMEs.

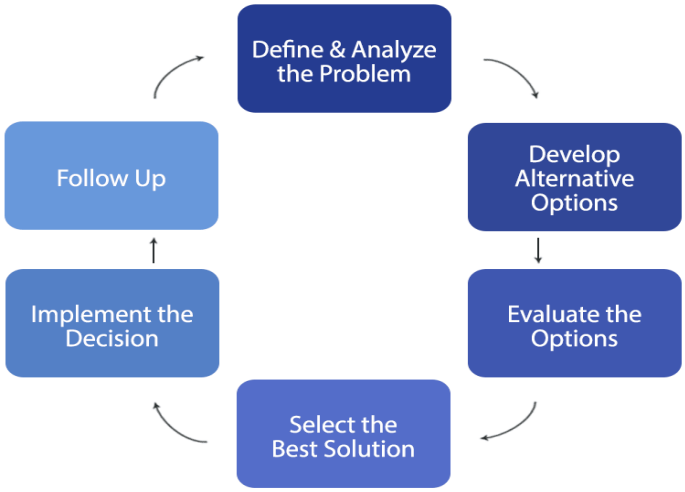


Figure 4: Pipilikosoft - Requirement Analysis

The third stage of the requirements phase is the Software Requirements Specification document that is well defined, evaluated, reviewed and approved by the key stakeholders. This document should be well organized such a way that each requirement is traceable to another requirement. The characteristics of a good specification shown in the figure below.

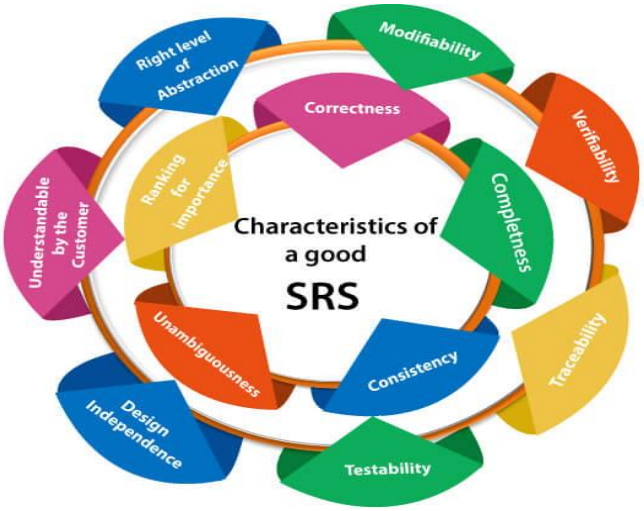


Figure 5: Javatpoint - Characteristics of a good SRS

In a nutshell, the requirement specification contains information about the expected system behaviour, features supported and technical capabilities. The guidelines and instructions for the SRS template can be adopted from the IEEE 830 standards.

Requirements Validation is a process of ensuring that the requirements stated in the SRS document are implementable and testable by the product team. Often this stage is ignored due to time-pressures and budget constraints. This leads to problems identified at a later stage which cause re-work and the cost of fixing a problem increases with the advancement of the product development (Elgabry, 2016). Some of the validation checks are the requirements reviews for consistency, completeness, testability and feasibility.

4.2 Agile Manifesto

In the traditional software development models, the detailed requirements are captured, documented and signed-off before the start of the next phases be it a waterfall or V-model. The current trend with all IT organizations goes towards the agile transformation and this is evident from the KPMG survey on agility that revealed 81% of the respondents have initiated the agile transformation in the last 3 years. This statistic also justifies that

the transformation can be achieved over a period because the agile concepts needs to be digested at the individual and team level. (KPMG Survey on Agility, 2019)

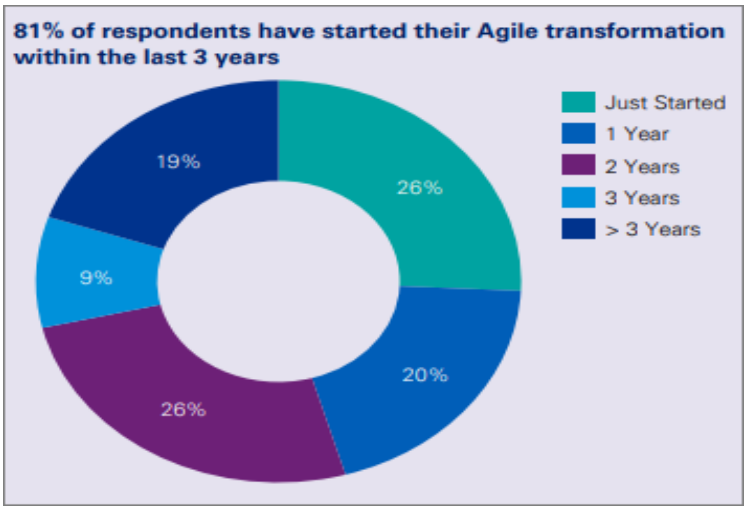


Figure 6: KPMG Survey - Agile transformation

The agile manifesto created by a group of software developers in 2001 with a new software development approach with four agile values and twelve supporting principles as shown in the below figures. Agile values enable the teams to make quick decisions and where possible the teams should avoid rework, duplication and minimise documentation. As shown in the below figure, the left-side items are valued over the right-side items.

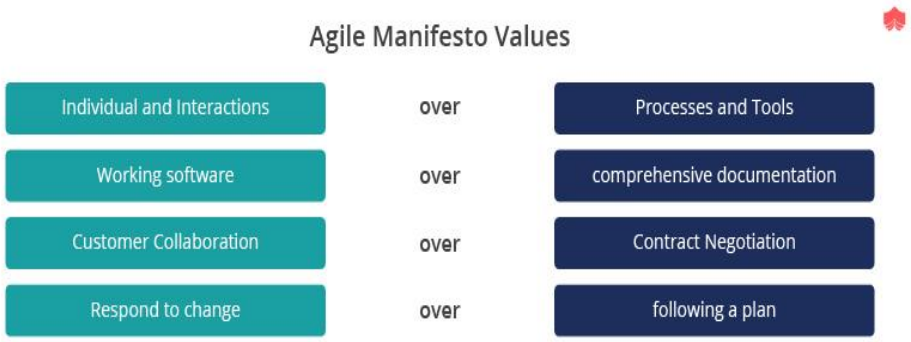


Figure 7: Knowledgehut - Agile Manifesto Values

Nevertheless, processes and tools play a significant part in the agile development model. It adds more value to have skilled and experienced people working together in addition to the tolls and processes. Requirement is all about clear and precise documentation

and the focus should be on delivering the working software. Increased customer collaboration and closely working together will increase the customer trust and confidence and therefore negotiations are easily met. It is good to have a plan, but it gives more value to be flexible to accommodate changes to the plan quickly and to prioritise the business needs for delivery.

Below figure gives a pictorial representation of 12 agile principles which helps and guides the team to follow in their agile development projects.

Agile Manifesto Principles

Figure 8: Knowledgehut - Agile Manifesto Principles

The main objective of the Agile development is to deliver a quality product with high business values and seek continuous feedback from the end users. This is achieved through team collaboration, iterative development and embracing change.

4.3 Iterative development

In the traditional model, the requirements are defined with a flat list of items and there is a high risk that the requirements can change at any phase of the project phase for various reasons like misunderstanding or market compliance or any other reason for that matter. In an agile process, it is done by creating a hierarchy structure and breaking down to smaller requirements which can be incrementally deliverable to gain the business value. Figure 7 on the left side shows the hierarchy in an agile process. The business goals and initiatives are added as Epics and often spread across multiple teams for

delivery. A feature is a deliverable component of the product and is created a back-log item to a project (Microsoft Azure, 2019).

