Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Guide for Elicitation Techniques applied to Agile Software Development

Rodrigo Pereira de Mesquita

Master's Degree Dissertation

Orientadora
Profa. Dra. Edna Dias Canedo

Brasília
2023

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Guide for Elicitation Techniques applied to Agile Software Development

Rodrigo Pereira de Mesquita

Master's Degree Dissertation

Profa. Dra. Edna Dias Canedo (Orientadora)
CIC/UnB

Prof. Dr. Vander Alves
CIC/UnB

Profa. Dra. Sabrina dos Santos Marczak
Escola Politécnica – PUCRS

Prof. Dr. Ricardo Pezzuol Jacobi
Coordenador do Programa de Pós-graduação em Informática

Brasília, April 27, 2023

Dedication

I dedicate this work to my wife Camila and my daughters Júlia and Letícia, who are always supporting and cheering for me. Nothing would be possible without my family. Thank you for being my inspiration.

Acknowledgments

Thanks for my advisor, Dra. Edna Dias Canedo, to whom I had the privilege of being supervised by and guided through the obstacles and decisions to be made for this research.

I also want to express my gratitude to all colleagues who took the time to answer the survey and attended to the evaluation sessions. All participants made possible to have the necessary data to analyze the practices, methods and tools being used or studied by the community.

Resumo

Contexto: As técnicas de elicitação de requisitos são essenciais para apoiar os engenheiros de requisitos a obter uma melhor compreensão das necessidades dos usuários e das partes interessadas. Embora existam diversas técnicas disponíveis para apoiar as fases da Engenharia de Requisitos (ER), podem ocorrer dúvidas nas equipes de desenvolvimento de software sobre qual técnica utilizar durante a atividade de elicitação de requisitos. **Objetivo:** O objetivo desse trabalho é identificar as técnicas de elicitação de requisitos mais utilizadas na literatura e comparar com as técnicas mais usadas pelos profissionais na indústria. Além disso, identificam-se os desafios relacionados à elicitação de requisitos, os prós e contras das principais técnicas identificadas na literatura, e baseado nos prós e contras, analisam-se combinações possíveis das técnicas de elicitação de requisitos que podem minimizar os desafios identificados na literatura e na indústria. **Método:** Realizou-se uma Revisão Sistemática da Literatura (SLR) para identificar as técnicas de elicitação de requisitos e os desafios discutidos na literatura ou na indústria. Ainda dentro da Revisão Sistemática da Literatura, foi realizada a técnica de snowballing, para revisão dos estudos primários encontrados dentro de outras SLR que foram trazidas pela string de busca. Além disso, realizou-se uma pesquisa de opinião (*survey*) para investigar a percepção dos praticantes da área de desenvolvimento de software (indivíduos trabalhando na indústria de Software, independentemente da posição ou papel desempenhado) em relação às técnicas identificadas e posteriormente compará-las com os resultados obtidos na SLR. Finalmente, usando a técnica Grupo Focal, realizou-se duas sessões de avaliação com dezenove especialistas para avaliar as combinações de técnicas e os resultados fornecidos neste estudo. **Resultados:** Foram identificados 54 estudos primários na SLR e eles demonstraram que as técnicas tradicionais ainda são as mais utilizadas tanto na literatura quanto nos projetos da indústria de software ao mesmo tempo que técnicas como Persona estão ganhando espaço. Além disso, investigou-se combinações de técnicas já discutidas e apresentadas na literatura, também com base nos pontos fortes encontrados na literatura para cada técnica, foi possível identificar combinações de técnicas que pelas forças ou pontos a favor identificados no SLR, poderiam ser combinados para superar a maioria dos desafios identificados. As sessões de validação proporcionaram coletar a visão dos especialistas que complementaram as técnicas e combinações em uso pela comunidade. **Conclusão** As técnicas mais mencionadas pela literatura e usadas na indústria de software foram: Prototipação, Entrevistas, Estórias de Usuário, Cenários e Etnografia. A combinação do uso dessas técnicas com, por exemplo, a técnica Persona podem ajudar a superar os desafios identificados na literatura. O estudo conta com a descrição das principais técnicas identificadas na literatura, contendo seus prós e contras e poderá apoiar os engenheiros de requisitos durante a elicitação de requisitos. A disponibilização deste estudo para apoiar os profissionais de software na elicitação de requisitos, permitirá que a comunidade de engenharia de software contribua com *feedback* relacionado à combinação do uso de técnicas, permitindo assim um aprimoramento e disseminação das percepções das combinações feitas entre as técnicas de ER por os profissionais das equipes de desenvolvimento de software. Portanto, o guia pode auxiliar os profissionais de software na escolha das técnicas a serem usadas e (ou) combinadas.

Palavras-chave: Desenvolvimento de Software Ágil, Engenharia de requisitos, Elicitação de Requisitos, Técnicas de elicitação, Prós e contras das Técnicas de Elicitação, Desafios do Desenvolvimento de Software Ágil, Combinações de Técnicas.

Resumo Expandido

Título: Guia para Elicitação de Requisitos aplicado ao Desenvolvimento de Software Ágil

Diversos estudos apresentam a fase de Elicitação de Requisitos como uma das mais difícies de realizar e, certamente, como uma das mais importantes para o bom desenvolvimento de software. Não é diferente com o modelo de desenvolvimento de Software Ágil que depende da realização da elicitação de forma robusta e eficiente para entregar o melhor resultado possível.

Para tanto, as técnicas de elicitação de requisitos são essenciais para ajudar os engenheiros de requisitos a obter uma melhor compreensão das necessidades dos usuários e partes interessadas. Atualmente, existem muitas técnicas de Engenharia de Requisitos (ER) para apoiar a atividade de elicitação de requisitos, algumas delas, mais utlizadas há mais tempo pela comunidade e, naturalmente, mais conhecidas, enquanto outras, desenvolvidas mais recentemente, ainda não atingem uma parcela maior de engenheiros de requisitos, desenvolvedores e demais praticantes ligados ao desenvolvimento de software (individuos trabalhando na indústria de Software, independentemente da posição ou papel desempenhado).

A vasta gama de técnicas existentes veio para auxiliar a todos os envolvidos, porém, por outro lado, a simples existência de uma técnica de elicitação, por si só, não garante o sucesso da etapa e, por consequência, do projeto. A falta de experiência teórica e prática em técnicas, tradicionais ou mais recentes, bem como falta de conhecimento sobre potencialidades ou dificuldades da técnicas, ou em outras palavras, as forças e fraquezas, pode levar a equipe de desenvolvimento e aos responsáveis pela análise de requisitos a não fazer as melhores escolhas possíveis ao projeto. Na prática, cada projeto tem seu ambiente organizacional, sua cultura, seu cliente, e suas partes interessadas específicas, que requerem que diferentes táticas e técnicas sejam empregadas de forma a conseguir o melhor resultado.

Baseado nessa análise de discussões anteriores, essa pesquisa trouxe três objetivos principais, de acordo com a revisão de literatura realizada.

- i) identificar as técnicas mais utilizadas na literatura e comparar com as técnicas mais usadas na indústria;
- ii) levantar os desafios listados na literatura para a elicitação de requisitos, bem como, os prós e contras das principais técnicas identificadas no item anterior;
- iii) baseado nos prós e contras identificados, analisar possíveis combinações de técnicas que ajudem a superar os desafios apresentados na literatura.

Para investigar os objetivos mencionados, as seguintes questões de pesquisa (QPs) foram definidas:

- QP.1 Quais são as técnicas, existentes na literatura para elicitação de requisitos que podem ser utilizadas nos processos de desenvolvimento ágil?
- QP.2 Quais são as forças e fraquezas (prós e contras) das técnicas identificadas bem como os desafios da elicitação de requisitos que estão reportados na literatura?
- QP.3 Quais técnicas poderiam ser combinadas para melhorar o processo de elicitação de requisitos?

Para responder às questões de pesquisa (QP) realizou-se uma Revisão Sistemática da Literatura (RSL) buscando identificar as técnicas de elicitação de requisitos mais discutidas na literatura, seus pontos fortes e pontos fracos, bem como comparou-se com a lista de técnicas mais utilizadas na indústria. Ademais, como alguns dos estudos trazidos pela *string* de busca eram, por sua vez, também Revisões Sistemáticas de Literatura, foi utilizada a técnica de "bola de neve" analisando os estudos primários constantes nesses estudos, para manter a compatibilidade das análises com foco nos estudos primários apenas. Ainda em relação às técnicas, realizou-se uma pesquisa de opinião com profissionais de Tecnologia da Informação para confrontar com os resultados obtidos na RSL, buscando analisar quais técnicas são mais conhecidas e utilizadas bem como quais combinações de técnicas são familiares aos praticantes da área.

Para a segunda pergunta de pesquisa, analisou-se, dentro dos papéis selecionados, estudos relacionados ao desenvolvimento ágil de software que mencionaram desafios diversos na utilização do modelo ágil de desenvolvimento. Após a análise dos desafios, esses foram categorizados em grandes grupos para facilitar as análises.

Durante a condução da Revisão Sistemática da Literatura (RSL) foram utilizados bancos de dados indexadores para a busca automática, a partir de uma *string* de busca e também, foram realizadas pesquisas manuais nos periódicos e conferências mais relevantes da área. Inicialmente, foram extraídos 496 estudos e, após a remoção dos estudos duplicados, bem como, a realização das análises baseadas nos critérios definidos de exclusão e qualidade, foram selecionados 54 estudos para contribuir com esse trabalho.

A partir dos 54 estudos selecionados, a RSL demonstrou que as técnicas tradicionais ainda são as mais utilizadas tanto na literatura quanto nos projetos da indústria de software. As análises foram ratificadas pela pesquisa de opinião realizada com participantes (desenvolvedores, engenheiros, analistas e gerentes de projeto) ligados ao Desenvolvimento de Software Ágil. Além de identificar as técnicas mais citadas e discutidas, foi possível identificar aquelas que ainda não apresentam o mesmo volume de estudos na literatura, porém já estão sendo bem utilizadas na indústria.

Foi identificado que Prototipação, Entrevistas, Estórias de Usuários, Brainstorming, Observação, Cenários, Questionários e Mapas Mentais são as técnicas mais citadas na literatura. Ademais, Etnografia, Joint Application Development (JAD) and Workshop também são bastante referenciadas na literatura, porém não sendo realmente utilizadas em projetos reais da indústria. As três primeiras técnicas (Entrevistas, Prototipação e Estórias de Usuários) também são as mais encontradas na indústria, por outro lado, Persona, pelo menos com base nos estudos analisados, não é ainda tão discutida na literatura enquanto mostra-se cada vez mais utilizada na indústria, trazendo formas diferentes de realizar o levantamento de requisitos junto a usuários e partes interessadas.

A técnica de Casos de Uso foi amplamente referenciada, porém essa é uma técnica de especificação, de forma que não foi considerada para análises. *Design Thinking* também foi identificado nos estudos selecionados. Isso foi levado em consideração para fins da pesquisa de opinião, porém não elencado para fins de combinações pelo fato de *Design Thinking* representar um processo que utiliza técnicas de elicitação, dentre outras.

Sobre os desafios, pergunta de pesquisa 2, foi organizada uma lista, novamente, baseada nos estudos selecionados e, a partir dela, foi realizada a categorização dos principais desafios em Desenvolvimento de Software Ágil, sendo os principais: tais como: documentação falha, negociação e priorização de requisitos, arquitetura e qualidade de software, comunicação incompleta ou falha, falta de habilidade do time ou dos usuários, escopo e volatilidade dos requisitos, complexidade e escalabilidade dos projetos, conhecimento de domínio por parte dos usuários, cultura e ambiente organizacional, disponibilidade e engajamento das partes interessadas, tarefas relacionadas puramente à elicitação de requisitos e dificuldades em traduzir os requisitos, além de outras citações que foram definidas como "outras", por não se incluírem em nenhuma das anteriores. Alguns dos desafios apresentados estavam relacionados a outras etapas do ciclo e não foram levados adiante na análise, e algumas categorias como "Cultura e Ambiente" e "Arquitetura e Qualidade do Software" foram consideradas não específicas para Elicitação de Requisitos. As out-

ras categorias: “Documentação”, “Comunicação”, “Tradução dos Requisitos”, “Escopo e Volatilidade dos Requisitos”, “Disponibilidade e Engajamento das Partes Interessadas”, “Complexidade e Escalibilidade dos Projetos”, “Negociação e Priorização de Requisitos”, “Expertise do time e dos usuários”, “Tarefas de Elicitação de Requisitos”, foram entendidas como relevantes ao estudo.

Por fim, após análise das características das principais técnicas sobre seus pontos fortes e fraquezas, o estudo buscou apresentar algumas combinações de técnicas que poderiam ser úteis para ajudar a superar esses mesmos desafios elencados pela pesquisa. Algumas combinações de técnicas foram encontradas na literatura como, por exemplo, “Estórias de Usuários e Entrevistas”, “Estórias de Usuários e Prototipação”, “Estórias de Usuários e Brainstorming”, “Entrevistas e Prototipação”, dentre outras.

Também foram realizadas duas sessões de validação usando a técnica de Grupo Focal, que, em resumo, corresponde à reunião de um grupo de especialistas convidados para discutir itens específicos. A técnica de grupo focal prega que perguntas abertas, para motivar a discussão, sejam realizadas. Nessas sessões, dezenove (19) especialistas debateram o uso das técnicas, de combinações de técnicas e avaliaram como as informações apresentadas nesse estudo poderiam ajudar aos praticantes e quais melhorias poderiam ser realizadas no Mapa de Técnicas.

Das sessões de Grupo Focal, foi identificada que a técnica de Modelagem de Processos de Negócio, uma técnica não específica para elicitação, é muito utilizada para levantamento de requisitos, sendo uma das mais lembradas pelos especialistas de ambas as sessões, inclusive, sendo utilizada em combinação com Prototipação, Brainstorming, Entrevistas e Questionários.

Esse estudo, portanto, apresenta uma lista contendo as principais técnicas identificadas na RSL, com seus respectivos pontos fortes e fracos, bem como, possíveis combinações de uso das técnicas, tanto encontradas na literatura, como sugeridas nos grupos focais e pesquisas realizadas, ou ainda, apresentadas e sugestionadas pelo autor,baseado nos pontos fortes e fracos encontrados para cada técnica analisada.

O objetivo desse trabalho foi apresentar o mapa mais completo possível com informações sobre ciclo de desenvolvimento de projetos, técnicas para elicitação de requisitos, desafios e categorias de desafios que poderão ser encontrados em casos práticos, análise de vantagens e desvantagens de técnicas de elicitação de requisitos, e ainda, baseado nas análises, técnicas e combinações de técnicas que poderiam auxiliar os profissionais de Tecnologia da Informação e Comunicação (TIC) a superar as categorias de desafios identificados durante esse trabalho.

Palavras-chave: Desenvolvimento de Software Ágil, Engenharia de requisitos, Elicitação de Requisitos, Técnicas de elicitação, Prós e contras das Técnicas de Elicitação, Desafios do Desenvolvimento de Software Ágil, Combinações de Técnicas.

Abstract

Background: Requirements elicitation techniques are essential to support requirements engineers to gain a better understanding of the needs of users and stakeholders. Although there are several techniques available to support the Requirements Engineering (RE) software development teams might be doubtful about which technique to use during requirements elicitation. **Objective:** The goal of this work is to identify Requirements Elicitation (RE) Techniques most used in the literature and compare with the techniques most used by professionals in the industry. In addition, we identified the challenges related to requirements elicitation, the pros and cons of the main techniques identified in the literature, and based on the pros and cons, analyze possible combinations of requirements elicitation techniques that can minimize the challenges identified in literature and industry. **Method:** We performed a Systematic Literature Review (SLR) to identify requirements elicitation techniques and challenges discussed in the literature or industry. Moreover, we performed a Survey to investigate the perception of software practitioners (individuals working in the software industry in a large variety of roles and positions) in relation to the techniques identified and subsequently compare them with the results obtained in the SLR. Finally, using Focus Group technique, we executed two validation sessions with nineteen specialists to evaluate technique combinations and the findings provided on this guide. **Results:** 54 primary studies were identified in the SLR and they demonstrated that traditional techniques are still the most used in both literature and software industry projects. In addition, some techniques, such as Persona, are gaining ground, helping requirements engineers to find different ways to elicit requirements from end users and stakeholders. Moreover, we have investigated combinations of techniques already discussed and presented in literature, also based on the strengths found in the literature for each technique, it was possible to identify combinations of techniques that by the forces or points in favor identified in the SLR, could be combined to overcome most of the challenges identified. Furthermore, validation sessions provided the view of specialists that complemented the techniques and combinations under use by the community. **Conclusion** The most mentioned techniques in the literature at the same time they are used in the software industry are: Prototyping, Interview, User Stories, Brainstorming, Observation, Scenarios, Questionnaires and Mind Mapping. In addition, Ethnography, Joint Application Development (JAD) and Workshop have many references in literature while are not appealing for real projects in industry. On the other hand, Persona at least from the papers retrieved during this search is not largely discussed in literature whereas it has shown to be widely used in the industry. Combine the use of RE techniques can help overcome the challenges identified in the literature. A guide with a description of all techniques identified in the literature, containing their advantages and disadvantages can support the requirements engineers during the requirements elicitation. The provision of this study to support software practitioners in eliciting requirements, will allow the software engineering community to contribute feedback related to the combination of the use of techniques, thus allowing an improvement and dissemination of the perceptions of the combinations made between the RE techniques by the professionals of the software development teams. Hence, the guide can support software practitioners in choosing the techniques to be used and (or) combined.