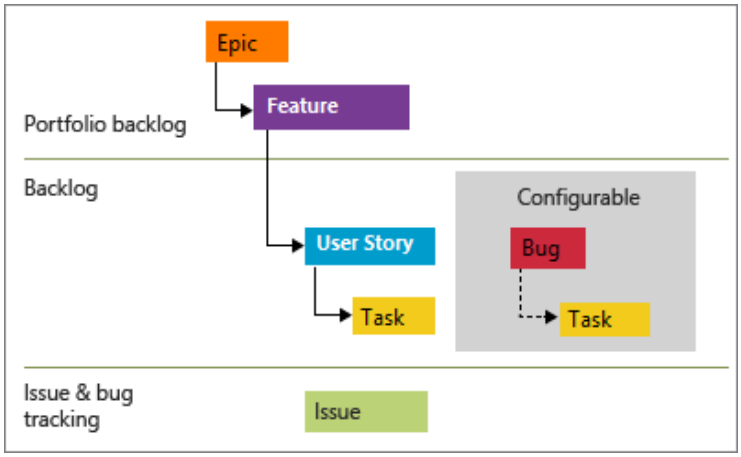


Figure 9: Microsoft Azure - Agile Process hierarchy

The features are then created with user stories and tracked the activities for closure with tasks. The product owner defines the prioritisation of the back-log items for the teams to deliver the product. The agile development is an incremental delivery with rapid cycles and each incremental delivery is tested to ensure the right product is built with quality. The agile process is interactive with all the stakeholders and the agile teams welcome the changing requirements even in the later stages of the development phase. The fact of the matter is that the development teams must invite the changing requirements otherwise there is no business value to the customer or the product. It is still better to change at the same time rather than coming back to the same point at later releases (Chappel, 2012).

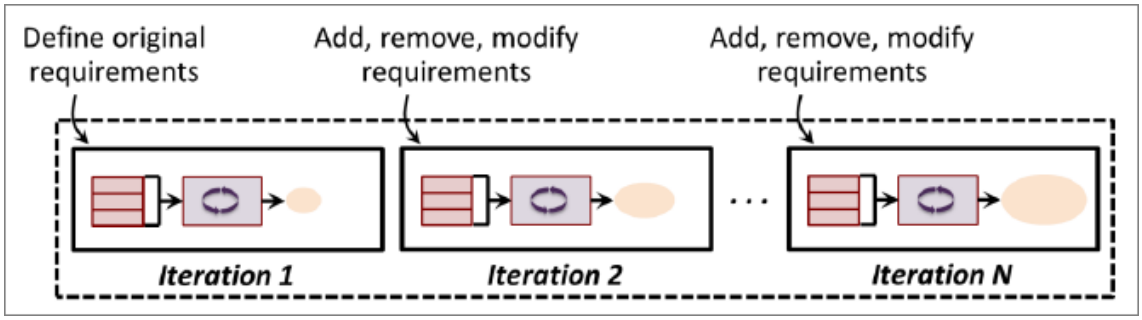


Figure 10: Agile Iterative development

Prioritisation of the requirements plays a key role in the agile development and it is a good practice to start negotiating with stakeholders at the beginning and ensure that the requirement is feasible to include in the product release. The product owner then evaluates if the requirement or feature is worth in the current release or later release. This will

ensure that all the developed work items are acceptable in line with the agreed acceptance criteria with all the stakeholders.

4.4 Requirement practices

As the products are enriched with new features and modification to existing features, the complexity of documenting requirements increases. Sometimes it is also very difficult to describe everything as a text, so it would be better represented with a diagram or a table. There are situations where reverse engineering requirements are done when the software product is already in testing phases, the client acceptance may need delivery of test cases and final requirements. In such, situations there is a very high risk that the software product do not meet the client expectations, and this will lead to contractual obligations (Berenbach et al., 2009).

There are various requirements techniques which can be appropriately used to better suit the need of the requirement type. The requirements can be represented using use case models, story boards, customer journeys along with the elicitation techniques briefly discussed in section 4.1 (see figure 3).

> *"A picture shows me at a glance what it takes dozens of pages of a book to expound."*
>
> *—Ivan Turgenev, Fathers and Sons, 1862*

Despite using the requirement techniques, there are also other factors that can influence the quality of the requirements, mainly when right stakeholders are not involved, failure to collect accurate information and finally failure to understand the customer needs. Misunderstanding the requirements is another aspect and this leads to re-work effort and missing the timelines.

In agile development, product requirements are created in the form of features and user stories which are essential for the release in scope. The five key elements of the product requirements are explained below (Kruger, 2018).

- Purpose – what problems are solved, who uses it and why is it important
- Features – what is being developed to deliver the purpose
- Release criteria – setting the right goals to achieve the purpose
- Timeline – estimate when the purpose is expected
- Review – ensure key stakeholders review the product requirements

4.5 Tools and Collaboration

Most of the small-scale organizations and startups do not invest much into the tools due to their limitations on the budget availability and real value with small teams. The alternatives for these organization are usage of the word or excel for all kinds of documentation for the project deliverables. The challenge lies with the large-scale organizations and complex products in maintaining the requirements, design, test artefacts for traceability, re-usability and version control. This enforces the organizations to use right set of tools to manage requirements, traceability to design and test artefacts. This is quite handy to change any of the existing requirements as the tool provides the linkage to the affected areas. It is extremely important to use right set of tools, preferably a single tool which can be used to trace the requirements, code items, test artefacts.

A single tool also makes the teams collaborate easily and all information can be found in one pace with linkage to different tasks. This also increases the transparency and reduce duplication of work. The success of a product delivery depends on how closely the teams are working together, share knowledge and resolve the constraints through discussions immediately.

It is quite common these days that most of the projects are run with distributed and remote teams. The best way to connect and collaborate with distributed and remote teams is to conduct online video conference meetings using Zoom, Microsoft teams, WebEx etc. These online interaction increases the clear understanding and get more clarity through walk-throughs and sessions. It can be better explained with online and face-face communications whereas email communication is not recommended for detailed knowledge sharing.

4.6 Summary

This section provides a summary of the literature review in the previous sections of this chapter. The requirements engineering process was a very broad topic and lot of information available from the existing literature. The requirement engineering process which is the first and foremost phase in a software development life cycle explains the need to adopt right elicitation techniques like brainstorming, workshops and need for elaborating the requirements through analysis. There should be an emphasis in every organization on defining the quality characteristics of a good requirement, requirements validation before commencing the development and managing the requirements were briefly explained.

Agile manifesto proven to be a successful approach in the current software development with faster delivery cycle to achieve high business value. In agile practice, the collaboration between the stakeholders plays a significant role. The product owner along with agile team defines the features to be translated into software product. The agile values and principles generate business value by delivering a working software product with incremental and iterative development approach with clearly defined acceptance criteria and definition of ‘done’. This is achieved through the break-down of requirements in to small and deliverable work items for a predictable outcome at the end of each delivery cycle. Just having right set of people and tools do not guarantee productivity, it is the collaboration between the individuals and teams work together with combined efforts and innovative ideas that delivers the best results.

5 Proposal and Conclusion

5.1 Initial Proposal

Requirements Engineering is an important phase in all types of software development methods. There is a misconception that the requirements are not documented in the agile process but that's only a myth. Requirements analysis is done for the agile projects but in a more agile way. Unlike traditional models, the requirements documents are updated throughout the software product development phase depending on the dynamic nature of the requirements.

The key findings from the current state analysis is listed in section 3.3. The solutions and improvement areas to the requirements process based on these findings are discussed in this section as shown in the figure below.

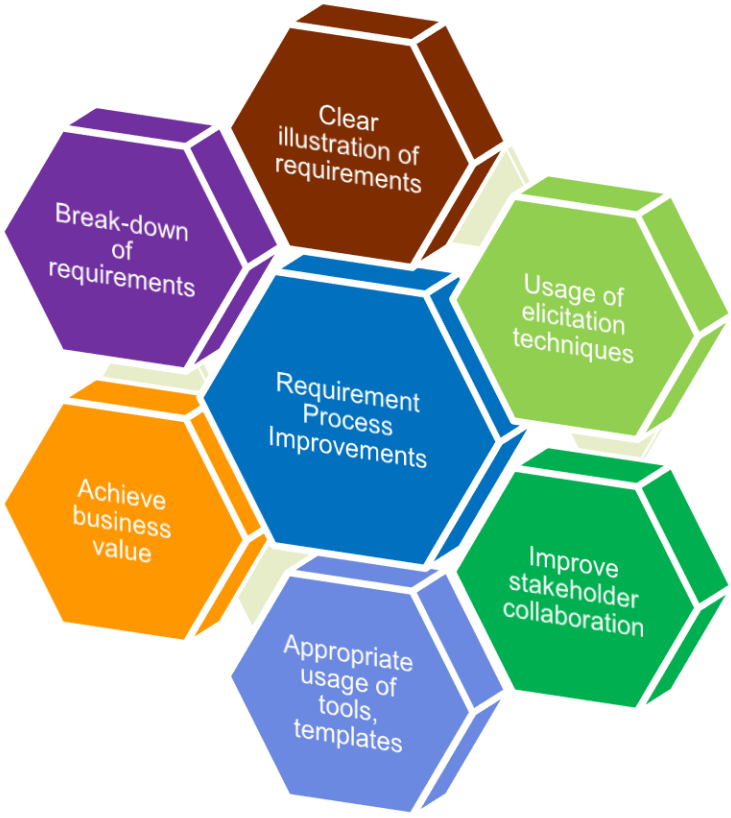


Figure 11: Key Findings

Finding 1: Clear illustration of requirements

The current challenges for many organizations are building a clear and precise requirement that every member can understand and contribute for a quality product. The requirements documentation is not an easy task as converting the customer inputs into verbose is very difficult. To some extent, this can be improved by using pictorial representation, flow diagrams, user stories etc. Some of the key characteristic of a good requirement is to define unambiguous, testable, traceable, clear and feasible requirement, where needed the assumptions, dependencies and out of-scope items are stated.

The level of the requirements illustration also depends on the experience level of the teams. Most of the product organizations has experienced resources and currently following agile methods. The requirements are defined at a functionality level with acceptance criteria defined for better understanding of the expected output and the detailed requirements are documented during the product development and testing. This also benefits for the agile teams to iron out any design issues and involve the stakeholders to intervene for immediate resolution.

It is also a good practice to invite the customer and other stakeholders to walk-through of the developed product for early visibility of issues which can be corrected immediately. If the issues are found at a later stage, then it becomes too complex in terms of processes and the future planned work gets affected.

Summary of Actions:

- Requirements team should get a buy-in from all stakeholders on the level of requirements documented and continuous feedback simultaneously so that the requirements are captured at the expected level by the stakeholders.
- The requirements should be precisely documented with user stories, flow diagrams where required and not to forget mentioning the assumptions, dependencies and out-of scope items clearly.
- Evaluate the requirements testability and traceability, where necessary define the acceptance criteria which makes easy for the team to achieve the desired output.
- It is a good practice to use consistent terminology and standard language to minimize the misinterpretation of requirements.

Finding 2: Usage of elicitation techniques

The common challenges faced during the elicitation and analysis stage is that the right people are not involved and the delayed decision making. From the current state analysis responses and the literature review, brainstorming and workshops turn out to be the best techniques popularly used. It is recommended approach to conduct elicitation and analysis sessions with face-face meetings especially when the teams haven't met before.

In an agile methodology, the elicitation and analysis process go along with the development phase unlike the traditional methods where the whole requirements are defined before proceeding with development activities. Due to the iterative nature of development and testing in agile, the product owner or the business analysts are recommended to continuous work or provide support to the agile teams to avoid any delays in clarifications. This way the product owner or business analyst can ask more information on the current behaviour of the product or if there are any design constraint challenges. At the same time, the agile teams also benefit in getting the clarifications to the requirements immediately and set expectations on the delivery planning. The reverse engineering helps to understand the existing product and update the requirements.

Summary of Actions:

- Adapt to the most suitable and efficient elicitation techniques like Brainstorming and workshops for close interactions. Make use of these elicitation techniques to work more efficiently with face to face interactions.
- Ensure right stakeholders are involved in discussions for quick decision making. Send the invitations early enough and engage them with updates, reminders etc.
- Track the open points to closure in a time bound manner during the elicitation stage itself to finalise the requirements and to further move on to next set of requirements
- Lead the discussions with open minded and set realistic expectations
- Collect evidences (screenshots, sample data etc.) from their current system to elicit the system behaviour

Finding 3: Improve stakeholder collaboration

Stakeholder collaboration increases the success rate of the product delivery. Stakeholders include customers, project teams, managers and sponsors. All the key stakeholders are listed in a common place so that the teams know whom to approach when in need

of any help. Involving the stakeholders early in the product requirements phase for reviews and feedback makes them feel inclusiveness and continuous engagement in the success of the delivery. This will benefit in requirements planning, decision making, invaluable feedback, resolving the conflicting requirements. It is important to invite only the right people to the discussions so that others can make use of the time efficiently.

Stakeholders need to provide continuous support throughout the product life cycle. The team collaboration is a continuous process for product knowledge through walk-through sessions, offline reviews, impact assessments and feasibility checks. The priority of the agile teams and all stakeholders is to achieve the quality product through close collaboration and working together. The collaborative teamwork will yield better results working with a common goal and the individual contributions are also improved.

Summary of Actions:

- Engage the key stakeholders early in the project requirements stage and create an influence to participate the discussions.
- Keep the stakeholders and management updated with the progress with a summary of key points for immediate action.
- Build healthy relationships to increase trust, confidence, resolve stalemate and quick decision making.
- Work towards achieving the common goal of the project

Finding 4: Appropriate usage of tools, templates

There are many tools available to support the software development life cycles, mainly for the requirements management, code maintenance and test management. It is very important to achieve traceability right from the requirement definition to design, coding, testing artefacts for the better assessment of the current state and reusability.

In an agile process, the tools play a vital role so that the agile teams can focus on the actual development and testing of the deliverable. Manual creation of such artefacts can be a very tedious work and mostly they cannot always be handy. As the complexity and size of the product increases, there is no replacement for the tools.

Depending on the existing process and tools used, the organisation needs to evaluate for the improvements. The ideal way is to use a single tool for all the SDLC phases.

Summary of Actions:

- Single tool recommended to minimise the re-work effort, re-usability and bi-directional traceability. The key traceability items are requirements, design, code items, review and test items.
- Define standard templates where the usage of automation tools not possible
- Improvise the tools and templates for efficient usage and simplified process

Finding 5: Achieve business value

There is not much business value seen in the traditional development approaches mainly for not accepting the changing requirements. The organisation and teams following the agile process are welcoming the requirements change and prioritisation because this will add a great value to the product and the customer. The product owner defines the requirements and prioritise them in the product backlog. Even if there is a change in the requirements, it is convenient for the agile teams to adopt to the change even at the flag end of the project. This is just possible only with the agile iterative development.

The organisation and sponsor of the project continuously expect the teams to deliver business value. So, it is important to accept only those requirements which can be delivered at the end of the delivery cycle. This increases the predicable and timely output at the end of every delivery cycle.

Summary of Actions:

- Prioritise the requirements backlog to deliver business value to the customer in agreement with all stakeholders.
- Welcome the changing requirements with clear understanding of the impacts on the internal and external product components.
- Set the delivery expectations due to the impacts from changing requirements and prioritisation in consultation with key stakeholders.
- Service the most business value items with less effort and minimal risk.

Finding 6: Break-down of requirements

The traditional development models produce a single requirement document comprehensively in the beginning requirements phase and there is no break-down of requirements like in agile process. This makes it difficult and confusing for the teams where to start their work, often ignores the dependencies with other components. This process is also time consuming to digest these comprehensive requirements and possible to find gaps in the requirements delaying the whole product readiness and delivery. As discussed in chapter 5, the agile way of breaking the requirements in to epics, features, user stories have several benefits. The primary goal of breaking down the agile requirements is to create to the level of user stories.

Breaking down of requirements does not mean that the requirements are made smaller. It focusses on the bringing the basic functionality required to deliver to the customer and invite their feedback as new features which are then planned for market delivery after prioritisation.