Keywords: Agile Software Development, Requirements Engineering, Requirements Elicitation, Elicitation Techniques, Agile Software Development (ASD) Challenges, Pros and Cons of Elicitation Techniques, Combination of Techniques.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Since the beginning of times, knowing what the other party wants is the essential very first step to succeed and accomplish anything. However despite decades of extensive studies by scholars, thousand of projects developed by companies and a large variety of techniques created, industry and academia still struggle to identify the correct requirements within a project. It is a fact that some software teams are not proficient at eliciting requirements from customers and other sources [6]. Although seems easy, the ugly truth is even the definition of what the term requirements mean is the object of a large discussion among software practitioners. We use the term software practitioners to refer to individuals working in the software industry in a large variety of roles and positions. This includes employees who analyze, design, develop, implement, operate, maintain, or manage software (for instance, software engineers, software testers, product owners, software consultants, and data scientists).

Returning to the requirements discussion, we first present some definitions for requirements. Ryan et al. [7] informs that the term requirement could be defined as something that is a desire or necessity to a customer [8], [7], [9], [10].

There are others definitions for the term, such as, a requirement is a defined behavior, characteristic or property, to be assumed for an object, a person or an activity which has to assure a certain result in a value creation process [11], [9] or a requirement typically refers to some aspect of a new or enhanced product or service [12], [13], [14] and, yet, a requirement is a collection of needs arising from the user and various other stakeholders [12], [15]. However, one of the most widely used and accepted definition comes from IEEE 610.12- 1990 standard [16] which defines a requirement as: (1) A condition or capability needed by a user to solve a problem or achieve an objective; (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents. A documented representation of a condition or capability as in (1) or (2). Still, as greatly summarized by

Ilin et al. [16], requirements consist of needs of different stakeholders: users, organization, industry (dictating standards) and others, which need to be addressed and properly met.

Specifically in the software development area the Software requirements express the needs and constraints placed on a software product that contribute to the solution of some real-world problem [17]. Elicitation, on its side, is all about determining the needs of stakeholders and learning, uncovering, extracting and /or discovering needs of the users and other potential stakeholders [18], [19].

In the software area, requirement elicitation is recognized as one of the most critical knowledge intensive activities of the development of software [20], [21] because at the end is the activity of translation the desires and needs of the stakeholder in a format that can be comprehended by the development team. Since a software requirement is a property that must be exhibited by something in order to solve some problem in the real world [17], [22], a correct definition or elicitation of the necessary requirements is the fundamental stone for any successful project which leads us to another important definition, the requirements elicitation techniques.

As well expressed, the Software Requirements knowledge area (KA) is concerned with the elicitation, analysis, specification, and validation of software requirements as well as the management of requirements during the whole life cycle of the software product [17] and all starts with the requirements elicitation techniques which are the means by which systems analysts determine the problems, opportunities, and needs of the customers [23], [24], [25]. The requirements elicitation activity, either in local development or in global software development projects, relies in communication and cooperation between stakeholders which makes collaboration crucial for the success of this activity [26], [27], [28], [29], [30].

Many requirements elicitation methodologies and techniques exist, all with the common aim to assist analysts in understanding needs [8], [31], [32], [33],[34],[35]. It is true that different requirements elicitation techniques [36], [37], [38], [39] have been developed by researchers to aid analysts in effectively determining user needs, however the requirements engineering process, specially during elicitation phase needs to be handled carefully by applying appropriate techniques effectively [40]. The task of collecting requirements with quality is hard, so analysts need to have thorough knowledge about almost all the techniques and only then they can select the appropriate technique. One technique might be suitable in developing one project notwithstanding may not be suitable for the other [32].

So, as important as knowing the techniques, software practitioners[1] must be able to

[1] We use the term software practitioners or practitioners to refer to individuals working in the software industry in a large variety of roles and positions. This includes employees who analyze, design, develop,

properly select the practice, tool or method that best fits with the context, environment, stakeholders and development team in which the software project will be developed since applying inappropriate techniques goes will probably result in negative impacts on the system development [32]. Hickey and Davis [41] argue that one methodology or technique cannot possibly be sufficient or appropriate for all conditions and that a methodology should be chosen based on the specific situation [42], [39], [43], [44]. Similarly, using a variety of techniques helps to ensure uncovering most of the requirements and therefore resulting in more effective requirements elicitation [32]. In summary, during elicitation, weakness of one technique can be neutralized by some other techniques [39], [32] to assist in overcoming challenges.

There are different factors that needs to be evaluated when selecting the appropriate elicitation technique, such as, project area and scope, company process and business directions, time and resources, individual preferences and level of knowledge of the techniques. In this work we engaged into the last factor (knowledge), which led us to perform this study presenting several techniques, each with their own strengths (or pros) and weakness (or cons).

Due to the importance of the requirements elicitation phase to successfully meet the customer or user demands, practitioners need to identify the environment, stakeholders and scenarios and then be able to properly select the best techniques to carry on the project or service. To assist the practitioners, we decided to investigate the use of elicitation techniques, aiming to point out their strengths and weaknesses (pros and cons) as well as the major challenges of the elicitation phase.

1.1 Research Problem

Requirements elicitation is one of the first activities that tries to define the project scope and elicit user requirements [26], [45], [46]. Requirements will come from many different stakeholders, involve multiple disciplines (e.g., sensors, scientific computation, artificial intelligence), and be presented at varying levels of abstraction [47], [48], [49]. In general, there are different sources for inconsistencies between requirements and these may cause problems in the success of the software development [50], [51].

The group of users and also any interested person or company are called stakeholders and, as famously coined by Freeman stakeholders are defined as “any group or individual that can affect or is affected by the achievement of a corporation’s purpose” [52], [53] and, most important, during a project it is usual to have stakeholders with opposite

implement, operate, maintain, or manage software (for instance, software engineers, software testers, product owners, software consultants, and data scientists).

views for the system or product leading the development teams to have a hard time in defining what is really important and what is not. Systems have different stakeholders with diverse interests that usually interact with each other and causes conflicts since the goals of different stakeholders are often incompatible [50], [54].

The success of the requirements elicitation activity gives high impact on the achievement of the goals set for RE, which leads to the development of correct application [26], [55], since one of the foundation basis of any project are the requirements elicitation activities. As stated by Pacheco et al, the requirements elicitation techniques are the means by which systems analysts determine the problems, opportunities, and needs of the customers [39], so each method can contribute to elicit data from different sources, while producing different artifacts documenting the proposed solution [56]. It is a fact that there are already several techniques in requirements engineering that help software engineers to elicit requirements with different stakeholders at the same time that some of these techniques may not always be effective in requirements elicitation [46].

There is a list of factors that influence the effectiveness and applicability of techniques such as, historical information and lessons learned repositories, organization's culture, skills of the business analysis practitioner, stakeholders who are involved and their group dynamics, etc. [57]. In other words, it depends on specific environments, personnel involved and context.

To know and properly select those techniques is, therefore, important for any practitioner or team to succeed to have a better quality on requirements and succeed with the project, product or service. Some of the difficulties related to requirement extraction could be: (i) it may not be related to elicitation techniques [58], (ii) because of not the best selection of elicitation techniques - it is difficult for requirement engineers to choose the right techniques in the right situation [59], [60], [58], [61], [44], (iii) due to project complexity, or insufficient time, or lack of knowledge of new techniques [23], [62], [63].

Specially on item (ii), the selection of elicitation techniques, although there are studies providing recommendation on the requirements elicitation techniques best selection [64], [39], [65], [66] and it is not the intention of this study to produce another set of recommendations based of empirical discussions. On the other hand, it is important for software development practitioners, and especially, the requirements engineers to have better knowledge on elicitation techniques to select the proper technique for requirements elicitation as it is crucial to improve the selection of the techniques [67], [39], [44].

Therefore, this work aims to provide assistance on selecting and using some of the best known techniques identifying them with a brief description and completing with pros (strengths) and cons (weakness) of the techniques confronting it against the identified challenges in Requirements Elicitation. Based on the goal already stated, the work is divided in three phases. First, a systematic literature review was conducted to identify studies that investigated the pros and cons of an elicitation technique, as well as, the challenges already discussed in the literature for Requirements Elicitation. Second, a survey as held with the software development teams to confirm the gaps and challenges of the most significant and known requirements elicitation techniques. Later, the identified challenges, pros and cons were organized together with the basic information and examples resulting in a guide to be used by practitioners and all stakeholders that take part in the requirements elicitation phase. Finally, as part of the guide, some techniques combinations were suggested in an attempt to use RE Techniques strengths to overcome the identified challenges. There, to steer our goal, research questions were defined, as presented in section 3.1.2.

1.2 Justification

Software that fails to deliver its promised functionality can have devastating consequences [42] therefore the elicitation of requirements represents an initial and critical stage in the development of computer based system [68]. In fact, requirements elicitation is considered the first, foremost and crucial stage of a requirements engineering process [33] and crucial to gather the information and data needed to specify the relevant requirements [69].

The Standish Group as well the Forbes expert panel mentioned that "although there are many possible reasons for these failures, problems related to understanding users' needs are consistently identified as among the most important" [70], [23], [71], [72]. Later on, studies went deep in understanding the issues. As interesting depicted by Davey [73] the problems were classified into nine categories:

1. There are human aspects of RE that preclude simple communication between consultant and client [74];

2. The language of humans is not always suitable for technological solution [74];

3. Requirements change as the project proceeds [75];

4. Clients will sometimes ask for requirements that the organization does not need or, in other words, are irrelevant [75];

5. The client cannot say what the business needs [74];

6. Some clients do not want to help you with the project (reluctant to participate) [75];

7. RE failed because it was not done properly [55];

8. RE is not deterministic [55].

Although this classification is not recent, it gives us a view about the areas when the core of the issues reside. As seen by the items one (1), two (2), four (4), five (5) and six (6) are, in its essence, communication problems. In addition, problem three (3) is related to volatility. Furthermore, problem number seven (7) is a large category that can be related to any task inside the requirements engineering phase.

The main objective of the elicitation technique(s), used to elicit requirements, is to discover many possible issues in the process requirement elicitation, which, in turn, facilitates obtaining appropriate software product for stakeholders, which can fulfill their requirements [76]. As mentioned by many studies, communication is the main activity for elicit the requirements from the user [77], [78], [79], [80], [81], [82], [83], [84], [85] as, for instance, the requirements elicitation process is a synthetic process consisting of intensive social communication that involves critical activities required for accurately capturing the requirements/needs of diverse stakeholders [67], [86].

There is a need, in any software development project, for engineers and specialists to be able to convert the user needs into a series of requirements [62], [87] in a way that developers can create a solution for it.

Despite advances in software development and requirements engineering over the past twenty five years, the requirements are almost never stable and fixed [88], [89], [90] and software projects still experience numerous changes during their life cycle [91], [92]. That view continues to drive software requirements as changes are an intrinsic characteristic of the software engineering discipline compared to other engineering disciplines [93]. In other words, the changes of the requirements are a certainty [28].

This is called requirements volatility and has many similar definitions such as the extent to which the initial requirements change over the trajectory of a project [42]; the tendency of requirements to change over time reacting to the evolving needs of customers, stakeholders, firms, and the work environment [94]; the emergence of new requirements or modification or removal of existing requirements [95], [96], [94], [97]. All of them relate to the fact that change is inevitable and teams must be prepared. It is necessary to overcome gaps by constructing a strong technical knowledge in the teams which is clearly one type of knowledge required to successfully develop a software-based system [42], [40], [98]. At the end, this author understands that before even think about the development process itself it is necessary to have the technical knowledge and how to apply them for each step throughout a software development project. Moreover, requirements volatility (problem 3) might also arise from the project team's failure to elicit the right requirements from the system's users [42] and there is growing consensus in the software development community that it is not the technique per se that matters, but the fit between the technique and the specific project context in which it is used [42], [32]. In addition, the technique is strongly related to the context in which it is carried out, the specific characteristics of the project, the organization, the environment, the experience, and knowledge of the analyst, as well as the characteristics of the elicitation technique employed [39].

There are a lot of elicitation techniques for requirement elicitation, but among them, some techniques are appropriate to use in some specific condition, while others are not that much suitable to use [99], [32] and since errors in this elicitation phase could be transferred in the subsequent phases of the software development and compromise the overall process [69]. Studies [100], [39], [38], [67], [32], [37], [101], [34] had been made over the years on the requirements elicitation phase and the techniques to be used on it.

Still, most models of requirements elicitation focus on specific methodologies or techniques that follow paradigms of goal-oriented, scenario-based, or viewpoint-oriented [19], and the applicability of these approaches is dependent on the requirement types gathered, the environment of the project, and features of the target technique [102] leading to a inevitable conclusion that development teams must be familiar with a great number of techniques with their strengths and weaknesses in order to be able to properly select be best fit for a project or environment.

In general, as mentioned by some studies [100], [103], [39], practitioners tend to choose based on the following grounds: it is the only technique they are acquainted with; it is the favorite technique for all situations; there is a methodology that prescribes a particular technique; they guess the technique is effective under the existing circumstances.

This decision can bias the elicitation results, degrade the quality of the output requirements, and, ultimately, have an impact on the quality of the final software product [18], [103], [39]. So, there is a scenario where requirements and their relationships are not correctly identified because the contextual conditions that hinders the requirements elicitation phase are not taken into account by engineers, the so-called implicit requirements [100], [104], sometimes because they are not using the most favorable techniques for the environment [39], [46].

This is the view that drives this study, which aims to provide a mapping tabulating strengths and weaknesses of techniques that can be used by the practitioners helping them to better select the technique according to the environment and characteristics of the project. Thus, the objective of this dissertation is to try to tie the loose ends and voids on the techniques, bringing as a benefit, the presentation of what are the strengths and best uses for them.

1.3 Research Goals

1.3.1 Main Goal

The goal of this study is to map the techniques for the community of practitioners to improve the knowledge on these techniques containing basic information (name and description) plus pros and cons, challenges of use and suggestions on possible combination of use with complementary techniques. By creating a guide we are focusing in presenting a clear view of the most relevant techniques, not only simply describing but helping the community to be able to use them. As a secondary objective, we hope to provide suggestions on combinations of use of techniques that can assist software practitioners in having better results from the software requirements elicitation process.

1.3.2 Secondary Goals

To achieve the general objective of this work, the following specific objectives were defined:

- To conduct a systematic literature review to identify studies that investigated techniques for requirements elicitation in a theoretical or practical approaches through principles, challenges, strengths and weakness;
- To carry out a focused survey on the regional community to find out possible challenges, pros and cons not identified by the systematic literature review;
- To organize the techniques found and select to present in the proposed map;
- To increment this map with some possible combination of the highly used and promising techniques.

1.4 Expected Results

- Identifying a list of techniques (tools, methods and practices) for eliciting requirements that are presented in the researched papers;
- Development a study containing valuable information about those techniques, including pros and cons, description, challenges and examples of use;
- Map and tabulate all collected information.

1.5 Research Methodology and Survey

As mentioned by Snyder [105], building the research on and relating it to existing knowledge is the building block of all academic research activities and continues stating that relevant literature is essential for all research disciplines and all research projects [105].

The need for a systematic literature review (SLR) arises from the necessity of researchers to summarize the existing information about some phenomenon in a thorough and unbiased manner [106]. Templier and Paré [107] suggested that there are four types of literature reviews: narrative, developmental, cumulative, and aggregative.

Narrative reviews assemble and summarize existing literature on a specific topic, providing a comprehensive understanding of the current state of knowledge in the area which is the type of literature review used on this work. On its side, the aim of developmental reviews is to provide new conceptualizations, based on previous research.

The developmental review involves a systematic search of the literature to be reviewed, while the narrative review addresses an illustrative sample. Cumulative reviews synthesize existing literature (as narrative reviews) but further aim to compile empirical evidence to map bodies of literature and to draw overall conclusions regarding particular topics. Aggregative reviews test research hypotheses or propositions. By collating and pooling prior empirical data, they provide validations of pre-specified theoretical models and propositions [108].

The different guidelines and approaches for systematic literature reviews, were also discussed in Snyder's study [105]. Since this work has been done by a single researcher, the steps undertaken were the ones from Kitchenham's work [106]: Developing a protocol; Defining the research question; Specifying what will be done to address the problem of a single researcher applying inclusion/exclusion criteria and undertaking all the data extraction; Defining the search strategy; Defining the data to be extracted from each primary study including quality data; Maintaining lists of included and excluded studies; Using the data synthesis guidelines; and Using the reporting guidelines.

The SLR was composed of automatic and manual searches, performed in digital libraries. The SLR was performed to identify the methodologies and techniques used. Hopefully, we were able to decrease the bias related to the analysis of the papers and studies found either from the automatic research or from the manual investigation. From our point of view, the approach of a systematic literature review study is very appropriate for the purpose and help reaching the objectives established for this work.

In addition, we conducted a opinion-based survey with practitioners to verify whether the perception of the Agile Software Development (ASD) community was similar to the data uncovered from the Systematic Literature Review. There was no restriction in terms of respondents, with people engaged with ASD being able to respond to the survey.

Despite the intention of having a broader set of responses, either due to the circle of relationships or the difficulty of spreading the survey, most of the responses received were regional, that is, from Brazil, more specifically, from the regions of the Federal District and São Paulo. Even considering this limitation, the responses were useful to demonstrate that the community's view is very close to the data found in the SLR.

We also conducted two Focus Groups sections with nineteen (19) specialists and discussed about this study, techniques, challenges, and combinations of techniques.

1.6 Publications

1. Edna Dias Canedo, Angelica Toffano Seidel Calazans, Geovana Ramos Sousa Silva, Pedro Henrique Teixeira Costa, Rodrigo Pereira De Mesquita, Eloisa Toffano Seidel Masson. Creativity and Design Thinking as Facilitators in Requirements Elicitation. International Journal of Software Engineering and Knowledge Engineering, pages:1–32, 2022. https://doi.org/10.1142/S0218194022500607.

2. Rodrigo Pereira de Mesquita, Geovana Ramos Sousa Silva, Edna Dias Canedo. On the Pros and Cons of Elicitation Techniques in Agile Software Development. Information and Software Technology. 2023. (under review)

1.7 Dissertation Outline

This work is organized into five Chapters, in addition to this one, consisting of:

- **Chapter 2:** the theoretical background presents some definitions of requirements as well as the most relevant development life cycle models to demonstrate the path from the first frameworks to the agile model, probably the most used in the recent years. Also taking into consideration the focus of this study in Agile Software Development (ASD), it shows the principles and practices extracted from the Agile Manifesto as a base for agile practices.

- **Chapter 3:** presents the Systematic Literature Review (SLR) and all the steps of the investigation under this study to answer the proposed Research Questions (RQ) as well as the survey held with the practitioners community and its results.

- **Chapter 4:** presents the description, strengths and weaknesses of each analyzed technique, as well as, the validation sessions containing analysis of the techniques and handful combinations.

- **Chapter 5:** presents main conclusions of this work and the future work.

Chapter 2

Software Development Life Cycle and Requirements Background

This Chapter presents the theoretical backgound for this study, starting with some definitions about requirements, software requirements, types of requirements, as well as brief descriptions on life cycle models and, in special, agile model and its principles, views and practices. There are all kinds of sizes for a software development project and software product. That being said, as we will elaborate further, although there are many phases to go through and many process models to follow in order to achieve, the requirement elicitation is the most crucial stage of requirement engineering, and many practices have been used for precise requirement elicitation [109].

2.1 Requirements

So, starting from the basics, all projects begin with a statement of requirements [110] whether it be an innovator's simple statement of concept, a formal Statement of Work (SoW) [111], system requirement specification [51], or a marketing analysis [112] and we can add, in the current world, a user story. A statement of work (SOW) is a document that defines the work activities, deliverable, and planned efforts that someone must perform in order to perform specific work for a client [111].

Since the 90's there is some evolution on the way the term requirements is described. The Institute of Electrical and Electronics Engineers (IEEE) Standard [113] defined the requirement as:

- (1) A condition or capability needed by a user to solve a problem or achieve an objective.

- (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.
- (3) A documented representation of a condition or capability as in (1) or (2).

Later, in 2017, a similar definition was expressed by International Organization for Standardization (ISO) / The International Electrotechnical Commission (IEC)[114]:

- (1) statement that translates or expresses a need and its associated constraints and conditions.
- (2) condition or capability that must be met or possessed by a system, system component, product, or service to satisfy an agreement, standard, specification, or other formally imposed documents.
- (3) provision that contains criteria to be fulfilled.
- (4) a condition or capability that must be present in a product, service, or result to satisfy a contract or other formally imposed specification.

More recently, ISO/IEC/IEEE [115] defined the requirement as a statement which translates or expresses a need and its associated constraints and conditions, with the following notes to this definition:

- Note 1: Requirements exist at different levels in the system structure.
- Note 2: A requirement is an expression of one or more particular needs in a very specific, precise and unambiguous manner.
- Note 3: A requirement always relates to a system, software or service, or other item of interest.

Also, conditions are described as measurable qualitative or quantitative attribute that is stipulated for a requirement and that indicates a circumstance or event under which a requirement applies whereas constraints are depicted as externally imposed limitation on the system, its design, or implementation or on the process used to develop or modify a system [115]. So, although the term requirement can be used widely throughout the different sciences and knowledge areas, in Computer Science these definitions can be summarized as a condition, necessity or capability that is desired for a system or application. So, software requirements are competencies that must be achieved or contained by a finally delivered product or its modules to fulfill an agreement, condition, or other officially required documents [109].

Although there are studies [51], [116] that embrace more detailed categories or specific divisions, such as, technical requirements, user requirements, security requirements, privacy requirements, ethical requirements, quality requirements, operational requirements, the major and accepted categorization division is between a functional and a non-functional requirement, traditionally used by many authors [109], [117], [2], [118]. From the definitions found in the literature, we highlight Pohl et al. [119] that states:

- **Functional Requirement** is a requirement concerning a result of behavior that shall be provided by a function of the system [119]. The Functional Requirement is, yet, defined in the International Standard - Systems and software engineering–Vocabulary by IEEE [120] as: 1. statement that identifies what results a product or process shall produce; 2. requirement that specifies a function that a system or system component shall perform.

  Another definition states that functional requirements are used to describe what the system must do, also what behaviors the system must have or what the system should accomplish [51] and elaborates that functional requirements specify specific results the system should attain [51].

- **Non-Functional Requirement** defines desired qualities of the system to be developed and often influence the system architecture more than functional requirements do [119]. Again, we resort to the International Standard - Systems and software engineering–Vocabulary by IEEE [120] which, in general defines the Non-Functional requirement as a performance requirement and gives a more detailed sentence: "software requirement that describes not what the software will do but how the software will do it'.

  Chung et al. [2] and, more recently, Habibullah and Horkoff [121] described these attributes in a form of a mapping tree. Moreover, non-functional requirement can be viewed as a set of restrictions imposed on the system to be developed, establishing, for instance, how attractive, useful, fast, or reliable it is [122], [123]. It also can be depicted in terms of something that describes how well, and under what conditions, the system must operate within in performing the functions elucidated in the functional requirements or describe the performance and quality attributes the system must meed [51]. Also, there is a representation by ISO/IEC 25010:2011 showing quality characteristics that is very similar to Boehm's model.

  Usually, despite no firm pattern being established for the non-functional characteristics, some of the most cited of these attributes are: Accuracy, Adaptability, Extensibility, Integrity, Testability, Usability [124], [121].

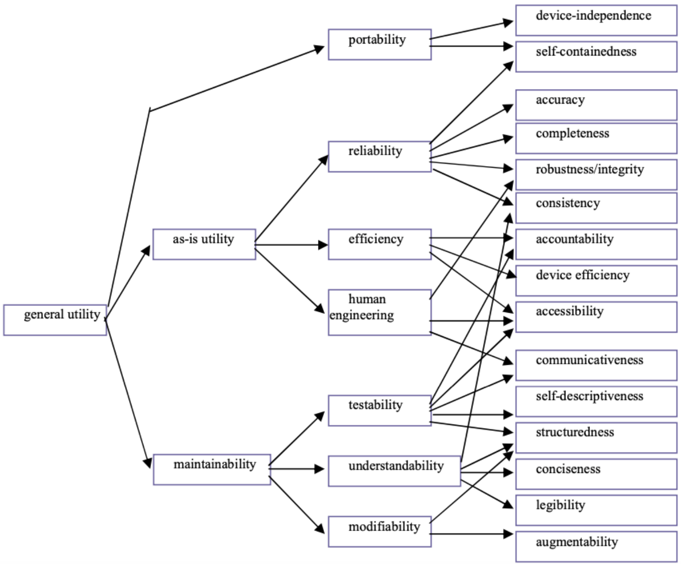


Figure 2.1: Non-functional requirements Boehm's model [1], [2]

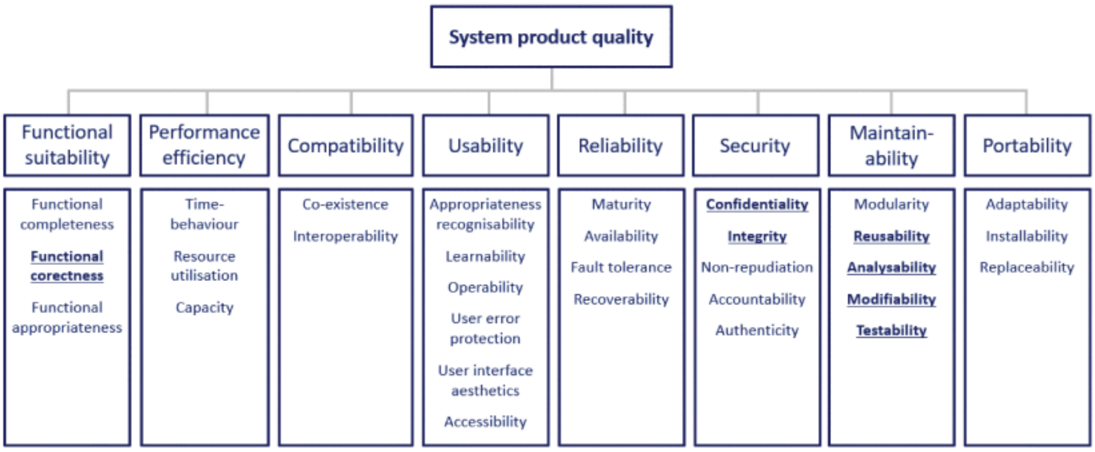


Figure 2.2: Non-functional requirements by ISO/IEC [3]

So, as stated by Purnamasari [125], “functional requirement means what the system should do and non-functional requirement means what obstacles exist on the system or its development” [125]. It is description of the desires and needs for a system, software or application. Moreover, requirements changes regularly due to unbalanced business conditions. New ideas and new technology come to fruition, new systems come into play, and users may change their mind [126], and that’s why it is necessary some process regarding the engineering.

2.2 Life Cycle Models

There are different models and frameworks that aims to organize the steps in order to better understand the customer and all stakeholders. Regardless the specific frameworks or adaptations, there are a some life cycle models well discussed in literature [127], [128], [4], [129], such as: Waterfall [4], Incremental Model [5], V-Model [4], Spiral Model [130], Rapid Application Development (RAD) [5], Unified Process Model [41] and, more recently Agile Methodology (or Model) [131], [132], [133], [134]. In all cases, we highlight the main information about the base life cycle models just to draw a line from the beginning to Agile.

2.2.1 Traditional Models

These models are referenced as “traditional” due to in this approach all process for developing a system are executed in a sequential order, in which progress is seen as flowing steadily downwards through the phases from the conception through operation and maintenance [117], [135]. Each of the traditional models has its own prescription however all based on the waterfall idea.

- **Waterfall model** is a sequential model when each phase of development proceeds in order without any overlapping [128]. The model begins with establishing system requirements and software requirements and continues with architectural design, detailed design, coding, testing, and maintenance. [4] Requirements should be clear before going to next phase of design [128]. model emphasizes planning in early stages, it ensures design flaws before they develop [4].

- **V-model** is a sequential path of execution of processes when each phase must be completed before the next [4]. The left leg of the V model represents the evolution of user requirements into smaller components through the process of decomposition and definition and the right side represents the integration and verification of the

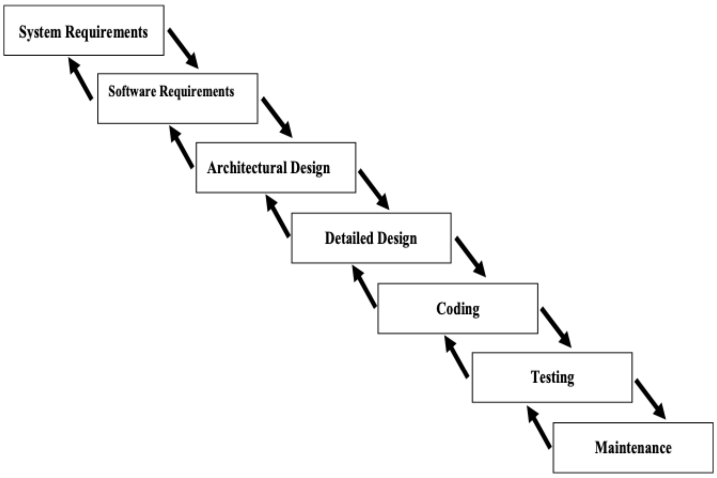


Figure 2.3: Waterfall model [4]

system components into successive levels of implementation and assembly [127]. Testing is emphasized in this model more than the waterfall mode [4].

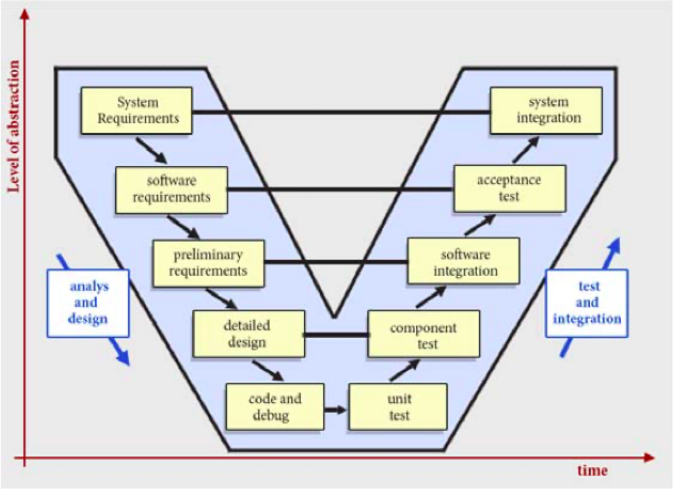


Figure 2.4: V model [4]

- **Spiral model** represents a paradigm shift to the risk-driven approach [127]. The spiral model has four phases: Planning, Risk Analysis, Engineering and Evaluation [4]. At completion of each cycle a review process takes place to ensure that stakeholders are committed to the approach to be taken during the next cycle [127]. Specific objectives for each phase are identified [4].

- **Incremental model** combines elements of the waterfall model in an iterative fashion [129] and is known as the iterative waterfall model, can be viewed as a three dimensional representation of the waterfall model [127]. Basic requirements are

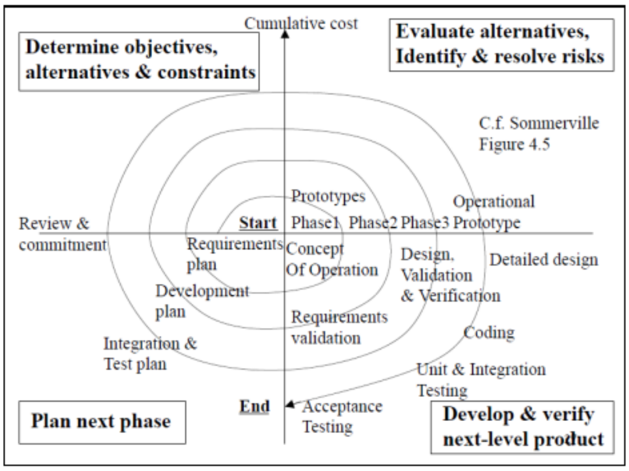


Figure 2.5: Spiral model [5]

addressed in the first increment and each linear sequence produces deliverable increments of the software [129].

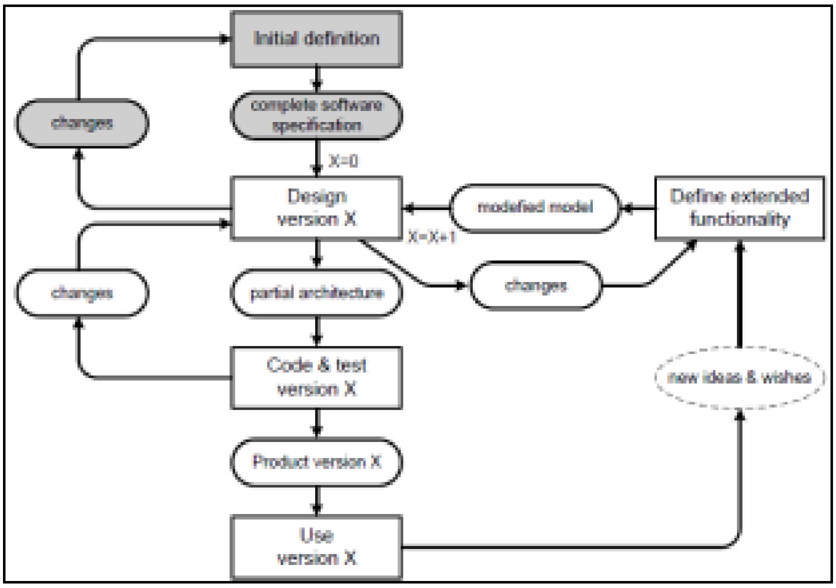


Figure 2.6: Incremental model [5]

2.2.2 Agile Methodology

Agile development approaches and the traditional methodologies are considered the opposite of each other [109]. The term “Agile” means the ability to create and respond to any change [136] being a model construct to provide flexibility in the form of welcoming requirement changes even late in the development cycle [136]. In comparison with traditional models, agile software engineering methods are useful and trendy but are missing a proper structure, whereas conventional methods are structured but heavy to implement for easy requirement change management [109].

The use of Agile Software Development (ASD) model and its methods is increasing and occupying more and more space within the industry. There are some reports that state that [137], [138], [139]. According to the 15th State of agile report [140], Agile adoption within software development teams increased from 37% to 86% in 2021 and 94% report their company is practicing Agile. The agile manifesto is the document that drives all basic agile methodologies [132], [131] and it describes purposes and practices to be followed. Regarding the purposes, it informs that the ones on the left should be preferred in comparison with the ones on the right.

- Individuals and interactions over processes and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.

Also, the Manifesto presented some principles [132], [131]:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.

- Simplicity—the art of maximizing the amount of work not done—is essential.

- The best architectures, requirements and designs emerge from self-organizing teams.

- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Besides the manifesto, there is no single and exact definition of agile, but rather there are various methods that share the same philosophy which call themselves agile. Most methods have iterations with continuous refinement of plans and goals. However, each method has its own practices and terminology, putting emphasis on different issues in software development, and hence can be adopted to suite for different situations.

One goal of agile methodologies is to improve efficiency by reducing the cognitive burden of capturing and updating complex system Software Development (ASD), specifications [141]. As well mentioned, Agile development triggers a reshaping of requirements engineering by relaxing the ordering of requirements engineering activities and allowing requirements engineering activities to run in parallel with software implementation [142]. One major critics mentions the agile approaches tend to focus solely on scoping and simplicity rather than on problem solving and discovery [134].

For one side, Agile methods strive to respond to requirements changes by integrating requirements, design, implementation, and testing processes as simplified processes that focus on changes acceptance and adaptation [143], meaning it is easier to embrace the inevitable changes that come from software design. On the other hand, agile practices are often undervalued, in part due a lack of theoretical foundations [144].

Since Agile Software Development typically does not use formal requirements modeling and do not prioritizes documentation, it usually works with small iterations with frequent deliverable, the development process is dynamic [145], and, by exclusion, makes difficult to analyze large collections of requirements and identify conflicts and track for completeness, consistency as well as provides little basis for evaluating the potential impact of changes to requirements [141].

Agile methods universally rely on an incremental approach to software development, and delivery. In incremental development, specification, development and validation activities are interleaved rather than separate, with rapid feedback across activities. Each increment of the system incorporates some of the functionality that is needed by the customer [146].