Summary of Actions:

- Breakdown the requirements into small features and user stories to support product basic functionality
- Take advantage of early delivery to the customers and seek their product feedback to plan the next round of enhancements.
- Propose an action plan for incremental delivery for the backlog items and product feedbacks.

5.2 Proposal validation

It was immense pleasure to demonstrate the initial proposal summary and the thesis overview to Mr. Viswanath Tadepalli, Project Manager at Capgemini for feedback and suggestions. Mr. Viswanath had been in the software industry for more than 18 years with extensive experience and understanding of the whole software development life cycle. The proposal validation was done through virtual presentation using Microsoft Teams and further exchange of information via emails and voice calls. The author of this document explained the business challenge, current state findings and the proposed solution to the problem statement.

A snippet of the proposed solution to the CSA findings is shown in the below figure, the highlighted actions are of the key interest for the proposal validator which would be taken for improvements on top his current priority.

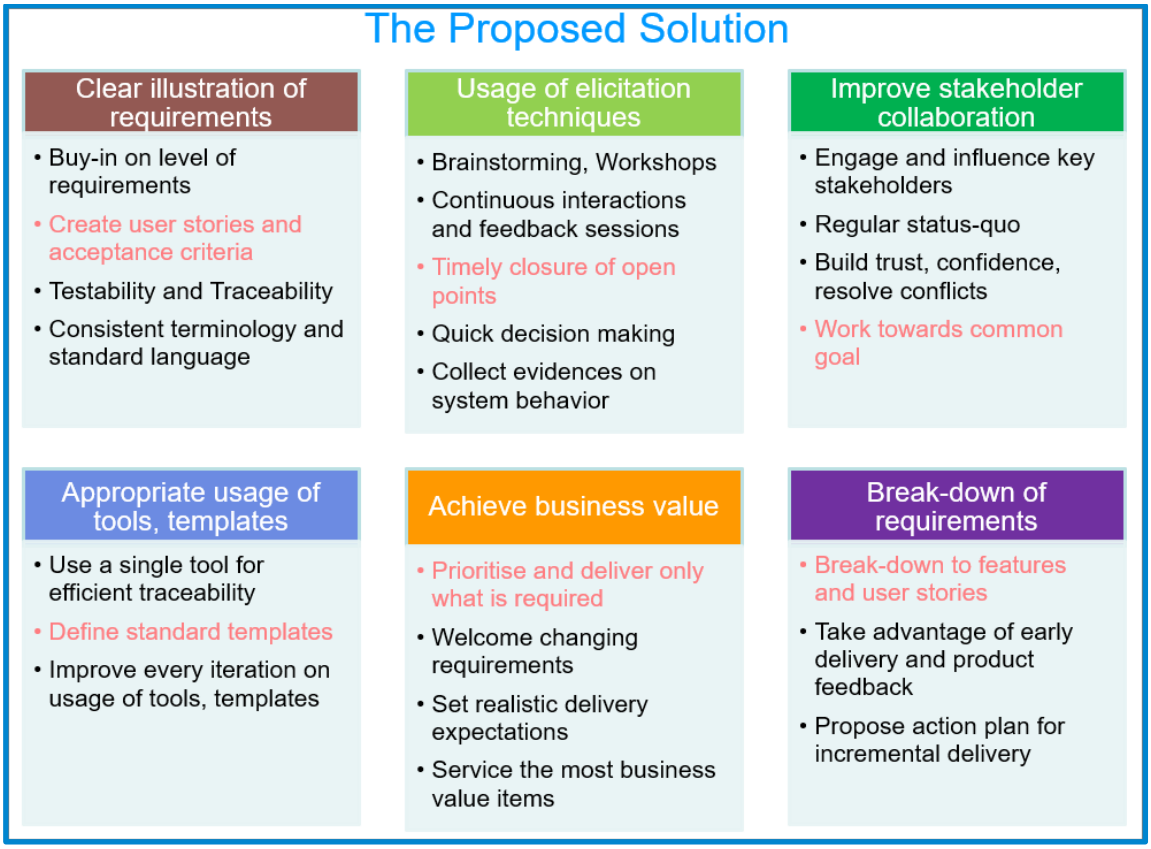


Figure 12: Proposal Validation

Mr. Viswanath felt that the findings and proposed solution covers comprehensive list of actions and are rightly spotted with the current challenges with software product requirements. He suggested to plan for the implementation of these actions iteratively and aim for 2-3 actions at a time in every release. The first step to take this forward is to conduct brainstorming with the key stakeholders and identify the priority list of actions that can be achieved in every release cycle. These priority actions should be mapped to the organisation and project goals to avoid duplication of target goals. Mr. Viswanath was kind to share his experience and thoughts on how the priorities keep changing over time, sometimes no control over the changes and hence the action plan may also undergo some adjustments to the original plan. The goal is to continuously improve the processes and increase the business value aligning with organisation and customer goals. According to Mr. Viswanath, there is a high potential for improvements in this area of the current findings and the proposed list of actions, these are applicable in almost all the projects.

The validation feedback from Mr. Viswanath is shown in the appendix D of this document. The summary of the feedback and next steps to take this forward to the implementation are presented below:

- Proposed actions are comprehensive and innovative, most relevant to the current challenges with requirements and software development.
- Prioritise the list of actions in collaboration with the key stakeholders and create an action plan of what can be achieved in every release cycle.
- Create a plan to implement one key challenge from each finding and define target metrics for each release cycle.
- Implement the most relevant challenges and easy to implement aligning with the organisation goals which are easily achievable and measurable.
- Some of these actions would lead to a new workflow to the existing processes, create entry/exit criteria with regular check points to see that the actions are implemented in the desired manner.
- It is recommended to simplify the processes for maximum benefits and increase the overall productivity.

5.3 Conclusions

The current state analysis helped to identify the key challenges with changing and unclear requirements in the software industry. An attempt was made to analyse the responses from the current state analysis and identify the key challenges which are potential improvements according to most of the respondents. The literature study from various journals, books and articles helped to find the industry best practices and techniques to the CSA key findings. These findings are some of the most recurring problems in the software product development and hence it is recommended to adopt the industry best practices that fits the organisation and teams to achieve the desired goals.

The proposed actions to the problem statement in this document are not limited to agile projects but may also be applied to other traditional projects. The challenges may vary for every organisation. It is recommended to study the internal processes and tools in the organisation before adopting the process improvements to their existing requirement engineering process. Not everything may fit or applicable to every organisation and just by enforcing the processes and techniques may not yield the best results. It is suggested

to simplify the existing processes without affecting the current values and benefits that are already yielding business value. It is recommended to take incremental approach with the proposed solution for the desired results rather than taking all or multiple actions at the same time.

The proposal validation feedback suggests defining achievable and measurable targets for every release aligning with organisational goals. The improvement actions need to be planned in a collaborative way and focus on the immediate values that can deliver quality product within the expected timeline. It is also very common that requirements keep changing and do not have control on how they evolve but one needs to adopt to the changes and mitigate the risks.

It is essential to define a clear requirement engineering process and set the realistic expectations with all the stakeholders in order to achieve the common goal of faster development with maximum business value. In addition, agile practices would enable to keep all the stakeholders intact to effectively manage the changing requirements and prioritisation in every iteration. The lessons from the past iterations should be learning for the future iterations to avoid repetition of the same problems with cascade effect.

For a successful and continuous delivery of software product, elicitation and analysis of the requirements forms the basis for smooth execution of the project in the latter phases of development and testing. This can be achieved with good planning, stakeholder collaboration, product knowledge and appropriate use of tools and techniques. In conclusion, well defined requirements process would be an advantage to the organisations and clients as the value is gained for both sides in terms of delivering a quality product that meets the end user expectations.

References

Young, R.R. (2004). *The Requirements Engineering Handbook.* London: Artech House.

Berenbach, Brian., et al. (2009). *Software & Systems Requirement Engineering: In Practice*. New York: McGraw-Hill.

Chappel, David. (2012). *The Business Value of Agile Development*. California: Chappell & Associates.