2.2.3 Requirements Engineering

Requirements engineering is considered a key area of the software development process, which can significantly influence the result of the project [33], [69], [20]. At the same time, many software development teams fail because the team members do not understanding what are the necessary requirements for the product outcome [147]. Sometimes named as requirements gathering, others as requirements elicitation or requirement engineering process is a decision-centric activity that produces specifications including both functional (FR) and non-functional (NFR) requirements [148], [149], [150].

As mentioned by Jarzebowicz [150] the Requirements engineering process is a holistic process that focuses on capturing all requirements and their interrelatedness, thus it is hardly possible to design a separate and isolated process for a specific requirements category, such as, non-functional requirements or security requirements instead the requirements engineering process provides the appropriate mechanism to understand what the customer wants, analyzing the needs and its relations, verifying and negotiating the feasibility and solutions [151], all of this notwithstanding not for any specific category but in a generic way. Requirement elicitation is all about knowing the needs and desires of stakeholders [20].

Requirement elicitation is the method to collect the actual requirements of the project [136] or concerned with defining what the system is required to do and the constraints under which it needs to operate [152]. In agile software development RE activities are not explicit, in part because they appear to have less distinct boundaries in this context than in more traditional software development processes [153].

The elicitation process can be done by collecting the opinions, choices or constraints from client, end users, development teams as well as other stakeholders of the project [136]. In traditional software development, requirements are prioritized at the initial level [136], in opposite of the requirements engineering in agile development which is more informal and based on the skills and knowledge of individuals [133].

While agile methods strive to respond to requirements changes by integrating requirements, design, implementation, and testing processes as simplified processes that focus on changes acceptance and adaptation [143], [154], [135], they disregard the importance of Requirement Engineering for the success of the development of the software as this activity is seen as bureaucratic in agile methods [155].

Furthermore, requirement elicitation is the most crucial stage of requirement engineering, and many practices have been used for precise requirement elicitation [109]. It is also perceived as the most important phase in agile software development [126]. Moreover, to ensure that the requirements elicitation process is the best it can be, requirements elicitation techniques come into play.

The requirements elicitation techniques are used to determine the software requirements (SRs) according to the need of the stakeholders [156]. They are basically the ways and procedures to obtain user requirements and later translate into functions in the system to be developed so that it satisfies the needs of all stakeholders involved [146].

Depending on the context, the environment or the complexity of the project, it might make more difficult to achieve a consensus or common understanding of one or more requirement and that is why the Requirement Elicitation techniques are so important and are object of this study. As an example stakeholders geographically distributed with different timezone might makes complex create meeting schedules with all the team.

2.3 Related Work

We have searched for surveys and systematic literature reviews (SLR) and/or mapping studies dealing with Agile and its elicitation techniques as well as challenges. Several authors deal with agile practices in different contexts, some more focused in the requirements engineering process as a whole, others in the agile software development challenges related to the process. In a related but different context, some studies focus in the requirements elicitation techniques only.

One of the first studies we came across from Sutcliffe and Sawyer [101] focus solely on demonstrate the unknowns of the elicitation process. Sharma and Pandey [37] highlights some requirements elicitation techniques along with some description of the techniques and some of their key features. The study does not focus on challenges, strengths or weaknesses.

An work from Shadab Khan et al. [36] had a interesting study on a specific classification of elicitation techniques, dividing into direct and indirect approaches. The study did not go deep on how direct and indirect approaches could be used to assist the practitioners but worth mention. Tabbassum Iqbal [38], on his hand, presented a study to complete Shadab Khan et al. [36] by trying to classify some of the most known techniques in the direct or indirect approach.

Inayat et al. [157] presented the results of a SLR study dedicated to requirements engineering challenges. Although they present a interesting view of Agile challenges, they searched and discussed their focus on agile practices as a tentative to address such challenges and not focused on any requirements elicitation techniques as part of a possible solution.

In 2015, Yousuf and Asger [32] delivered a interesting study giving an overview of a few elicitation techniques, their characteristics, advantages and disadvantages. Some of these techniques, such as, interviews, questionnaires, ethnography, scenarios, laddering and prototypes are included in this study and the findings from Yousuf and Asger were put to a good use. The study however do not mention challenges from requirements elicitation phase that could be overcame with some techniques combination.

Schon et al. [158] presented a work focused on Agile challenges however not focusing on elaborate on how the techniques could assist in solving the identified challenges. Zamudio et al. [159] reported a SLR study on requirements engineering techniques in Agile Software Development with the focus on identifying which techniques were to be used in the major Agile methods, such as Scrum, XP, Crystal Reports and Kanban. The study do not discuss challenges, combinations, strengths or weaknesses of the techniques aiming on listing the most used techniques for each of the major approaches.

On her side, Pacheco et al. [39] performed a systematic literature review to identify other studies about elicitation techniques and focused their work in discover, first, which techniques were used for eliciting software requirements and which of them were considered mature techniques and could improve the elicitation effectiveness. They relied on third-party studies that previously stated a technique as effective. Therefore, Pacheco et al. [39] focused in listing characteristics that could justify some techniques to be effective. Although the work do not discuss weaknesses or challenges, this work was useful as we identified that some of these techniques could be viewed as strengths of the described techniques.

Alsaqaf et al. [160] put together a list of challenges and focused their work in confirming those with practical studies by interviewed a group of selected software developers, testers, analysts, architects, managers and scrum masters trying to address the source and mechanisms behind the challenges.

Palomares et al. [161] focused their study in interviewing professionals and came up with a list of the techniques they used and the challenges they encountered. Since the challenges were analyzed from the viewpoint of the interviewed specialists, the challenges were demonstrated based on their relationships with the projects therefore presented as “related to process”, “related to stakeholders”, “related to evolution of the system” and “related to stability of the requirements”, among others.

Moreover, Li et al. [102] brought a framework and process based on quantitative methods aiming to evaluate the attributes with high influential on techniques selection. From their side, Jarzebowicz and Weichbroth [150] focused the work in non-functional requirements and provided a discussion on the techniques, challenges and difficulties applied to this specific type of requirements [102].

Anwar et al. [162] presented their study focused on transform intrinsic or tacit knowledge into explicit requirements. They ran a SLR plus experts judgment to bring up a list of prioritized challenges along with a list of requirements elicitation techniques and their frequency of citation.

The Table 2.1 summarizes the list of related work with the last column providing a brief description about the paper’s goals and what is not discussed in the paper in relation of this work's research questions and objectives.

Year	Ref	Title	Paper goal and gaps related to this work
2013	[101]	Requirements elicitation: Towards the unknown unknowns	Work demonstrates the unknowns of the elicitation process and describes a framework using basic elicitation techniques to discover the unknowns within the process. Paper did not focus on pros and cons or combination of techniques.
2013	[37]	Revisiting requirements elicitation techniques	Highlights some requirements elicitation techniques along with some description of the techniques and some of their key features. The study do not focus on challenges of the elicitation process, and does not present pros, cons or combinations of techniques.
2014	[36]	Systematic review of requirement elicitation techniques	It presents a classification of techniques between direct and indirect approaches. The study did not go deep on the techniques itself, specially in relation to pros and cons.
2014	[38]	Requirement elicitation technique:-a review paper	This paper is a tentative of completing the work from [36] and focus solely classify and separate techniques with direct or indirect approach, using Interview and prototyping as examples. It does not mention anything on techniques characteristics, strengths or weakness.
2015	[157]	A systematic literature review on agile requirements engineering practices and challenges	SLR dedicated to requirements engineering challenges with focus on agile practices as a tentative to address such challenges and not focused on any requirements elicitation techniques.
2015	[32]	Comparison of various requirements elicitation techniques	It presents elicitation techniques, their characteristics, advantages and disadvantages. It covers less techniques than this work and, most important, it does to work with technique combination to try to cover the weakness from one technique by using another.

Year	Ref	Title	Paper goal and gaps related to this work
2017	[158]	Key challenges in agile requirements engineering	Work focused on Agile challenges raised by a panel of experts however did not focused in elaborating how the techniques could assist in solving the identified challenges.
2017	[159]	A requirements engineering techniques review in agile software development methods	SLR on requirements engineering techniques aiming on identifying which techniques were to be used in the major Agile methods, such as Scrum, XP, Crystal Reports and Kanban. The study do not discuss challenges, combinations, strengths or weaknesses of the techniques.
2018	[39]	Requirements elicitation techniques: a systematic literature review based on the maturity of the techniques	Performed a SLR to identify other studies about elicitation techniques. The study focus on listing characteristics that could justify some techniques to be effective. The work does not discuss weaknesses or challenges, as well as, does not provide any view about combining techniques to help improving the elicitation process.
2019	[160]	Quality requirements challenges in the context of large-scale distributed agile: An empirical study	Study lists the challenges and focus the work in confirming it with practical studies by interviewed a group of selected software technology workers trying to address the source and mechanisms behind the challenges.
2020	[102]	Attributes-Based Decision Making for Selection of Requirement Elicitation Techniques Using the Analytic Network Process	Brings a framework and process based on quantitative methods aiming to evaluate the attributes with high influential on techniques selection. It does not mention neither techniques nor challenges.
2021	[161]	The state-of-practice in requirements elicitation: an extended interview study at 12 companies	Interviews professionals and brings a list of techniques they use and the challenges they encountered. No mention about techniques pros, cons and combinations.

Year	Ref	Title	Paper goal and gaps related to this work
2021	[150]	A qualitative study on non-functional requirements in agile software development	Work focus on non-functional requirements and discuss techniques, challenges and difficulties applied to this specific type of requirements, again with no combination, weakness or strengths.
2022	[162]	A Tacit-Knowledge-Based Requirements Elicitation Model Supporting COVID-19 Context	SLR analysis joint with experts judgment to bring up a list of prioritized challenges along with a list of requirements elicitation techniques and their frequency of citation. The study didn't went through any pros, cons or technique combinations.

Table 2.1: Related works

As this work is focused in present the challenges, describe techniques along with their strengths and weakness, as well go through some possible combinations that might assist in overcome the difficulties and reduce the cons from some of the most common techniques, we claim this work as relevant to complete gaps of previous researches.

Chapter 3

Systematic Literature Review and Survey

This Chapter presents the Systematic Literature Review (SLR) and the survey held with the Brazilian development community, as well as its results and the relevant remarks.

3.1 Systematic Literature Review

A systematic literature review, often referred to as a systematic review [163], [106], is a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest. Individual studies contributing to a systematic literature review are called primary studies; a systematic literature review is a form of secondary study [106].

Based on the just mentioned definition, we have conducted a Systematic Literature Review (SLR) to identify the techniques that are described in the literature and used by the industry to elicit requirements for a software or a system and investigate the pros and cons of the most referenced, based on the selected papers (for details, refer to section 3.2.

The initial search yielded both primary and completed SLR studies. To avoid a protocol deviation and possible bias in the results, it was decided that the SLR would be unraveled and the primary studies contained therein would be reanalyzed, meaning the snowball technique as performed and each of the papers that applied to the quality and criteria rules were reviewed.

In turn, such a decision brought another problem, what to do with primary studies that did not adhere to the criterion of the year of the study, in other words, that had been carried out before 2010? For this scenario, since we are reanalyzing each study contained in the SLR, we decided not to consider them. After a careful analysis, the studies presented in the table 3.8 are the SLR that were unraveled with studies from 2010 or newer that were taken to be evaluated. All other inclusion and exclusion criteria were applied during the analysis of the studies selected from these SLRs.

- **Planning, conduction and reporting**:

  A Systematic literature review starts by defining a review protocol that specifies the objectives, PICOC structure, research(s) question(s) being addressed and the methods that will be used to perform the review [106].

  In seeking to answer the posed Research Questions the very first step was to define the search string that led us to retrieve the studies that were evaluated.

  Although protocols does not define a mandatory or specific minimum number of years to address during the SLR, in general, many reviews presents from ten (10) to fifteen (15) years as a range scope for analyzing papers. In defining the initial year for our research we noticed that the year of 2010 was the one that had an exponential grow from previous years therefore we understood that this year (2010) would give us a interesting range of analysis.

  The conduction followed the inclusionexclusion criteria and quality assessment questions and finally, this work is the reporting of those findings.

- **Research string definition**:

  SLR is a protocol-based research method that is used to examine, classify, and assess the current available literature related to a specific study of a specific research area by using inclusion and exclusion criteria [164].

  It was carefully analyzed which sort of papers should be included in this study and therefore had to set-up the inclusion and exclusion criteria in order to initially select the papers to fully analyze. During the review we continued to review the primary and secondary studies that were initially selected, whereas this time the studies were fully analyzed and to be selected.

The research protocol of this work was developed to meet the objective of identifying techniques, or, in other words, methods to support the requirements elicitation process.

3.1.1 Question structure

According to Petersen et al. [165], the fundamental input to be considered is the set of research questions and the very first step to elaborate it is to define the basic concepts of population, intervention, comparison, outcome and context. That is the PICOC structure, based on Kitchenham [106] description, as we present here:

1. **Population** In software engineering experiments, the populations might be any of the following:

   - A specific software engineering role, e.g. testers, managers.
   - A category of software engineer, e.g. a novice or experienced engineer.
   - An application area, e.g. IT systems, command and control systems.
   - An industry group such as Telecommunications companies, or Small IT companies.

2. **Intervention** The intervention is the software methodology/tool/technology/procedure that addresses a specific issue, for example, technologies to perform specific tasks such as requirements specification, system testing, or software cost estimation.

3. **Comparison** This is the software engineering methodology/tool/technology/procedure with which the intervention is being compared. When the comparison technology is the conventional or commonly-used technology, it is often referred to as the "control" treatment. The control situation must be adequately described. In particular "not using the intervention" is inadequate as a description of the control treatment.

4. **Outcomes** Outcomes should relate to factors of importance to practitioners such as improved reliability, reduced production costs, and reduced time to market. All relevant outcomes should be specified.

5. **Context** For Software Engineering, this is the context in which the comparison takes place (e.g. academia or industry), the participants taking part in the study (e.g. practitioners, academics, consultants, students), and the tasks being performed (e.g. small scale, large scale). Many software experiments take place in academia using student participants and small scale tasks.

Table 3.1 presents the PICOC definitions for this work.

3.1.2 Research Questions

The research questions identify the scope of the SLR, guide the research protocol construction and the most critical elements of the SLR [166]. To meet the objectives set out in Section 1.3.1, we set the following Research Questions (RQ) to guide the execution of this research.

RQ.1: What are the existing techniques in the literature to elicit software requirements that can be used in agile processes?

PICOC terms	**Related description**
Population	Requirement Engineers, Developers, Analysts, Managers, Customers, Organizations.
Intervention	Tools, Methods and Practices on Agile Software Development.
Comparison	How to use, pros and cons, challenges.
Outcome	Guide presenting the findings, descriptions, pros (strengths), cons(weakness), Requirements Elicitation challenges, suggestions of RE Techniques combinations.
Context	Papers on literature, from January-2010 to October-2022, that presented or reviewed Agile Software requirements techniques.

Table 3.1: PICOC definition

In this research question we systematically reviewed the literature that studied and identified techniques for requirements elicitation process within agile software development and then presenting the basic information about them, such as, name, main reasons that was created, general description of how it works or main practices, when applicable.

By answering this question allowed this work to narrow and focus the search into the most cited and used techniques as well as, being able to provide a clear description and characteristics of the techniques.

RQ.2: What are the strengths and weaknesses (pros and cons) in the identified techniques, as well as the challenges reported in the literature for the requirements elicitation phase?

In this research question, we aimed to catalog the main challenges identified both in the literature and in the industry by using a systematic literature review research in addition of the strengths and weaknesses (advantages and disadvantages) of the techniques that could help surpass the commonly identified challenges. Also, we held a survey with the practitioners community and and evaluation focus group with specialists to confirm the findings.

Although there are numerous papers which provide a list of challenges and issues, they do by showing the scenario in a generic way. The goal of this question in this work was to find not only the general and most known challenges but try to identify challenges related to specific techniques.

RQ.3: What techniques were combined to improve the requirements elicitation process?

In this research question, the focus was to investigate within the literature for studies or practical cases that combined the elicitation techniques presented by this study.

Also, we analyzed the strength and weakness of requirements elicitation techniques either presented in other studies or discussed in proposed guides and based on these strengths suggested possible combinations that might overcome the identified challenges.

3.1.3 Search String

The aim of a systematic literature review is to find as many primary studies relating to the research question as possible using an unbiased search strategy [106]. As mentioned before, the first rule was the at least ten years of timeline should be included in the research. Also, the main idea of this study was to work with Agile Software Development, as well as to work with techniques that focus on requirements elicitation.

In addition, we have filtered for studies written in English, Spanish or Portuguese. Although there was some risk of not retrieving relevant papers written in other languages we understand this possible bias is minimized due to all relevant conferences request the work being in English or, at least translated. Nevertheless, the filtering was not complete since we also reduced the scope to the Computer Science, decision sciences and engineering subject areas to avoid papers linked to medicine, health professions or any sort of non-related areas.

To improve the initial structure of the search string, some synonyms were determined, as sometimes the requirement elicitation is also known as specification or gathering. All these definition criteria narrowed the scope of the search. After a few analysis we realized the string could not be same for all indexers previously defined to be used, so in order to thrive in the search and cover a more extensive ground on the candidate studies to analyze we decided to adjust the search string for each indexer.

The base string was defined used **Scopus** (Table 3.2) and thereafter some adjustments were made to cover each of the other main indexers. In order to provide a better and more complete view of search routing, we also present each search string generated for each indexer.

The automatic search string used in this research was adapted according to the possibility of using the digital libraries' connectors. Thereby the resultant base string is presented in the box below.