Canty, Denise. (2015). *Agile for Project Managers.* Florida: Taylor & Francis Group.

Kaur, Navneet., et al. (2019). Enhancing Quality in Agile Requirement Elicitation Process. *American Journal of Applied Sciences* [Online], 16(1), 1-10. Available from: https://www.researchgate.net/publication/333498133_Enhancing_Quality_in_Agile_Requirement_Elicitation_Process [Accessed 10 May 2020].

Parviainen, Päivi., et al. (2003). Requirements engineering. *Inventory of technologies* [Online], Espoo: VTT Publications 508. Available from: https://www.vttresearch.com/sites/default/files/pdf/publications/2003/P508.pdf [Accessed 10 May 2020].

KPMG. (2019). Agile Transformation. *KPMG Survey on Agility* [Online]. Available from: https://assets.kpmg/content/dam/kpmg/nl/pdf/2019/advisory/agile-transformation.pdf [Accessed 10 May 2020].

Dayel, Mashael., et al. (2014). Reengineering Requirements Specification Based on IEEE 830 Standard and Traceability. *ResearchGate* [Online]. Available from: https://www.researchgate.net/publication/287489669 [Accessed 10 May 2020].

Damian, D.E., et al. (2003). Requirements Engineering challenges in multi-site software development organizations. *Requirements Engineering Journal* [Online], 8, 149-160. Available from: https://www.academia.edu/5095655/Requirements_Engineering_Challenges_in_Multi-site_Software_Development_Organizations [Accessed 10 May 2020].

Swebok. (2015). Software Requirements. *Swebok Board* [Online]. Available from: http://swebokwiki.org/Chapter_1:_Software_Requirements [Accessed 10 May 2020].

FHWA. (2007). Systems Engineering for Intelligent Transport System. *Federal Highway Administration* [Online]. Available from: https://ops.fhwa.dot.gov/publications/seitsguide/ [Accessed 10 May 2020].

Anarsolutions. (2018). Top 5 Requirements Elicitation Techniques. *Anarsolutions* [Online]. Available from: http://www.anarsolutions.com/top-5-requirements-elicitation-techniques/ [Accessed 10 May 2020].

Pipilikasoft. (2015). Requirements Analysis. *Pipilikasoft* [Online]. Available from: https://pipilikasoft.com/requirement-analysis/ [Accessed 10 May 2020].

Javatpoint. (2018). Software Requirements Specifications. *Software Requirements* [Online]. Available from: https://www.javatpoint.com/software-requirement-specifications [Accessed 10 May 2020].

KnowledgeHut. (2019). The Ultimate Guide to the Agile Manifesto. *KnowledgeHut* [Online]. Available from: https://www.knowledgehut.com/blog/agile/agile-manifesto-principles-values [Accessed 10 May 2020].

Microsoft. (2019). Agile Process. *Azure Board Documentation* [Online]. Available from: https://docs.microsoft.com/en-us/azure/devops/boards/work-items/guidance/agile-process?view=azure-devops [Accessed 10 May 2020].

Elgabry, Omar. (2016). Requirements Engineering - Requirements Validation (Part 4). *Medium* [Online]. Available from: https://medium.com/omarelgabrys-blog/requirements-engineering-requirements-validation-part-6-29778d7bde24 [Accessed 10 May 2020].

Agile Alliance. (2020). 12 Principles behind the Agile Manifesto. *Agile Manifesto* [Online]. Available from: https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/ [Accessed 18 May 2020].

Kruger, Nico. (2018). How to Write A PRD – With Examples. *Perforce* [Online]. Available from: https://www.perforce.com/blog/alm/how-write-product-requirements-document-prd [Accessed 18 May 2020].

Appendix A: Abbreviations

CSA	Current State Analysis
SPR	Software Product Requirements
RE	Requirements Engineering
SDLC	Software Development Life Cycle
IEEE	Institute of Electrical and Electronics Engineers
SRS	Software Requirements Specification
IT	Information Technology
PRD	Product Requirement Document

Appendix B: Survey Questionnaire

1. Q1: Your primary areas of work or experience? *
   ☐ Banking & Finance ☐Healthcare ☐eCommerce ☐Insurance ☐Mobile Applications ☐ Other:

2. Q2: Choose your experience level in the software/product industry *
   ◯ > 20 years ◯ < 20 years ◯ < 15 years
   ◯ < 10 years ◯ < 5 years

3. Q3: Do you think that the requirements process can be same for all type of software products irrespective of its size and complexity? *
   ◯ Yes ◯ No ◯ May be ◯ Other:

4. Q4: You had worked or currently working with *
   ☐Sales and Marketing ☐Requirements ☐Development ☐Testing ☐Infrastructure ☐Project Management ☐Other:

5. Q5: Software development model followed in your organization/project? *
   ◯ Waterfall ◯ V Model ◯ Agile ◯ DevOps ◯ Other:

6. Q6: Do you think that the end user needs are translated into quality requirements based on your experience with delivery of software product? *
   ◯ Yes ◯ No ◯ Maybe ◯ Other:

7. Q7: Do you agree that the requirements must be clearly documented with examples, illustrations and user stories etc.? *

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

8. Q8: What are the best ways of conducting requirements elicitation and analysis phase according to you? *
   ☐Workshops ☐Online meetings ☐Brainstorming ☐Observations ☐Experiments ☐Other:

9. Q9: What do you think, the collaboration between the requirements, development & testing teams be during the whole product life cycle? *
   ☐Continuous walk-through sessions with Dev & Testing teams
   ☐Cross-check with development on feasibility
   ☐Feedback from Testers on current behaviour & its impact
   ☐Requirements signed-off independently
   ☐Requirements support required until the finished product
   ☐Other:

10. Q10: Are you satisfied with your requirements team knowledge and competence in defining quality product requirements? *

	1	2	3	4	5	
Less Satisfied	◯	◯	◯	◯	◯	Extremely Satisfied

11. Q11: Do the requirements team make efforts to understand the end user needs and estimate the product feasibility and design constraints well in advance? *

Yes No Maybe Other:

12. Q12: Is your requirements team able to communicate the end user expectations to the development and testing teams clearly? *

Yes No Maybe Other:

13. Q13: Have you used any of the tools for maintaining product requirements? *

Yes No Other:

14. Q14: What are the benefits and values in using tools for requirements? *☐Requirements maintenance ☐Traceability and Coverage ☐Efficient Reviews ☐Version control ☐eSign ☐Other:

15. Q15: What are your current challenges with respective to your requirements analysis phase which impacts both development and testing? *

☐Misunderstanding or Unclear requirements ☐Integration challenges ☐Requirements volatility ☐No Acceptance criteria ☐Lack of knowledge sharing ☐Conflict requirements ☐Other:

16. Q16: Do you think the requirements need to be reviewed by all the stakeholders before they are signed-off? Stakeholders include architects, developers and testers. *

	1	2	3	4	5	
Strongly disagree						Strongly Agree

17. Q17: Do you have a standard template for capturing product requirements? *

Yes No Other:

18. Q18: Would it be helpful to classify the requirements as mandatory, conditional, future, nice to have etc. so that they can be prioritized for delivery planning? *

	1	2	3	4	5	
Less Helpful						Extremely Helpful

19. Q19: How often the requirements change even after the design phase and during final testing phases? *

	1	2	3	4	5	
Very Often						Less Often

20. Q20: Do you think that the developers and testers should be given an opportunity to participate in the requirements discussions and interact with customers/end users? *

Occasionally Conflict resolution Every iteration
At regular intervals Never Other:

21. Q21: What is your opinion on the project plan, does it give enough importance to elicitation and analysis of requirements? *