TITLE-ABS-KEY ((requirement) AND (elicitation OR gathering OR specification) AND (tool OR technic OR method OR methodology OR practice) AND ("ASD" OR "agile Software Development" OR "agile requirement" OR "agile practice" OR "agile technique" OR "agile approach")) AND (LIMIT-TO (SRCTYPE , "p") OR LIMIT-TO (SRCTYPE , "j")) AND (LIMIT-TO (DOCTYPE , "cp") OR LIMIT-TO (DOCTYPE , "ar")) AND (LIMIT-TO (SUBJAREA , "COMP") OR LIMIT-TO (SUBJAREA , "ENGI") OR LIMIT-TO (SUBJAREA , "DECI")) AND (LIMIT-TO (PUBYEAR , 2022) OR LIMIT-TO (PUBYEAR , 2021) OR LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014) OR LIMIT-TO (PUBYEAR , 2013) OR LIMIT-TO (PUBYEAR , 2012) OR LIMIT-TO (PUBYEAR , 2011) OR LIMIT-TO (PUBYEAR , 2010)) AND (LIMIT-TO (LANGUAGE , "English") OR LIMIT-TO (LANGUAGE , "Spanish") OR LIMIT-TO (LANGUAGE , "Portuguese"))

- **ACM Digital Library** - in this indexer it was necessary to search using the full text clause since the title clause was not bringing adequate results.

  [Abstract: requirement] AND [[Abstract: elicitation] OR [Abstract: gathering] OR [Abstract: specification] OR [Abstract: engineering]] AND [[Abstract: tool] OR [Abstract: technique] OR [Abstract: method] OR [Abstract: methodology] OR [Abstract: practice]] AND [[Abstract: "asd"] OR [Abstract: "agile software development"] OR [Abstract: "agile requirement"] OR [Abstract: "agile practice"] OR [Abstract: "agile technique"] OR [Abstract: "agile approach"]] AND [Full Text: requirement] AND [[Full Text: elicitation] OR [Full Text: gathering] OR [Full Text: specification] OR [Full Text: engineering]] AND [[Full Text: tool] OR [Full Text: technique] OR [Full Text: method] OR [Full Text: methodology] OR [Full Text: practice]] AND [[Full Text: "asd"] OR [Full Text: "agile software development"] OR [Full Text: "agile requirement"] OR [Full Text: "agile practice"] OR [Full Text: "agile technique"] OR [Full Text: "agile approach"]] AND [Publication Date: (01/01/2010 TO *)]

- **IEEE Explore**

  ("All Metadata":"requirement" AND ("All Metadata":"elicitation" OR "All Metadata":"gathering" OR "All Metadata":"specification") AND ("All Metadata":"tool" OR "All Metadata":"technic" OR "All Metadata":"method" OR "All Metadata": "methodology" OR "All Metadata":"practice" OR "All Metadata":"agile approach")

AND ("All Metadata":"ASD" OR "All Metadata":"agile Software Development" OR "All Metadata":"agile requirement" OR "All Metadata":"agile practice" OR "All Metadata":"agile technique")) AND PUBLICATION (2010-2022)

- **Science Direct** - the same base string was used by this indexer. The only change was to remove the clause related to limit the subject area, due to incompatibilities of the indexers with this clause.

- **Springer** - in this indexer it was necessary to change the clauses not using the "AND" clause which was limiting the results and returning empty in most cases. After the "NEAR" clause was used, with the indexer searching for specific words using the proximity within the sentence, the results were more promising.

  ("Requirements" NEAR/5 ("elicitation" OR "gathering" OR "specification" OR "engineering")) NEAR ("technic" OR "technique" OR "method" OR "methodology" OR "practice") NEAR ("ASD" OR "Agile Software Development" OR "agile requirement" OR "agile practice" OR "agile technique" OR "agile approach") AND NOT "medicine" AND NOT "Mathematics" AND NOT "Psychology" AND NOT "Finance" AND NOT "Humanities"

The search strategy included two stages: (i) automatic search in electronic databases based on the search string presented for each indexer and (ii) manual search in newspapers, conferences, and workshops. The decision was to made to dminish any possible bias of the search string. In general, in case the paper was not included as a result of the automatic research and still there was a reference pointing the study as a review related to elicitation techniques the paper was elected and included to be analyzed (manual search). Once the manual analysis was performed and the view was that the paper could aggregate value for our study, then the paper was included for further analysis.

3.1.4 Databases and Journals

Systematic literature reviews are based on a defined search strategy that aims to detect as much of the relevant literature as possible [106]. Aiming to obtain diverse studies to encompass the theoretical and practical aspects of requirements elicitation techniques to ensure a broad scope for the review, the databases presented in Table 3.2 were the ones taken into account for the automatic search of this work.

Also, the journals and conferences from Table 3.3 were consulted in a manual research. These were selected because they were known to include studies or literature surveys, and to be well known as sources for other systematic literature reviews related to software engineering. Although the Digital Databases probably have indexed the papers which

Digital Databases	URL
Academia	https://www.academia.edu
ACM Digital Library	https://dl.acm.org
IEEEXplore	https://ieeexplore.ieee.org/Xplore/home.jsp
Scopus document search	https://www.scopus.com/search/form.uri?display=basic#basic
Springer	https://www.springer.com/br
Science Direct	https://www.sciencedirect.com

Table 3.2: Digital Databases for automatic research

were selected in manual search, this approach is intended to decrease the bias of the Search String, as better detailed in the 3.5 Section. Also, other sources could be selected, however, we understood that by working with the ones listed in the Tables 3.2 and 3.3 this work would be covering the necessary ground for the stated research questions.

Source Name	Type
Requirements Engineering (RE) Conference	Conference
ACM IEEE International Conference on Software Engineering (ICSE)	Conference
International Working Conference on Requirements (REFSQ)	Conference
International Conference on Human-Computer Interaction (HCII)	Conference
IEEE Transactions on Software Engineering (TSE)	Journal
Empirical Software Engineering Journal (EMSE)	Journal
Requirements Engineering (RE) Journal	Journal

Table 3.3: Journals and conferences

Although there was a possibility of Scopus database overlapping some studies that could be found in other databases and journals, the primary goal was to try to cover as many studies as possible in the systematic literature review therefore we opted to perform the automatic research using the databases, as shown in Table 3.2, even though there was a chance to return a few duplicated studies.

Furthermore, to minimize the duplication of studies the decision was to primary use the digital databases from the Table 3.2, since they usually indexes a great amount of papers and to manually explore the defined journals, as presented in Table 3.3. Finally, although not included directly as a database source for our research, Google Scholar was used to help us locate the manually referenced papers.

3.1.5 Selection Criteria (Inclusion and Exclusion)

According to Kitchenham [106], Systematic Literature Reviews (SLRs) require explicit inclusion and exclusion criteria to assess each potential primary study, so taking into account this precept, we have defined a solid inclusion and exclusion criteria to identify

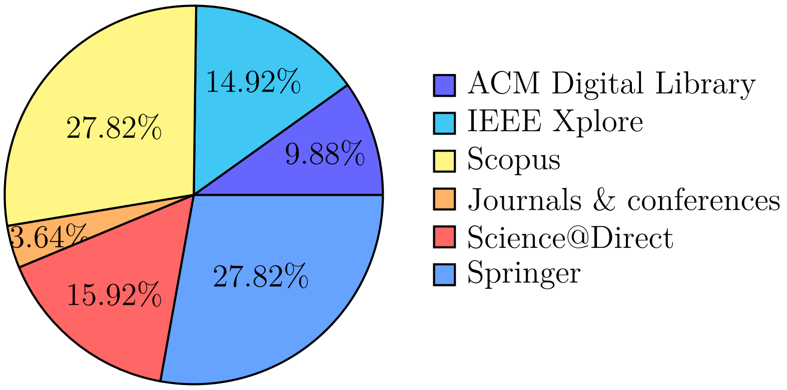


Figure 3.1: Percentage of papers per source

the relevant studies to be included in the research and meet our goals. Table 3.4 presents the selection Criteria.

Inclusion Criteria	**Exclusion Criteria**
IC.1. Studies related to Agile Software Development	EC.1. Studies that present a specific and proprietary approach
IC.2. Studies that describe challenges related to ASD in general or to any requirement elicitation technique	EC.2. Studies published before 2010
IC.3. Studies that reference or describe RE techniques	EC.3. Studies that only provides statistics about a elicitation technique
IC.4. Studies that provide advantages (pros) or disadvantages (cons) of any RE technique that can be applied to ASD	EC.4. Duplicated studies

Table 3.4: Selection Criteria

3.1.6 Conduction of the SLR

Systematic Literature Reviews (SLRs) are based on a defined search strategy that aims to detect as much of the relevant literature as possible [106]. Once all the criteria were determined we start the conduction of the SLR.

The Figure 3.1 shows the contribution of each indexer for this research. Also, for conducting the review we decided to use a tool named Parsif.al[1], which is a free open source web platform for supporting SLRs and contains features to assist the researcher, from the import of the searched papers through the definition of inclusion and exclusion criteria and quality assessment. Moreover, its workflow is based on the SLR process proposed by Kitchenham [106].

[1] https://parsif.al

We initially defined the indexers and databases to perform the automatic search, as already stated in Table 3.2. The execution of the search strings (section 3.1.3) for automatic search returned a total of 338 papers. Furthermore, the manual search and analysis were held by the researcher using the defined journals and conferences (3.3 which resulted in more 18 works potentially relevant for the review. All this search resulted in a total of 496 papers retrieved either from automatic search or from manual search. These papers were collected for subsequent analysis. Table 3.5 shows the initial numbers of the search proceedings, including automatic and manual findings.

Digital Database	Quantity of Studies
IEEEXplore	49
ACM Digital Library	74
Scopus document search	138
Springer	138
Science Direct	79
Manual inclusion	18
Total	**496**

Table 3.5: Number of Studies per Digital Database

Moreover, we went through extraction and quality assessment phases by reviewing each of the papers based on the Quality Assessment questions.

3.1.7 Quality Assessment (QA)

During the step of full analysis, usually known study selection, all the remaining papers were fully read and for each them the analysis being done according to the data extraction criteria from Table 3.6. Section 3.2 presents the cutoff rule.

ID	Quality Assessment (QA) Question
QA1.	Does the study describe pros, cons or challenges for using the literature cited technique in agile software development?
QA2.	Does the study describes the context of using the cited technique in agile software development?
QA3.	Does the study suggest any combination of using the cited technique in agile software development?

Table 3.6: Quality extraction criteria

3.2 Systematic Literature Review Results

The first step of systematic literature review was to remove the duplication, and this was done by comparing the name, authors and publication dates of the papers. In case there was the literally equal one of the similar documents was marked as duplicated. This initial step removed 89 papers from the list.

Furthermore, after filtering the studies applying all the defined criteria and rules, the final list of studies that entered extraction filtering and quality assessment the cutoff score higher than 0.5 was set with the possibility of assign half point for QA1 (see Table 3.6) in case the study shows challenges or pros and cons only. However, as explained in section 3.1, some of the selected studies were also Systematic Literature Reviews, so to avoid bias in the results, we unraveled the systematic literature reviews presented on table 3.8 and analyzed all primary studies within those SLR(s).

It is important to highlight that for this analysis we maintained the same protocol, with the same Inclusion and Exclusion criteria (3.4) therefore, studies either published before 2010 or presenting proprietary approaches or providing only statistics about a elicitation technique or duplicated studies were discarded from analysis. During the analysis some of the studies presented on those SLRs were also captured either in the automatic search or the manual search, therefore they were also not included to avoid any duplicated analysis and counting. Also, for reference, from this thorough analysis, 14 other studies emerged as selected. These were included in the numbers when describing the protocol.

After removing the duplicated ones, the analysis continued by the researcher reading and verifying the title and abstract of the papers. At this step the major direction was to remove the ones that were not related to agile challenges, elicitation techniques or any close-related description. The analysis was done by using the inclusion and exclusion criteria described in Table 3.4. It is important to highlight that in case of any doubt or mind-breaker about the paper the main direction was to promote to the next phase of the analysis to avoid discard any paper that could be relevant to the research.

At this point another 291 papers were excluded from the researching, with exactly ninety eight (98) studies remaining to full analysis. As one can see in Figure 3.2 the papers were published between January, 2010 and October, 2022 with most of the accepted papers being from 2017, 2019, 2021 and 2022.

In analyzing the scoring of the papers the decision was to score as whole, partially or not score for each quality assessment question according to the brief definition on “Quality extraction cutoff score”, as shown in Table 3.7.

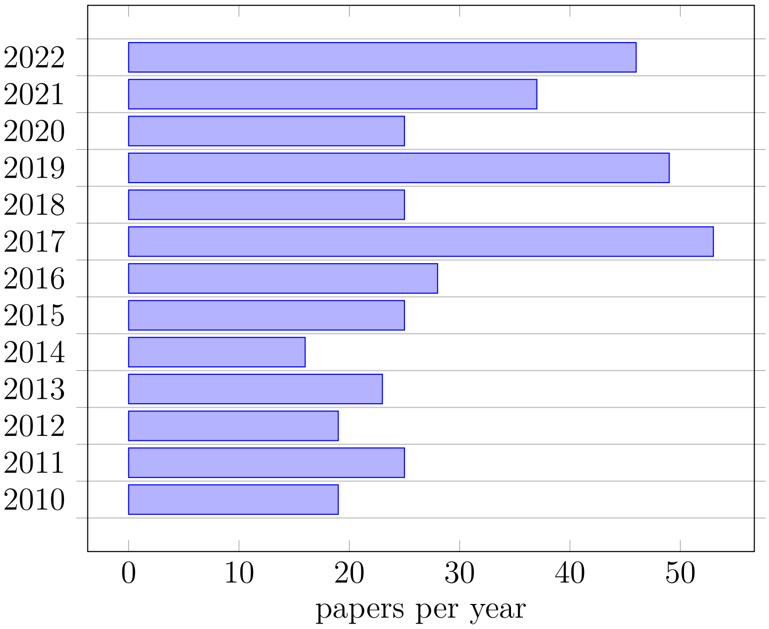


Figure 3.2: Initially accepted papers per year

Score	Definition
Zero (0)	when the study did not mention the information required.
A half (1/2)	when the study did mention or reference the information although did not elaborate on it.
One (1)	when the study mentioned or referenced the information also providing at least a brief explanation or examples.

Table 3.7: Quality extraction cutoff score

The score cutoff for the review was to discard any study with Zero (0) or Half (0.5) points as the outcome was that these papers were not the real scope of this work and would not aggregate the necessary information for this analysis. By the ending of the phase, another 48 papers were rejected. Figure 3.3 presents the each step of this review.

Furthermore, after filtering the studies applying all the defined criteria and rules, the final list of studies that entered extraction filtering and quality assessment the cutoff score higher than 0.5 was set. However, as explained in Section 3.1, some of the selected studies were also Systematic Literature Reviews, so to avoid bias in the results, we unraveled the systematic literature reviews presented in Table 3.8 and analyzed all primary studies within those SLR(s). Hence, the snowballing strategy was further employed to enhance the search process.

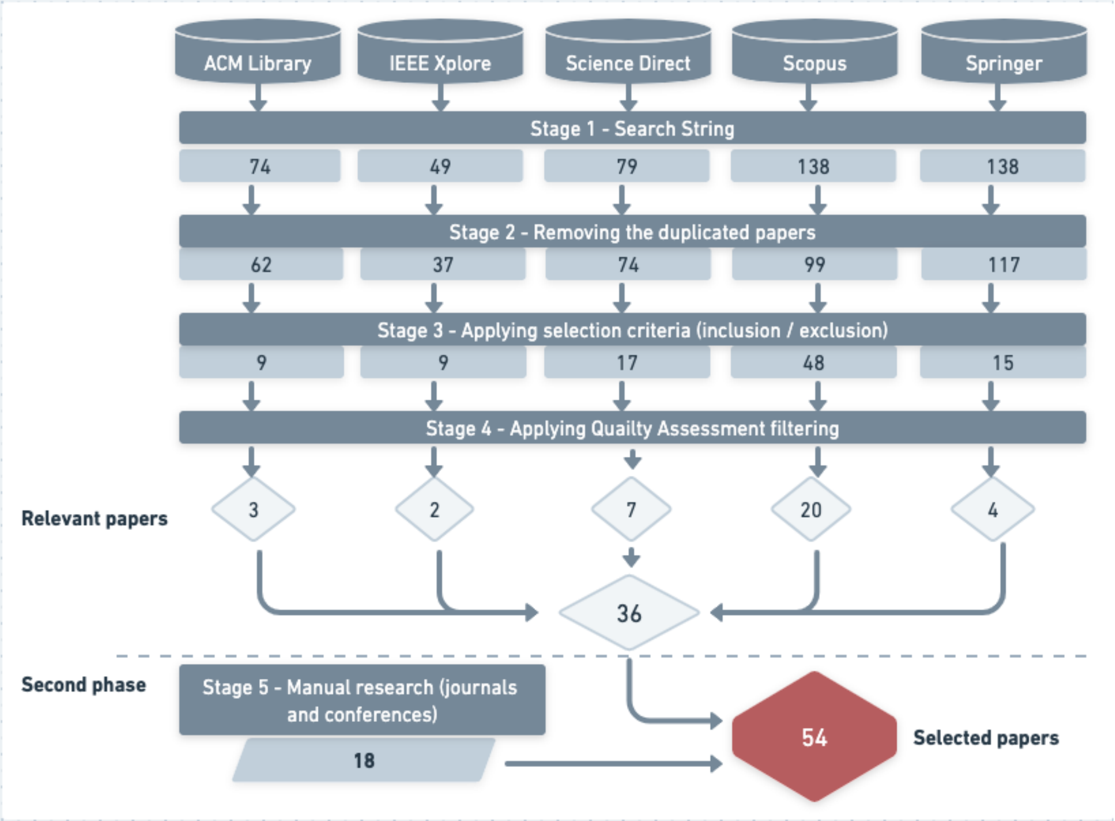

ACM Library
IEEE Xplore
Science Direct
Scopus
Springer
Stage 1 - Search String
74
49
79
138
138
Stage 2 - Removing the duplicated papers
62
37
74
99
117
Stage 3 - Applying selection criteria (inclusion / exclusion)
9
9
17
48
15
Stage 4 - Applying Quailty Assessment filtering
Relevant papers
3
2
7
20
4
36
Second phase
Stage 5 - Manual research (journals and conferences)
18
54
Selected papers

Figure 3.3: SLR conduction evolution

Paper title	Reference
A Qualitative Study on Non-Functional Requirements in Agile Software Development	[150]
Agile Requirements Engineering: A systematic literature review	[167]
Quality requirements challenges in the context of large-scale distributed agile: An empirical study	[160]
Requirements engineering: A systematic mapping study in agile software development	[145]
A Requirements Engineering Techniques Review in Agile Software Development Methods	[159]
A Systematic Literature Review on implementing non-functional requirements in Agile Software Development: issues and facilitating practices	[168]
Requirements elicitation techniques: a systematic literature review based on the maturity of the techniques	[39]

Table 3.8: Snowballed studies (SLRs)

For this analysis we maintained the same protocol, with the same Inclusion and Exclusion criteria (3.4) therefore, studies either published before 2010 or presenting proprietary approaches or providing only statistics about a elicitation technique or duplicated studies were discarded from analysis. During the analysis some of the studies presented on those SLRs were also captured either in the automatic search or the manual search, therefore they were also not included to avoid any duplicated analysis and counting. From this thorough analysis, 14 emerged as selected.