	1	2	3	4	5	
Less Important						Extremely important

22. Q22: Do you agree to this statement "Poor requirements always leads to rework and budget overruns"? *

	1	2	3	4	5	
Strongly Disagree	◯	◯	◯	◯	◯	Strongly Agree

23. Q23: Do you think that breaking down to small/medium requirements helps to deliver a quality product on time? *

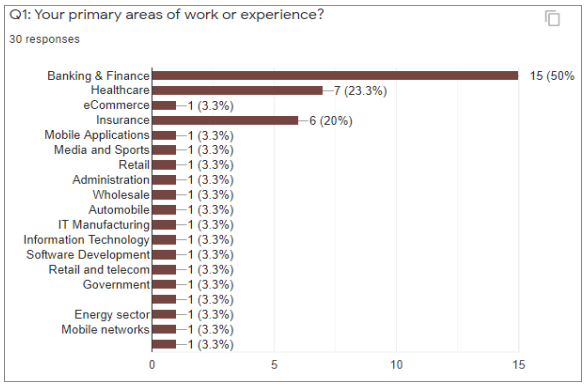
◯ Yes ◯ No ◯ Maybe ◯ Other:

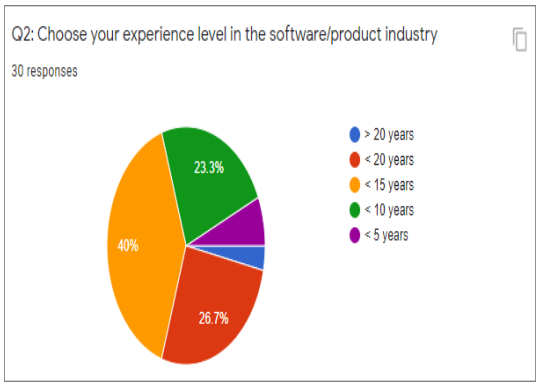
24. Q24: Do you think it's important to measure quality characteristics for the product requirements? *

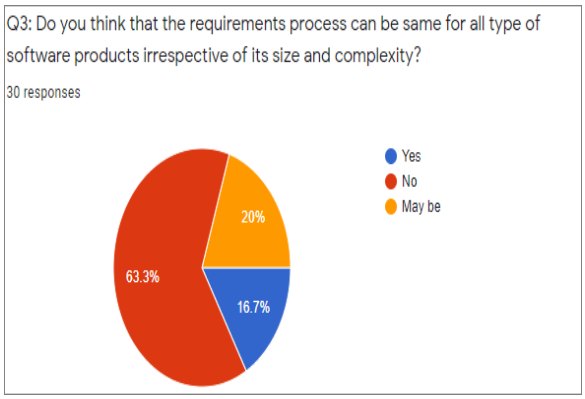
	1	2	3	4	5	
Less Important	◯	◯	◯	◯	◯	Extremely Important

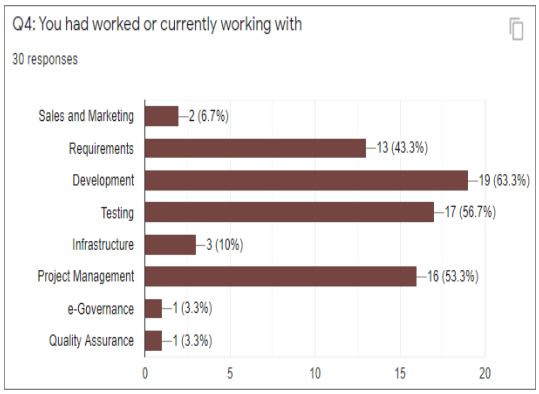
25. Q25: Do you have some suggestions/improvements to make the requirements process more efficient and simpler? *

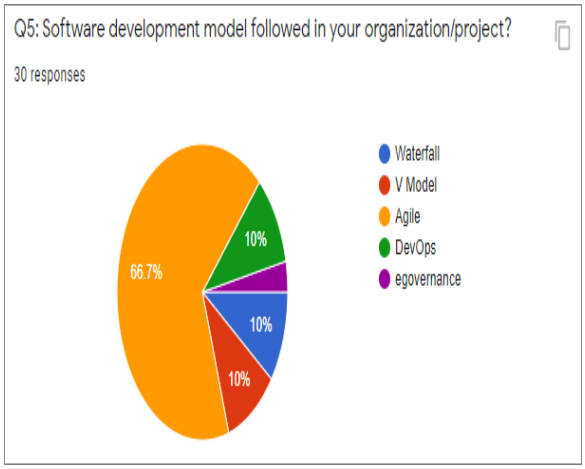
Appendix C: Survey Responses

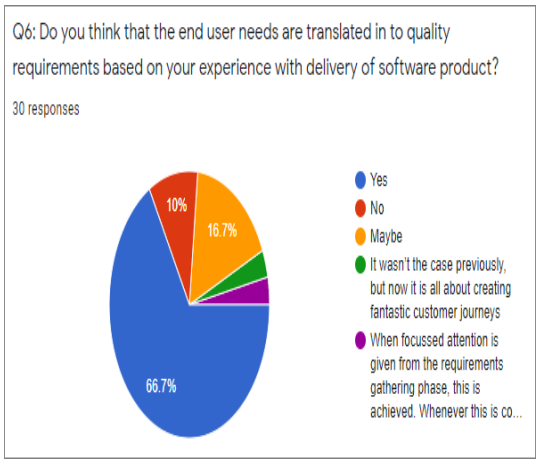


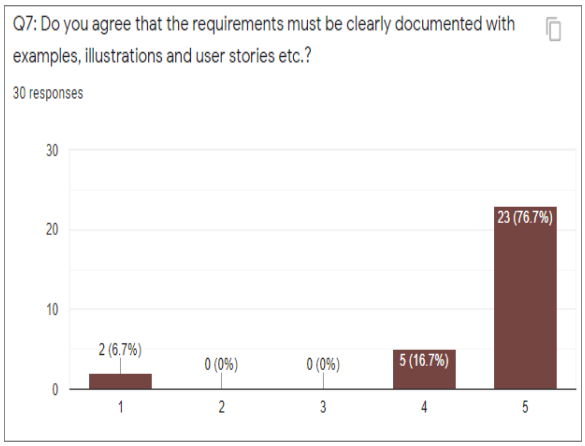


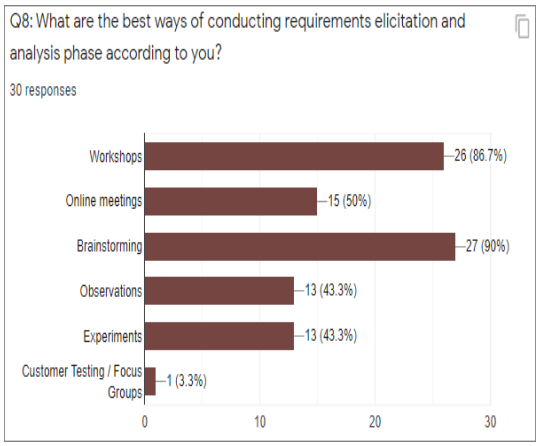






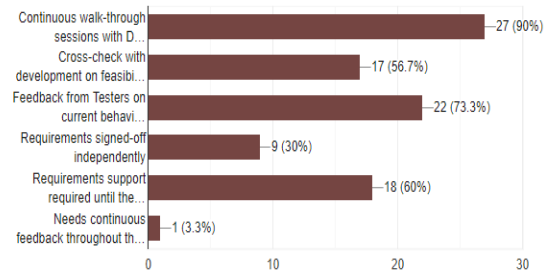






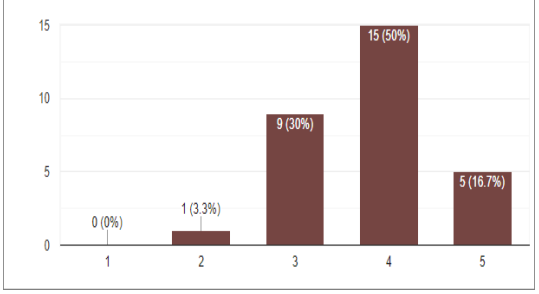
Q9: What do you think, the collaboration between the requirements, development & testing teams be during the whole product life cycle?

30 responses



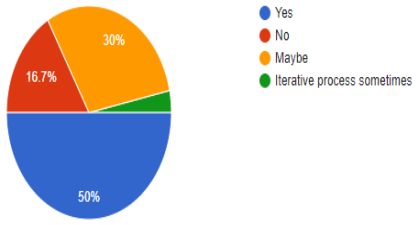
Q10: Are you satisfied with your requirements team knowledge and competence in defining quality product requirements?

30 responses



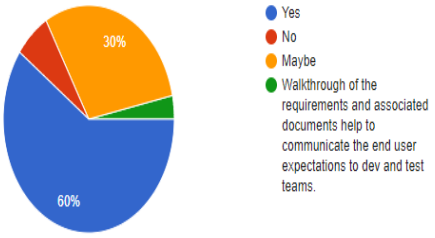
Q11: Do the requirements team make efforts to understand the end user needs and estimate the product feasibility and design constraints well in advance?

30 responses



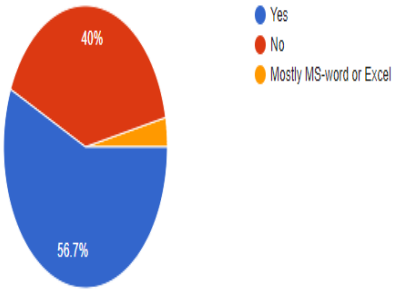
Q12: Is your requirements team able to communicate the end user expectations to the development and testing teams clearly?

30 responses



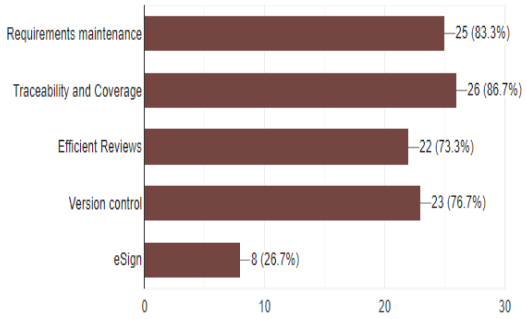
Q13: Have you used any of the tools for maintaining product requirements?

30 responses



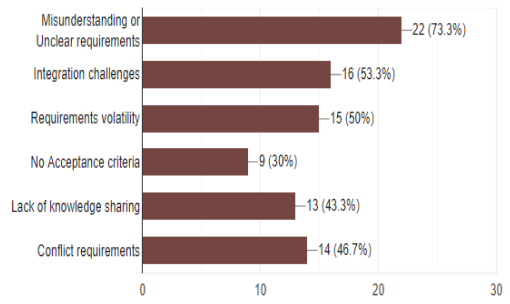
Q14: What are the benefits and values in using tools for requirements?

30 responses



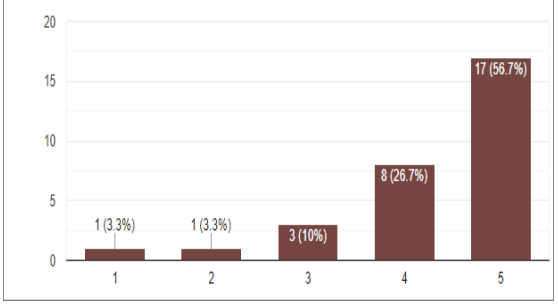
Q15: What are your current challenges with respective to your requirements analysis phase which impacts both development and testing?

30 responses



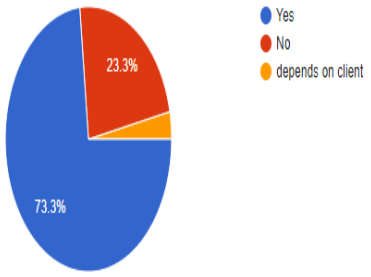
Q16: Do you think the requirements need to be reviewed by all the stakeholders before they are signed-off? Stakeholders include architects, developers and testers.

30 responses



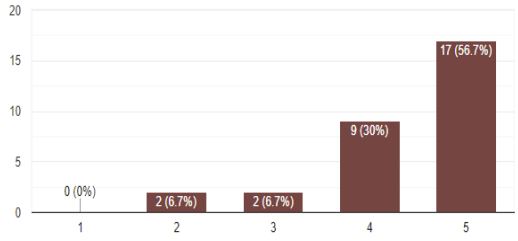
Q17: Do you have a standard template for capturing product requirements?

30 responses



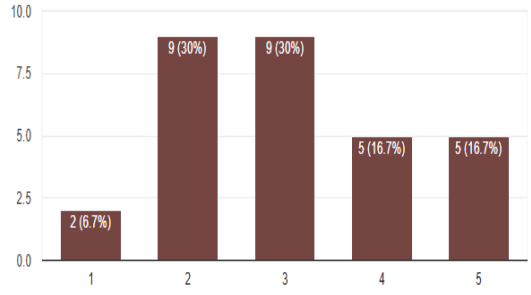
Q18: Would it be helpful to classify the requirements as mandatory, conditional, future, nice to have etc. so that they can be prioritized for delivery planning?

30 responses



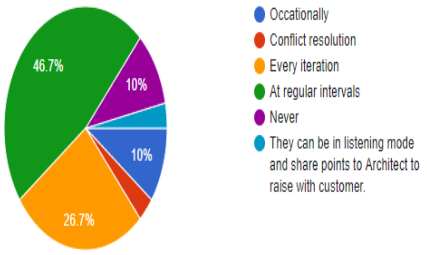
Q19: How often the requirements change even after the design phase and during final testing phases?

30 responses



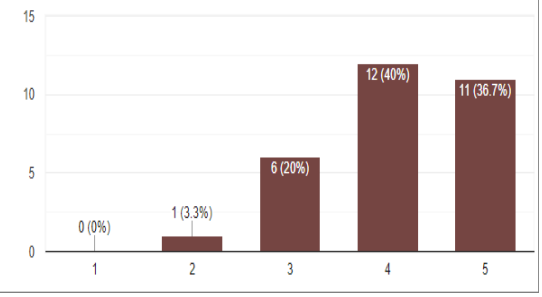
Q20: Do you think that the developers and testers should be given an opportunity to participate in the requirements discussions and interact with customers/end users?

30 responses



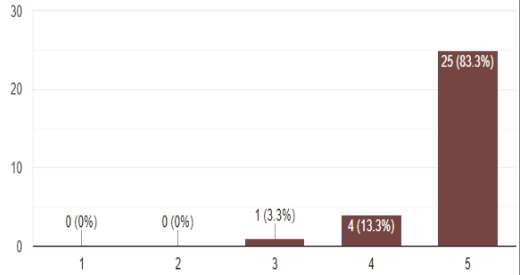
Q21: What is your opinion on the project plan, does it give enough importance to elicitation and analysis of requirements?

30 responses



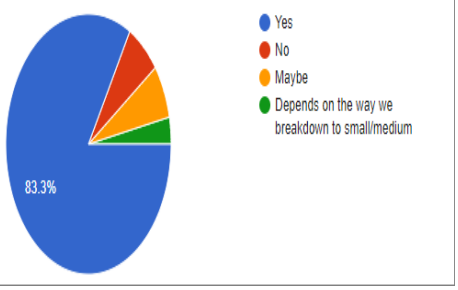
Q22: Do you agree to this statement "Poor requirements always leads to rework and budget overruns"?

30 responses



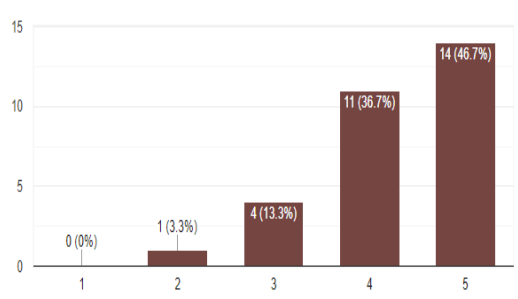
Q23: Do you think that breaking down to small/medium requirements helps to deliver a quality product on time?

30 responses



Q24: Do you think its important to measure quality characteristics for the product requirements?