Finally, the final list of papers that were taken into account for the Systematic Literature Research (SLR), can be seen in Table "Selected studies" (3.9).

Id	Paper title	Reference
[S1]	Information Extraction on Requirement Prioritization Approaches in Agile Software Development Processes	[136]
[S2]	A Qualitative Study on Non-Functional Requirements in Agile Software Development	[150]
[S3]	A proposed framework for improved software requirements elicitation process in SCRUM: Implementation by a real-life Norway-based IT project	[109]
[S4]	Requirements specification for developers in agile projects: Evaluation by two industrial case studies	[169]
[S5]	Robust approaches, techniques and tools for requirement engineering in agile development	[126]
[S6]	The role of ethnography in agile requirements analysis	[146]
[S7]	Infusing Design Thinking into a Software Engineering Capstone Course	[170]
[S8]	Modelling agile requirements using context-based persona stories	[171]
[S9]	Agile Requirements Engineering: A systematic literature review	[167]
[S10]	HCI usability techniques in agile development	[172]
[S11]	Impacts of agile requirements documentation debt on software projects: A retrospective study	[173]
[S12]	Investigating the Link between User Stories and Documentation Debt on Software Projects	[174]
[S13]	Integration of agile practices: An approach to improve the quality of software specifications	[155]
[S14]	Generating feature model from creative requirements using model driven design	[175]
[S15]	Handling requirements using FlexREQ model	[176]

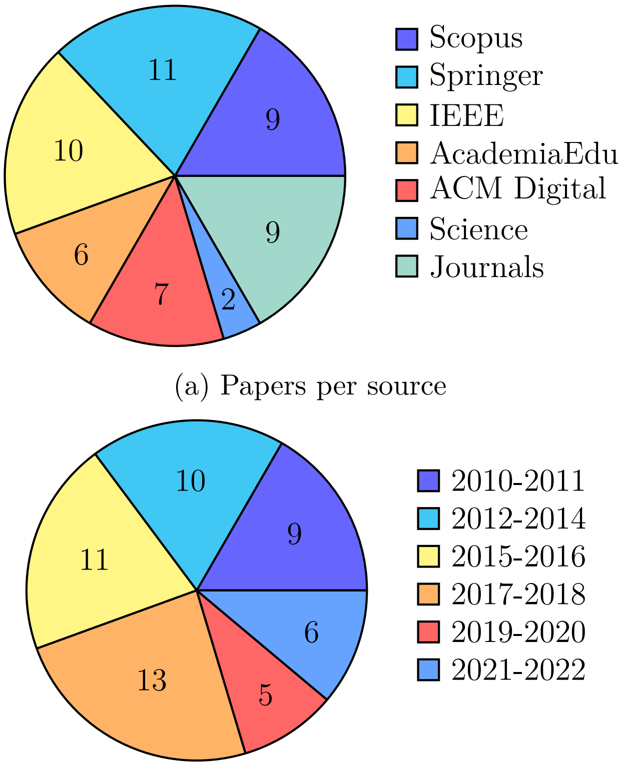
Id	Paper title	Reference
[S16]	Communication patterns of agile requirements engineering	[177]
[S17]	Automated acceptance test refactoring	[178]
[S18]	An approach to requirements elicitation and analysis using goal	[179]
[S19]	Quality requirements challenges in the context of large-scale distributed agile: An empirical study	[160]
[S20]	Quality of software requirements specification in agile projects: A cross-case analysis of six companies	[180]
[S21]	The challenges that challenge: Engaging with agile practitioners' concerns	[181]
[S22]	Requirements engineering: A systematic mapping study in agile software development	[145]
[S23]	Understanding information needs of agile teams to improve requirements communication	[182]
[S24]	Mind-mapping: An effective technique to facilitate requirements engineering in agile software development	[183]
[S25]	A Model of Software Prototyping Based on a Systematic Map	[184]
[S26]	User and System Stories: An Agile Approach for Managing Requirements in AOSE	[185]
[S27]	A Requirements Engineering Techniques Review in Agile Software Development Methods	[159]
[S28]	Key Challenges in Agile Requirements Engineering	[158]
[S29]	A Systematic Literature Review on implementing non-functional requirements in Agile Software Development: issues and facilitating practices	[168]
[S30]	Procedural Model of Requirements Elicitation Techniques	[60]
[S31]	Requirements elicitation techniques: a systematic literature review based on the maturity of the techniques	[39]
[S32]	Requirements Elicitation: Towards the Unknown Unknowns	[101]
[S33]	Qualitative comparisons of elicitation techniques in requirement engineering	[65]
[S34]	Comparison of Various Requirements Elicitation Techniques	[32]
[S35]	Software Development Lifecycle Models	[127]
[S36]	Effective Requirements Development–A Comparison of Requirements Elicitation Techniques	[67]
[S37]	Requirement Elicitation Technique: - A Review Paper	[38]

Id	Paper title	Reference
[S38]	Revisiting Requirements Elicitation Techniques	[37]
[S39]	(chapter 5) Requirements Elicitation	[122]
[S40]	Systematic Review of Requirement Elicitation Techniques	[36]
[S41]	Ambiguity in user stories: A systematic literature review	[186]
[S42]	An industry experience report on managing product quality requirements in a large organization	[187]
[S43]	Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation	[188]
[S44]	A case study on benefits and side-effects of agile practices in large-scale requirements engineering	[189]
[S45]	Agile requirements engineering with prototyping: A case study	[190]
[S46]	Non-functional requirements elicitation guideline for agile methods	[191]
[S47]	Gamified requirements engineering: model and experimentation	[192]
[S48]	Identification of requirements using goal oriented requirements elicitation process	[193]
[S49]	Understanding the use of elicitation approaches for effective requirements gathering	[194]
[S50]	Extending application of non-verbal communication to effective requirement elicitation	[195]
[S51]	Distributed Elicitation of Software Requirements: an experimental case from Argentina and Colombia	[196]
[S52]	Requirements elicitation with web-based focus groups	[197]
[S53]	A new approach to requirements elicitation using paper prototype	[198]
[S54]	A hybrid approach of requirement engineering in agile software development	[199]

Table 3.9: Selected primary studies

3.3 SLR Results

Interest remains in this topic of elicitation techniques and its usage in either traditional models or agile models. From the initial 496 papers retrieved either by using the research string or by manual investigation, a total of 54 studies were considered after the final analysis. Figure 3.4 "Selected papers by source" illustrates a view of the final list of the accepted papers, by showing the numbers by source and by year of the study had been published.



(a) Papers per source

(b) papers per year

Figure 3.4: Selected papers by source and year.

3.3.1 RQ.1. What are the existing techniques in the literature to elicit software requirements that can be used in agile processes?

From the total amount of retrieved papers we have selected 54 studies for the period 2010–2022 on the requirements elicitation process topic. Our findings reveal that the studies found in the literature have diverse depths with some of these studies simply categorizing techniques according to their approach, others identifying which techniques are more useful in specific situations. On the other hand, some are more thorough and highlights the characteristics of one or more requirements elicitation technique and promote their advantages or elects their disadvantages.

After analysis, we noticed that within the studies extracted either via the search string or by manual research in conferences and journals, there are two clear focuses for the papers. Part of the selected papers aims to demonstrate and discuss challenges of

Requirements Engineering and Agile Software Development while another set of the retrieved papers focuses on the presentation, discussion and study of requirements elicitation techniques. Both study profiles are valid for this work.

Focusing in answering RQ.1, we first separated the studies that aimed to list requirements elicitation techniques from those that discussed in depth one or more RE techniques. For this RQ, we have used studies presented in the Table 3.10. Within these studies, we identified several techniques. Note that the majority of studies, regardless of other discussions, also address Interview and Prototyping techniques.

In all cases, as an answer to RQ.1, we have the list in Table 3.10, containing the techniques presented in the literature, for the selected studies.

Id	Reference	Elicitation techniques used.
[S1]	[136]	Quality Function Deployment (QFD); Interview; Questionnaires; Survey; Prototyping; and User Stories.
[S2]	[150]	User stories; Story Cards; Interview; Observations; Workshop; Prototyping; and Brainstorming.
[S3]	[109]	Brainstorming; Prototyping; and Mind mapping.
[S4]	[169]	User Stories; Story Cards; and Scenarios.
[S5]	[126]	Joint Application Development (JAD); Mind Mapping; and Story Cards.
[S6]	[146]	Feature-Driven Design (FDD); Questionnaires; Interview; Surveys; Workshop; Data and Document Analysis; Observation; and Ethnography.
[S7]	[170]	Prototyping; Persona; Storyboards; and Journey Maps.
[S8]	[171]	Persona; User Stories; and Storyboards.
[S9]	[167]	User Stories; Mind mapping; Persona; Scenarios; Prototyping; Participatory Design; Qualitative/ Quantitative Customer-driven Development (QCD); and Human-Centered Design (HCD).
[S10]	[172]	Persona; Scenarios; Storyboards; Prototyping; Ethnography; and Observation.
[S11]	[173]	User Stories.
[S14]	[175]	Mind Mapping.
[S15]	[176]	Interview.
[S16]	[177]	Ethnography; User Stories; and Story Cards.
[S17]	[178]	Test-Driven Design (TDD); and Behavior-Driven Design (BDD).
[S18]	[179]	Goal based analysis; and Scenarios.

Id	Reference	Elicitation techniques used.
[S20]	[180]	Feature-Driven Design (FDD); Interview; Observation; Data and Document Analysis; and User Story.
[S22]	[145]	Joint Application Development (JAD); Prototyping; Mind mapping; and User Stories.
[S23]	[182]	Persona; Prototyping; Goal model and User Stories.
[S24]	[183]	Mind Mapping.
[S25]	[184]	Prototyping.
[S26]	[185]	Interview; Questionnaires; Workshop; and User Story.
[S27]	[159]	Interview; Focus Groups; Brainstorming; Questionnaires; Prototyping; Observation; and Document or Data Analysis.
[S30]	[60]	Interview; Questionnaires; Focus Group; Joint Application Design (JAD); Prototyping; Document analysis; Card Sorting; Laddering; Observation; Social Analysis and Brainstorming.
[S31]	[39]	Interview; workshop; Focus Groups; Joint Application Design (JAD); Quality Function Deployment (QFD); Ethnography; Scenarios; Prototyping; Protocol Analysis; Card sorting; Ontology; Modeling; Goal-based approach; Repertory Grids; User Story; Mind Mapping; and Storytelling.
[S32]	[101]	Use Cases; Interview; Observation; Workshop; Scenario; Joint Application Design (JAD); Prototyping; Laddering; Ethnography; Participatory Design; and Analytical Hierarchy Process (AHP).
[S33]	[65]	Brainstorming; Workshop; Prototyping; Joint Application Design (JAD); Group work; Ethnography; User Scenarios; and Introspection.
[S34]	[32]	Interview; Document Analysis; Questionnaires; Observation; Ethnography; Prototyping; Joint Application Design (JAD); Brainstorming; Group work; Workshop; User Scenarios; and Laddering.
[S35]	[127]	Joint Application Design (JAD)
[S36]	[67]	Interview; Workshop; Focus group; Brainstorming; Protocol Analysis; Ethnography; Observation; Laddering; Scenario; Joint Application Design (JAD); Prototyping; and Storyboards.
[S37]	[38]	Prototyping; Brainstorming; Questionnaires; Laddering; Ethnography; and Interview.

Id	Reference	Elicitation techniques used.
[S38]	[37]	Interview; Document Analysis; Questionnaires; Prototyping; Brainstorming; Joint Application Design (JAD); Ethnography; Protocol Analysis; and Laddering.
[S39]	[122]	Interview; Questionnaires; Introspection; Ethnography; Brainstorming; Prototyping; Legacy system analysis; Scenario; Goal Modeling; and Persona.
[S40]	[36]	Prototyping; Document Analysis; Observation; Questionnaires; and Interview.
[S41]	[186]	User Stories.
[S45]	[190]	Prototyping.
[S46]	[191]	Interview; Observation; Social Analysis; Focus Groups; Brainstorming; Prototyping; Joint Application Development (JAD).
[S47]	[192]	User Stories.
[S48]	[193]	Interview; Data and Document Analysis; Surveys; Card Sorting; Laddering; Focus group; Brainstorming; Joint Application Development (JAD); Prototyping; Participatory Design; Ethnography; Observation.
[S50]	[195]	Interview.
[S51]	[196]	Interview; Questionnaires; Brainstorming.
[S52]	[197]	Ethnography; Interview; Joint Application Development (JAD); Scenarios.
[S53]	[198]	Interview; Questionnaires; Workshop; Brainstorming; Storyboards, Use Cases; Role Playing; Prototyping.
[S54]	[199]	Joint Application Development (JAD).

Table 3.10: Techniques used in primary studies

From the list of papers Table 3.10 we analyzed two interesting scenarios. First the number of studies that reference, cite or discuss any requirements elicitation technique. The word cloud 3.5 gives a graphic perspective of the most referred techniques found throughout these papers. The word cloud shows Prototyping, Interview and User Stories as the three most referenced techniques as well as Observation, Scenarios and Questionnaires being well ranked. However, it is interesting to note that JAD, Mind mapping and, especially Ethnography are among the most referenced techniques referred in literature, as presented in the list 3.3.1, although, at least in the community of respondents of this work, these RE techniques are the ones with the least percentage of knowledge as one can compare by seeing in the Figure 3.14.

Design Thinking, which is much viewed as a methodology [200], a process, an action or a protocol for solving problems and discovering new opportunities [201], [202], [203], [204], [205], [206] were referred in a few papers, however in this study we are working with techniques and not with complete methodology or process therefore, although Design Thinking is being mentioned for comparison effects, it is not being taken into account for strengths or weakness perspective as well as for suggestions of techniques combinations.

Use Cases, on its side, is also listed in some papers [136], [126], [167], [179], [180], [39], [101] nevertheless this technique are more related to specification than the requirements elicitation itself, moreover the decision was to not include in the techniques analysis and combination as well.

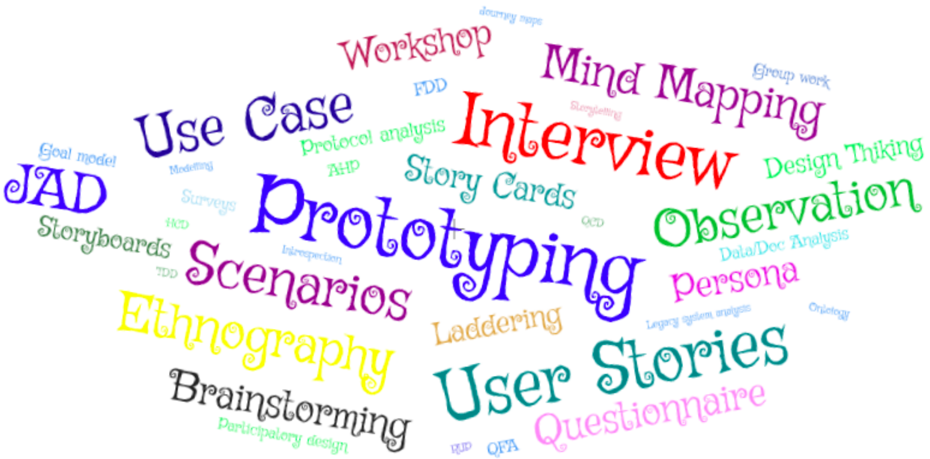
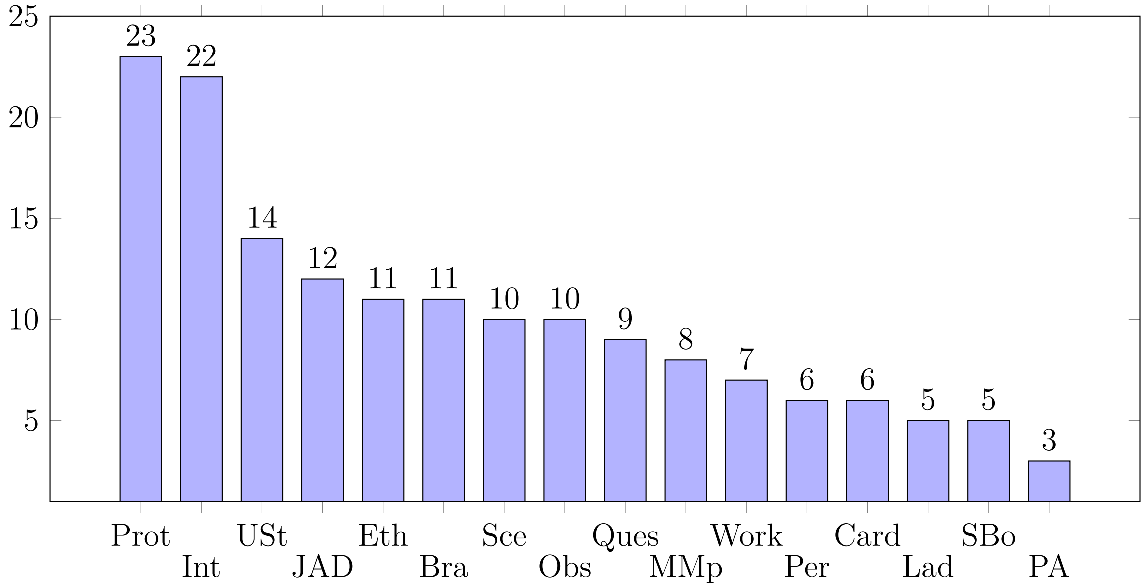