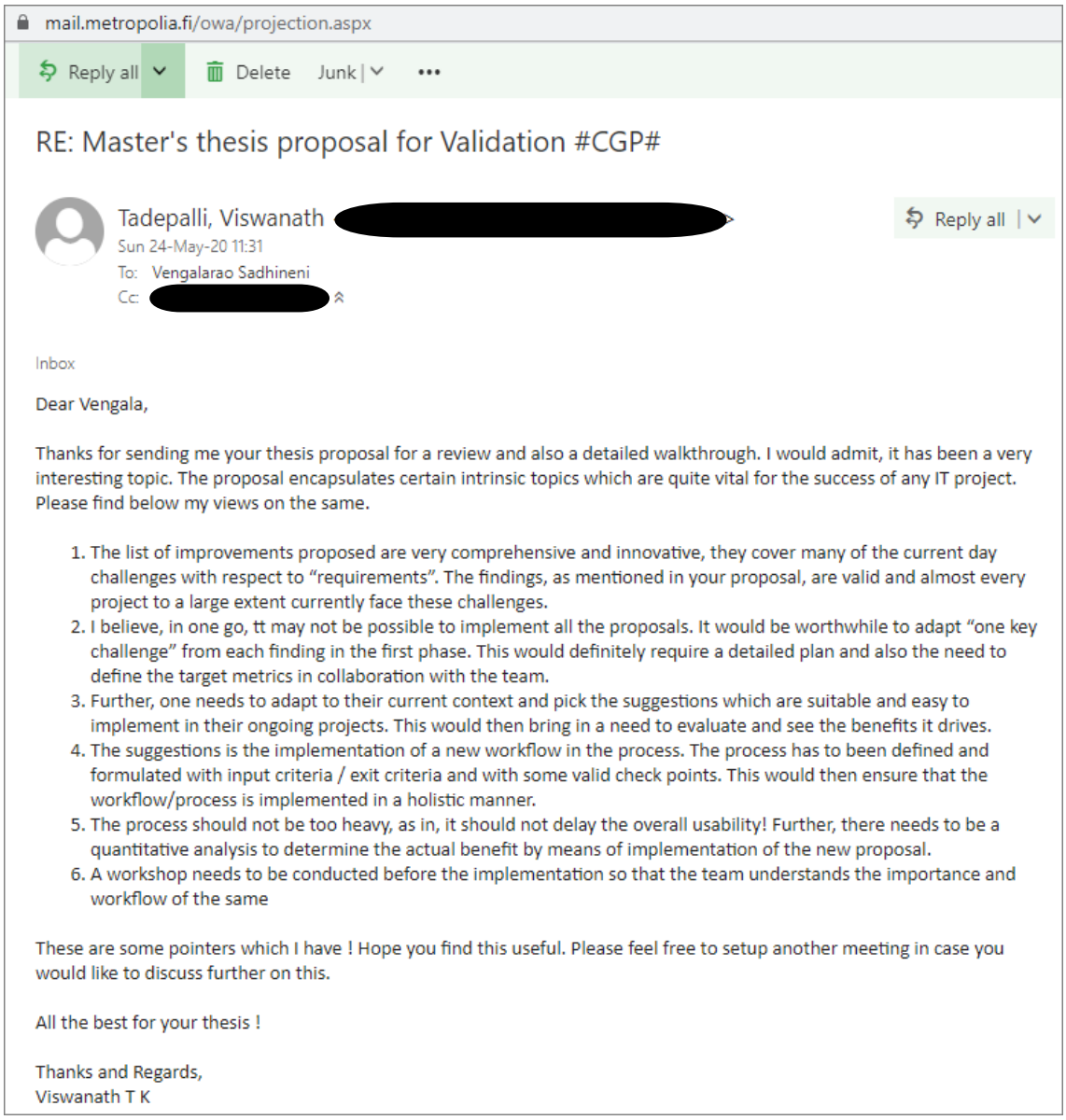
30 responses



Q25: Do you have some suggestions/improvements to make the requirements process more efficient and simpler? 30 responses

Resp.#	Respondent Comments
1	No
2	NA
3	Clarity of thoughts, sign-off, do not allow scope creep, effective planning and prioritization
4	Knowledge of the requirements team about the market and software are key factors that determine the quality of the requirements. Team must be able to think out of box as well as in different perspective to tackle and understand the requirements.
5	Think-tank, developers and testers must know the understanding of product process and the important parameters and procedure to test.
6	Clearly defining the process and proper training
7	Make sure everyone is on the same page of what they are building
8	Take help of a good tool or process or template to illustrate requirements.
9	Not many as of now.
10	Don't just document what BA hears from business. He needs to have application knowledge, perform impact analysis and then write requirements. Documentation is the key and open discussion with all stakeholders
11	Good
12	Not much
13	Dev and QA teams should be given proper knowledge sharing sessions to understand the requirements
14	The process of capturing requirements for 3 months at the Initial stages of the project, signing them off and passing it off to design, development and testing with the finished product arriving many months later may no longer be relevant to customer needs. So, the process of capturing all the requirements at the start of the project has gone. One must plan and develop products iteratively giving the flexibility to adapt to client needs quickly.
15	NA
16	Make sure need to follow standard guidelines to requirement process
17	No
18	Breakdown of requirements to meet end user needs.
19	Breakdown requirements as in Agile methodology
20	providing a domain-based training will helps better for developers.
21	No
22	No
23	More focus on clearly specifying the requirements in a document. Representation and specifications should be clear.
24	Following some standards best practices that are proven which leads to a better product delivery might be a good bet.
25	Based on the domain, during requirements phase it would be always better to have a questionnaire of relevant areas or topics to be covered and always cross check for each functionality that's being discussed. Always better to have sample illustration or example to be notes and confirmed by the Client. It would be better to specially note down areas of impact or areas of dependency of requirements.
26	1. Participants in the Requirements collection process should be collected from all relevant departments. They should be fully trained and able to prioritize business needs based on the product situation 2. Requirements collection and prioritization should be based on time and scope 3. Requirements should be well defined with examples, all business rules and data. Requirements should be testable 4. There should be a process or tool to trace all the requirements 5. Requirements should be walked through and signed off by all relevant stakeholders
27	Qualified consultants and enough time
28	None
29	Requirement gathering to be done in accordance with developers, testers and other business users.
30	Communication should be clear and transparent between BIZ users and BSA's and Dev Team

Appendix D: Validation Feedback



mail.metropolia.fi/owa/projection.aspx

Reply all | Delete | Junk | •••

RE: Master's thesis proposal for Validation #CGP#

Tadepalli, Viswanath [redacted]

Sun 24-May-20 11:31

To: Vengalarao Sadhineni

Cc: [redacted]

Reply all

Inbox

Dear Vengala,

Thanks for sending me your thesis proposal for a review and also a detailed walkthrough. I would admit, it has been a very interesting topic. The proposal encapsulates certain intrinsic topics which are quite vital for the success of any IT project. Please find below my views on the same.

1. The list of improvements proposed are very comprehensive and innovative, they cover many of the current day challenges with respect to "requirements". The findings, as mentioned in your proposal, are valid and almost every project to a large extent currently face these challenges.
2. I believe, in one go, tt may not be possible to implement all the proposals. It would be worthwhile to adapt "one key challenge" from each finding in the first phase. This would definitely require a detailed plan and also the need to define the target metrics in collaboration with the team.
3. Further, one needs to adapt to their current context and pick the suggestions which are suitable and easy to implement in their ongoing projects. This would then bring in a need to evaluate and see the benefits it drives.
4. The suggestions is the implementation of a new workflow in the process. The process has to been defined and formulated with input criteria / exit criteria and with some valid check points. This would then ensure that the workflow/process is implemented in a holistic manner.
5. The process should not be too heavy, as in, it should not delay the overall usability! Further, there needs to be a quantitative analysis to determine the actual benefit by means of implementation of the new proposal.
6. A workshop needs to be conducted before the implementation so that the team understands the importance and workflow of the same

These are some pointers which I have ! Hope you find this useful. Please feel free to setup another meeting in case you would like to discuss further on this.

All the best for your thesis !

Thanks and Regards,
Viswanath T K