Figure 3.5: Word cloud for elicitation techniques

According to the selected papers, the list of the existing requirements elicitation techniques in literature are as follows:

- Most cited techniques (number of referencing studies)
  - Prototyping (23)
  - Interview (22)
  - User Stories (14)
  - JAD (12)
  - Ethnography (11)
  - Brainstorming (11)
  - Scenarios (10)
  - Observation (10)
  - Questionnaires (9)
  - Mind Mapping (8)
- Least cited techniques (all with only 1 study citation)

- Analysis of Legacy Systems
- Human-Centered Design
- Introspection
- Journey Maps
- Ontology
- Role Playing
- Social Analysis
- Storytelling
- Qualitative/Quantitative Customer-driven Development (QCD)
- Test-Driven Design (TDD)

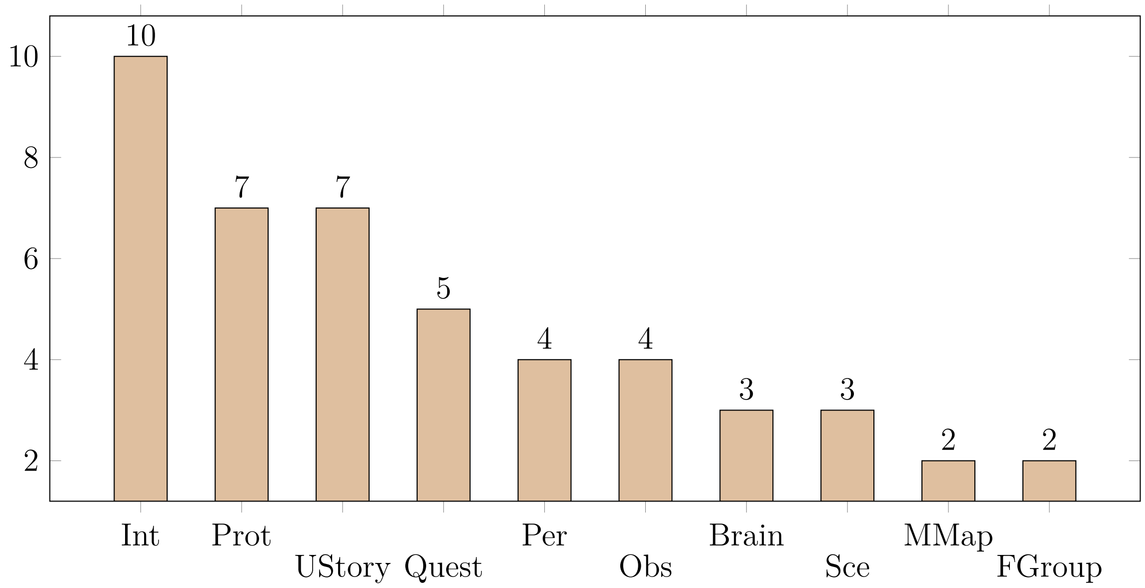


Pro=Prototyping; Int=Interview; USt=User Stories; JAD=Joint Application Development; Eth=Ethnography; Bra=Brainstorming; Sce=Scenarios; Obs=Observation; Que=Questionnaires; MMp=Mind mapping; Work=Workshop; Per=Persona; Card=Story Cards; Lad=Laddering; SBo=Story Boarding; PA=Protocol Analysis.

Figure 3.6: RE Techniques referred in literature

The second scenario was to analyze the requirements elicitation techniques that were actually being used in a project, process, framework or case study, or, in other words being used in industry or real word project.

For this scenario the central idea was to consider the technique as being referenced for industry when the paper was describing the use of the technique in a project or case study conducted in the real world within any branch of industry or services. Specific cases, such as the use of techniques such as Interview or Prototyping only as part of the theoretical study, were not considered as equivalent to studies related to industry. The findings are present in the Figure 3.7.

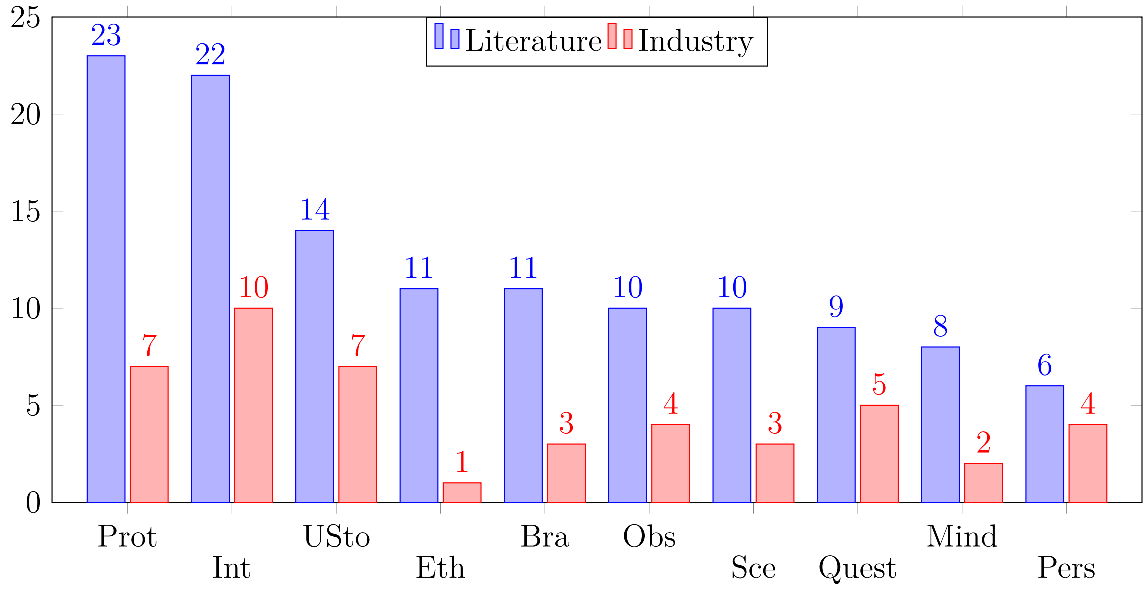


Int=Interview; USt=User Stories; Pro=Prototyping; Que=Questionnaires; Per=Persona; Obs=Observation; Bra=Brainstorming; MMp=Mind mapping; Sce=Scenarios; FGr=Focus Group.

Figure 3.7: Elicitation techniques used in industry

These numbers were obtained during the SLR by reviewing the studies that discussed or described projects taken in the real world or practical cases of study . In reference to the use of techniques in industrial projects or in practical cases, according to the documents obtained in the SLR, we have that in descending order, Interviews [180]; [181], [60], [185], [190], [191], [194], [195], [196], [182] is the most referenced, followed by Prototyping [170], [172], [184], [60], [190], [191], [195]; User Stories [173]; [174]; [180], [60], [190], [191], [192]; Questionnaires [60] [192], [196], [198], [182]; Persona [170], [172], [192], [182]; Observation [180]; [181], [60], [191]; Scenarios [60], [172], [182]; Brainstorming [190], [191], [196]; Mind Mapping [175], [183] and Focus Group [197].

Furthermore, it is interesting to compare the numbers for the most cited techniques on the literature side against the ones cited as part of a practical case in the industry. In summary, the most cited techniques in both cases are the same, with Interviews, Prototyping and User Stories having the highest numbers of records. On the other hand, we see elicitation techniques such as Persona, more recent, and still with a lot of field for study in the literature, but which are already being well used in practical applications in the industry. In addition, it is noteworthy that the Ethnography technique is recurrent in literature studies, however, at least in this set of selected papers, it was not remembered during practical experiences. The Figure 3.8 demonstrates it for the ten most referred techniques in literature comparing with the citations in industry.



Pro=Prototyping; Int=Interview; USt=User Stories; Eth=Ethnography; Bra=Brainstorming; Obs=Observation; Sce=Scenarios; Que=Questionnaires; MMp=Mind mapping; Per=Persona.

Figure 3.8: Comparing techniques described in literature and used in industry

As one can see some techniques are much referred in literature however there are differences when dealing with real projects or case studies.

RQ.1. Summary: The most referenced techniques in the literature are Prototyping, Interviews, User Stories, Joint Application Development (JAD), Ethnography and Brainstorming, while the most cited for industry are Interviews, User Stories, Prototyping, Questionnaires, Persona and Observation. Results show the first three techniques consolidated in both worlds, with the industry also working with techniques that can cover a larger number of users such as Questionnaires or simplify user profiles as Persona technique can do.

3.3.2 RQ.2. What are the strengths and weaknesses (pros and cons) in the identified techniques, as well as the challenges reported in the literature for the requirements elicitation phase?

Regarding the RQ.2, we started analyzing the reported challenges covered by studies that focused on this perspective (analyzing challenges). In general, including the studies extracted by the search string and the ones collected by using manual research in journals and conferences, we have found fourteen (14) papers seeking to discuss the challenges of ASD and Requirements Engineering. Table 3.11 summarizes the main findings related to challenges. Aiming for a more complete view, we opted to initially detail all referred challenges without remove the ones not specific related to elicitation. Later, during the categorization, we justify any exclusion or classification. Additionally, the table 3.10 presented during the RQ.1 3.3.1 summarized the studies that brought information, including pros and cons of the Requirements Elicitation Techniques.

Id	Ref.	Referred challenge
[S2]	[150]	i) insufficiency of the user story format; ii) reliance on tacit requirements knowledge given by users.
[S3]	[109]	i) lack of precise requirements can cause system failure; ii) scalability limitations towards large-scale projects in agile; iii) less documentation and quick processing can lead to skipping requirements; iv) clients that have less knowledge of necessary requirements; v) clashes among software experts and stakeholders; vi) requirement prioritization; vii) communication and coordination with all teams for requirements elicitation.
[S9]	[167]	i) lack of allocated time for upfront activities; ii) difficulty of modularization of the software; iii) lack of documentation.
[S11]	[173]	i) lack of information before jumping into development; ii) difficulty to prioritize Requirements due to high abstraction level; iii) lack of non-functional requirements identification; iv) volatility of requirements; v) definition of requirements with a low level of detail; vi) difficulty to define dependencies between requirements; vii) difficulty to predict impacts of changes; viii) problems with the communication and collaboration with users.
[S12]	[174]	i) lack of documentation; ii) requirements prioritization; iii) lack of identification of non-functional requirements; iv) communication and collaboration with users; v) incomplete design specification.

Id	Ref.	Referred challenge

[S13]	[155]	i) customer unavailability;
		ii) turnover of the team;
		iii) poor documentation;
		iv) communication on remotely distributed teams and clients.

[S15]	[176]	i) incomplete requirements;
		ii) incomplete domain knowledge;
		iii) overlooking tacit assumptions;
		iv) definition of system boundaries;
		v) ambiguous requirements and inconsistent requirements;
		vi) different views of users;
		vii) too many "primary" users;
		viii) requirements variability.

[S19]	[160]	i) minimal documentation and difficult to document the requirement ii) inappropriate architecture when new requirements arise iii) customer not available iv) difficulties to estimate and plan efforts v) requirements prioritization vi) difficulties to discover dependencies between sub-systems in early stages vii) neglect of quality requirements viii) lack of customer's domain knowledge ix) team coordination and communication; x) uneven teams maturity; xi) difficulties to promote innovative ideas xii) hidden assumption.

Id	Ref.	Referred challenge
[S21]	[181]	i) misconceptions and interpretations ii) sparse or limited information iii) organizational culture issues iv) lack of commitment or engagement from stakeholders v) distributed teams vi) lack of training in specific techniques or practices vii) knowledge sharing viii) large and complex projects ix) poor documentation x) standard contracts that require upfront specifications xi) poor measurement to control project evolution; **Obs.** Categories of challenges:organization and management; people; process; and tools related challenges.
[S23]	[182]	i) insufficient requirements communication with customer(s); ii) insufficient requirements communication in teams; iii) lapses and rework due to low quality of documented requirements; iv) lack of communication of relevant changes to relevant people; v) substantial changes of initial estimates of time and cost; vi) rework in the architecture design; vii) delay in completing the assigned work on time; viii) system security, usability or performance is at risk; ix) neglecting of non-functional requirements; x) communication lapse due to sudden changes in the requirements.
[S27]	[159]	i) inadequate understanding of the end user's needs; ii) inability to meet changing requirements; iii) modules that can not be coupled to work together; iv) software difficult to maintain or expand; v) late detection of critical faults; vi) software of low quality; vii) unacceptable software performance.

Id	Ref.	Referred challenge
[S28]	[158]	i) functional or technical dependencies with other teams; ii) independent (detailed) decisions from development team regardless stakeholders; iii) understand the big picture for complex requirements; iv) requirements volatility; v) work out user requirements and quality of use in cooperation with end users; vi) involve stakeholders throughout the whole development process in regular iterations.
[S29]	[168]	i) neglecting non-functional requirements (NFRs); ii) misunderstandings regarding user stories; iii) lack of traceability mechanisms of NFRs iv) lack of cost-effective real integration test; v) ambiguous communication process; vi) misunderstanding the architecture drivers between teams; vii) unmanaged architecture changes; viii) hidden assumptions regarding NFRs implementation in inter-team collaboration; ix) moving to Agile with a waterfall mind-set; x) sporadic adherence to quality guidelines by Agile teams; xi) overlooking sources (stakeholders); xii) insufficient knowledge or competencies in the project team.
[S30]	[60]	i) lack of participation of user; ii) lack of experience of the analyst; iii) lack of understanding of elicitation techniques.
[S42]	[187]	(i) difficulties in communication of requirements; (ii) large projects; (iii) reuse of the knowledge from one project to another.
[S43]	[188]	(i) difficulties with customer involvement; (ii) difficulties in translating requirements.

Id	Ref.	Referred challenge

[S44]	[189]	(i) communication gaps;
		(ii) overscopping;
		(iii) waste requirements;
		(iv) low quality on requirements;
		(v) customer expectation not met;
		(vi) low motivation for requirements work;
		(vii) weak requirements prioritization;
		(viii) lack of documented work.

[S45]	[190]	(i) lack of documentation; (ii) motivation issues related to RE work; (iii) hard to achieve sufficient customer presence in project; (iv) difficulties in reaching consensus among more than one customer group; (v) ensuring sufficient customer competence or knowledge; (vi) neglect of non-functional requirements; (vii) constant re-prioritization.

[S49]	[194]	(i) incomplete requirements;
		(ii) incomplete domain knowledge;
		(iii) ambiguous or inconsistent requirements;
		(iv) different views of different users;
		(v) excessive requirements;
		(vi) incomplete understanding or translating of needs,
		(vii) poor user collaboration;
		(viii) continuous changing or addition of requirements.

[S53]	[198]	(i) scope;
		(ii) volatility;
		(ii) lack of understanding.

Table 3.11: Studies related to challenges

Based on the challenges found in the selected studies within the literature we analyzed and concluded that most of them could be organized into major categories related to the item that they were linked to. Moreover, in order to categorize the challenges we first analyzed each of the reported challenges and grouped them based on the text.

First, we categorized based on similarity or proximity, as for instance, anything related to documentation issues were placed in the "Documentation (Doc)" category or challenges related to knowledge sharing or communication among teams or other stakeholders were added to “Communication (Comm)”. Any challenge that did not fit into any defined category was placed in the “Others (Oth)”.

Therefore, we have created categories related to documentation, prioritization, quality, communication, experience or expertise, volatility, availability, scalability, domain knowledge, culture, requirements translation, as well as, the ones related to different items which were classified as other.

In general, the categories are self-explained by the name, however it is important to summarize all the categories. So, Table 3.12 presents the description of each category.

Category	Description
Documentation	problems related to lack, incomplete or low quality of the necessary documentation.
Communication	problems related to the flow (send, receive or share) of the information.
Translation of requirements	interpretation and understanding of the requirements.
Stakeholder Engagement and Availability	difficulties related availability, engagement, collaboration of users and customers.
Scope and Volatility of the requirements	issues on defining a stable set of requirements and project boundaries.
Complexity and Scalability of the projects	issues related complexity of projects.
Negotiation and Prioritization of Requirements	negotiation on requirements priorities among stakeholders.
Expertise from team or user	problems related to knowledge or experience in the domain of the software or with the methods being used.
Architecture and SW Quality	difficulties related to design.
Culture and Environment	related to organization and teams structure and method of work.
Requirements Elicitation work	problems related to non-functional requirements or handling of RE task itself.
Project issues and Others	problems related to project control or team decisions.

Table 3.12: Description of categories

Furthermore, the list of challenges collected on each of the selected papers and the defined categories with their respective name and initials can be seen in the Table 3.13.

Also, the the most cited challenges retrieved from the studies are related to poor or lack of documentation [182], [150], [109], [189], [194], [190]; miscommunication [182], [160], [187], [174], [173], [109], [189], [150]; difficulties in define or interpret the requirements [109], [194], [173], [176], [189], [159], [188], [194] and customer availability [60], [188], [158], [150], [194], [190], [194].

Documentation (Doc)

i) skipping requisites due to less paperwork and fast processing;

ii) lack or poor of documentation;

iii) insufficiency of the user story format;

iv) rework due to low quality of documented requirements;

v) work not documented;

vi) incomplete or poor documentation.

Stakeholder Engagement and Availability (Engg)

i) lack of participation of user or customer not available;

ii) lack of commitment or engagement from stakeholders;

iii) work out user requirements and quality of use in cooperation with end users;

iv) involve stakeholders throughout the whole development process in regular iterations;

v) overlooking sources (stakeholders);

vi) poor user collaboration;

vii) hard to achieve sufficient customer presence in project.

Communication (Comm)

i) ambiguous communication process;

ii) insufficient requirements communication with customer(s);

iii) insufficient requirements communication in teams;

iv) poor communication of relevant changes to relevant people;

v) communication lapse due to requirements sudden changes;

vi) communication on remotely distributed teams and clients;

vii) communication and coordination with all teams for requirements elicitation;

viii) team coordination and communication;

ix) problems with communication and collaboration with users;

x) knowledge sharing;

xi) sparse or limited information;

xii) communication gaps;

xiii) incomplete or poor documentation.

Scope and Volatility of the Requirements (Vol)

i) requirements variability;

ii) requirements volatility;

iii) overscopping;

(iv) waste requirements;

(v) excessive requirements;

(vi) neglect of non-functional requirements;

(vii) customer expectation not met;

(viii) continuous changing or addition of requirements;

(ix) scope.

Complexity and Scalability of the projects (Cplx)

i) scalability limitations towards large-scale projects in agile;

ii) large and complex projects.

Negotiation and Prioritization of Requirements (Prio)

i) hardship to prioritize requirements due to high abstraction;

ii) requirement prioritization;

iii) clashes among software experts and stakeholders;

iv) difficulties in reaching consensus among more than one customer group;

v) different views of users or too many “primary” users;

vi) weak requirements prioritization;

vii) constant re-prioritization.

Translation of requirements (Trl)

i) incomplete requirements;

ii) lack of information before jumping into development;

iii) understand the big picture for complex requirements;

iv) dependencies discovery between sub-systems in early stages;

v) inadequate understanding of the end user’s needs;

vi) inability to meet changing requirements;

vii) misconceptions and interpretations;

viii) ambiguous requirements and inconsistent requirements;

ix) lack of precise requirements can cause system failure;

x) definition of system boundaries;

xi) definition of Requirements with a low level of detail;

xii) difficulty to define dependencies between requirements;

xiii) misunderstandings regarding user’s description;

xiv) standard contracts that require upfront specifications;

xv) difficulties in translating requirements;

xvi) incomplete understanding or translating of needs;

xvii) low quality on requirements.

Expertise from team or user (Exp)

i) insufficient knowledge or competencies in the project team;

ii) lack of experience of the analyst;

iii) lack of understanding of elicitation techniques;

iv) lack of customer's domain knowledge;

v) moving to Agile with a waterfall mind-set;

vi) lack of training in specific techniques or practices;

vii) clients have less knowledge of necessary requirements;

viii) incomplete domain knowledge.

Architecture and Software Quality (Qlty)

i) software difficult to maintain or expand;

ii) late detection of critical faults;

iii) software of low quality;

iv) unacceptable software performance;

v) misunderstanding the architecture drivers between teams;

vi) unmanaged architecture changes;

vii) incomplete design specification;

viii) modules that can not be coupled to work together;

ix) difficulty to predict impacts of changes;

x) difficulty of modularization of the software;

xi) inappropriate architecture when new requirements arise;

xii) rework in the architecture design;

xiii) sporadic adherence to quality guidelines by Agile teams;

xiv) delay in completing the assigned work on time.

Culture and Environment (Cult)

i) organization culture;

ii) distributed teams;

iii) functional or technical dependencies with other teams;

iv) substantial changes of initial estimates of time and cost;

v) difficulties to promote innovative ideas;

vi) uneven teams maturity.

vii) turnover of the team.
Requirements Elicitation work (REw)
i) low motivation for requirements work;
ii) hidden assumption, specially on NFRs implementation;
iii) overlooking tacit assumptions;
iv) reuse of the knowledge from one project to another;
v) lack of allocated time for upfront activities;
vi) lack of identification of Non-functional requirements;
vii) neglecting non-functional requirements (NFRs);
viii) neglect of quality requirements.
Project issues and Others (Oth)
i) difficulties to estimate and plan efforts;
ii) poor measurement to control project evolution;
iii) system security, usability or performance is at risk;
iv) development team autonomy decisions despite stakeholders;
v) Reliance on tacit requirements knowledge given by users.

Table 3.13: Challenges by category

After having a list of the challenges listed by the selected papers and having organized the list in major categories, we also analyzed each of the categories were indeed related to requirement elicitation tasks. Although relevant to the development cycle, the categories "Culture and Environment (Cult)", "Architecture and Software Quality (Qlty)" and "Project issues and Others (Oth)" are not related to Requirements Elicitation.

In simple terms, there is nothing that can be done during the RE phase to, for instance, overcome problems related to high level of turnover in the company or bad architecture decisions or implementation during design and development. As this work is focused in Requirements Elicitation the categories "Cult", "Qlty" and "Oth" were removed from further analysis.

In addition, by analyzing each of the discussed techniques searching for its documented strengths and weakness either in the selected papers or in other academic studies it was possible to classify which RE technique would be able to assist on one or more category challenge. In summary, if there is one or more papers providing information that an specific requirements elicitation technique assists or is useful in facilitating "translating requirements" or in improving "communication" or facilitating "engagement with users or stakeholders”, then the referred technique was considered to be useful in assisting in these categories.

After the analysis of the identifying the strengths (pros), we classified the techniques that could assist for each of the challenge categories by organizing the Table 3.14(colors on the table are only for visual purposes and it does not imply anything). The rationale behind the process of targeting the techniques to the categories of challenges they could assist with was based on the information collected on strengths and weaknesses of these techniques that can be seen in Chapter 4.

An example is when a technique has as its positive point or strength the ease of engaging customers and users, either because it is simple to understand, or because it is a quick process, in which there would not be much resistance from the parties interested in taking part. In this case, this technique, with a strong characteristic of engaging users or those responsible for requirements elicitation, was included in the challenge category of "Stakeholder engagement and Availability". The classification in relation to the categories, therefore, took into account only and specifically the strengths and weaknesses analyzed for each technique discussed in this work.

Category	Detailed Challenge	Technique Assistance
Doc	Documentation	Mind Mapping QFD Observation Ethnography
Trl	Translation of requirements	Interview Storyboards Laddering
Comm	Communication	Prototyping Persona User Stories Questionnaires Brainstorming Focus group JAD QFD Scenarios Workshop
Exp	Expertise from team or user	User Stories Interview Mind Mapping

Category	Detailed Challenge	Technique Assistance
Exp		Laddering Focus group Legacy systems analysis
Prio	Negotiation and Prioritization of Requirements	Questionnaires Persona User Stories FDD Laddering QFD TDD
Cplx	Complexity and Scalability of the projects	Mind mapping JAD TDD Workshop
Engg	Stakeholder Engagement and Availability	User Stories Questionnaires Prototyping Laddering Scenarios Stakeholder analysis Workshop
Vol	Scope and Volatility of the Requirements	Prototyping Interview Focus group JAD TDD
REw	Requirements Elicitation work	Brainstorming Laddering Mind Mapping Persona QFD Storyboards

Table 3.14: Techniques that could assist for each challenge

Regarding the pros and cons, or, in other words, strengths and weakness of the techniques identified and discussed in this study, as mentioned at the beginning of this RQ

3.3.2, the Chapter 4 contains more details about each technique, with some information and description of it, as well as the advantages (pros) and disadvantages (cons) identified during this SLR and corroborated with deeper investigation in manual research. Each session of Chapter 4 has some information about the technique and a table containing the pros and cons identified.

For now Table 3.15 presents the papers used on this research to identify the strengths and weaknesses of the analyzed techniques.

Technique under analysis	**References for pros and cons**
Analysis of Legacy Systems	[207], [208]
Brainstorming	[209], [210], [211]
Data and Document analysis	[212], [213], [214]
Ethnography	[209], [101], [215], [216], [217]
Feature-Driven Design (FDD)	[218], [219], [220], [221]
Focus Groups	[209], [222], [223]
Interview	[224], [223], [225], [226]
Joint Application Development (JAD)	[217], [65], [32], [39], [127], [209], [126]
Laddering	[217], [217], [32], [60]
Mind Mapping	[158], [227], [175], [228], [229]
Observation	[209], [217], [101], [164], [101], [32], [169]
Persona	[171], [77], [230], [122]
Prototyping	[32], [170], [109], [184], [209], [65], [65], [209], [167], [159], [39]
Questionnaires	[159], [209], [217], [231], [39], [32]
Quality Function Deployment (QFD)	[136], [232], [233]
Scenarios	[32], [209], [65], [101]
Stakeholder analysis	[234], [235], [236]
Storyboards	[237], [171], [238], [239], [67], [170]
User Stories	[39], [177], [186], [167], [171], [32]
Workshop	[209], [240], [217]

Table 3.15: Studies that references Pros and Cons of RE Techniques

RQ.2. Summary: This research question was discussed in this section and also in Chapter 4 as first the major difficulties of requirements elicitation were analyzed with the most challenging being problems related to Documentation; Communication; Translation of Requirements, Stakeholder Engagement and Availability; Negotiation and Prioritization of Requirements; Scope and Volatility of Requirements; and Complexity and Scalability of the projects. Once challenges were identified, Chapter 4 presents the techniques pros and cons to assist in handling it.

3.3.3 RQ.3. What techniques could be combined to improve the requirements elicitation process?

For the third research question (RQ.3), we did not find strong combination scenarios within the analyzed primary studies. This was expected since it is the major gap from the studies that led us to produce this work.

Before discussing the combinations, it is important to highlight that the intention here is not to create another requirements elicitation techniques guide since there are a few that can be pointed out that provide description and examples for use for several techniques and can be leveraged for this discussion. Our focus is on techniques combinations based on their strengths and weakness.

Although there might be others, Table 3.16 provide guides analyzed during this work that we view as interesting when in need to learn about requirements elicitation techniques.

Guide	URL
DTA4RE	https://sites.google.com/view/dta4re/?pli=1
RE Training	https://retraining.inf.ufsc.br/guia/app/classificacoes/tecnicas-de-elicitacao-de-requisitos
Técnicas de Elicitação de Requisitos	https://sites.google.com/site/tecnicaselicitacao/
Software Testing Help	https://www.softwaretestinghelp.com/requirements-elicitation-techniques/
University of Ottawa	https://www.site.uottawa.ca/ bochmann/SEG3101/Notes/-SEG3101-ch2-3%20-%20ElicitationTechniques.pdf

Table 3.16: Requirements Techniques - Guides that can be useful

That being said, the focus is to discuss and suggest the combinations that could help during Requirements Elicitation (RE) phase. From the selected papers there are a few studies that at least cite the possibility of combination of requirements elicitation techniques.

First study is from Al-Zawahreh et al. [60], who mentioned that elicitation techniques can be combined to obtain high quality requirements and suggest a model for improving eliciting requirements using a combination of RE techniques. However, their study only mentions the use of combination of Interview with Brainstorming without neither discuss the pros and cons of those nor describe whether the combination was successful or not.

Second study is from Saeeda et al. [109], who cited that Interview and Prototyping could assist each other. Once again, the study mentions that Prototyping is very helpful to clarify requirements and can be a great complement to Interview without point their strengths, weakness or how exactly one technique could diminish the weakness of the other. Although we did not retrieve many possible combinations from other studies, it is important to mention this is the very difference of this study from others and this Research Question (RQ) aims to suggest some.

Third study is from Yousuf and Asger [32], who suggests some combinations captured in previous studies they investigated. Combinations such as Interview with Prototyping (same suggestion from [109]), Interview with Observation, Interview with Ethnography, Ethnography with Questionnaires, and, finally, Prototyping with Joint Application Development (JAD).

Meligy et al. [146] report two combinations Interview with Observation and Ethnography with Prototyping with special attention in describing valuable insights on Ethnography assistance for Prototyping.

As informed previously, one of the goals of this study is to analyze and suggest combinations of RE techniques to assist in overcoming the identified challenges. In a different view to help the reader to understand how a requirement elicitation technique could help surmount the identified challenges, Table 3.17 refers to the categories of challenges and lists the requirements elicitation techniques that have characteristics that could assist the practitioners in overcoming them as this table will help clarify the combination suggestions. Furthermore, in the Chapter 4 presents a compilation of pros (advantages) and cons (disadvantages) of 21 requirements elicitation techniques to assist practitioners in analyzing and comparing them.

Also, when using techniques that are very similar and sometimes overlaps themselves, such as Observation and Ethnography, the rationale was to combine techniques with different characteristics to provide greater coverage of the possible categories of challenges (problems) that can be found during the requirements elicitation task.

Since environments and projects are different, it is necessary to combine Requirements Elicitation (RE) techniques for a better coverage and more complete RE phase and based on the findings and information posted in the Table 3.14 and also, on the described strengths and weakness of the techniques analyzed on this study, we suggested some

Challenge **Technique**	Cplx	Trl	Comm	Exp	Prio	Doc	Engg	Vol	REw
Interview		X		X				X	
Prototyping			X				X	X	
User Stories			X	X	X		X		
Persona			X		X				X
Observation						X	X		X
Questionnaires			X		X		X		
Mind Mapping	X			X		X			X
Ethnography						X			X
Brainstorming			X						X
Data & Document analysis	X								
FDD					X				
Focus group			X	X				X	
JAD	X		X					X	
Laddering		X		X	X		X		X
Legacy systems analysis				X				X	X
QFD			X		X	X			X
Scenarios			X				X		
Stakeholder analysis							X		
Storyboards		X							X
TDD	X				X			X	
Workshop	X		X				X		

Cplx=Complexity and Scalability of the projects; Trl=Translation of Requirements; Comm=Communication; Exp=Expertise from team or user; Prio=Negotiation and Prioritization of Requirements; Doc=Documentation; Engg=Stakeholder Engagement and Availability; Vol=Volatility of the Requirements; REw=Requirements Elicitation work.

Table 3.17: Techniques and the challenges they can assist

smart combinations showing, for some challenges, which RE technique could be used to assist the most common difficulties when dealing with RE tasks.

Each scenario has its own particularity, whether due to the culture, size or line of business of the industry or the line of research, type of project or user/client profile, it is a fact that there is no single model to be followed. To better illustrate the objective of this research question for the work, we present, below, some hypothetical scenarios that could happen and what could be the techniques used to overcome them.

a. Customers do not have time or are not willing participate in long or several meetings and have different priorities on possible requirements

In a more crude analysis, lack of time or customer involvement, while their priorities are different, can mean two categories of challenges: “Stakeholder Engagement and Availability (Engg)” and “Negotiation and Prioritization of Requirements (Prio)”. Furthermore, the client’s lack of time or wiliness in collaborating can also pose a challenge related to “Communication (Comm)”. So, to overcome this scenario, we might use techniques which their strengths are related to those challenges. In this case some combinations were referred in the literature. For this example, we could use one of the following combinations:

- **Prototyping + Questionnaires [241], [242], [243], [101], [244], [44]**
  - **Prototyping** the engineer can get some assistance on “Stakeholder Engagement and Availability (Engg)” and “Communication (Comm)”
  - **Questionnaires** it can get aid for “Negotiation and Prioritization of Requirements (Prio)”
- **Persona + Scenarios [245], [39], [246], [247], [248]**
  - **Persona** to assist with “Communication (Comm)” and “Negotiation and Prioritization of Requirements (Prio)”
  - **Scenarios** to help with “Stakeholder Engagement and Availability (Engg)”

b. Customers do not have proficiency on the problem being solved, which is also not previously document and hard to explain. Project is complex due to business rules and difficult in making consensus among stakeholders

When reading carefully we realize that this is a sort of project that is quite common these days and it leads to a few categories of challenges, such as: “Expertise from team or use (Exp)”, as clients are not able to clear explain the problem; “Complexity and Scalability of the projects (Cplx)”, as stated in the example, and very common in industry; “Documentation (Doc)”, as previous documentation is sparse and rules difficult to explain and document. For this example, we found in the literature some combinations that we could use as follows.

- **Workshop + Interviews [32]**
  - **Workshop** which is useful to complex and large systems [209] to assist with “Complexity and Scalability of the projects (Cplx)”
  - Interviews which helps avoid misinterpretations [226] and provide useful insights [224] so it can assist with “Expertise from team or use (Exp)”

- **Observation** to help with “Documentation (Doc)” category of challenge as it helps understand how users operate the system and identify their needs [209]

c. Sponsor reported that project need to address issues related to some categories of challenges “Expertise from team or use (Exp)” as stakeholders can only address parts of the domain, “Complexity and Scalability of the projects (Cplx)” as system has complex business rules, “Documentation (Doc)” as there are a large number of users and stakeholders, “Negotiation and Prioritization of Requirements (Prio)” due to the number of stakeholders and “Requirements Elicitation work (REw)” as some non-funcional requirements are being demanded

In this straight forward example, we will use the identified pros and cons of some techniques to build our combination. Based on the identified strengths, we could use the combination of Mind Mapping and Persona the techniques as follows.

- **Mind Mapping + Persona**
  - **Mind Mapping** can help with “Documentation (Doc)” due to strength on information visualization and organization [158].
  - **Persona** on its turn can assist with “Negotiation and Prioritization of Requirements (Prio)” as the prioritization of the requirements [230], [122] is one of its strengths
  - **Mind Mapping** can also assist with “Complexity and Scalability of the projects (Cplx)” due to its advantage of reducing the cognitive load making easier to human minds [175].
  - **Persona** also can help describing how users interacts with the system [230], [122]
  - **Mind Mapping** still can help with “Expertise from team or use (Exp)” by helping associating ideas and promoting creativity [175].
  -

We also could try a combination of Workshops and Questionnaires along with User Stories to cover the categories of challenges identified in this hypothetical scenario b.

- **Workshops + Questionnaires + User Stories**
  - **Questionnaires** can assist with “Dcocumentation (Doc)” issues in case the problem is to reach a large number of stakeholders [209]

  - **Workshops** can help with “Complexity and Scalability of the projects (Cplx)” as it is useful for the elicitation of complex requirements [209]
  - **Workshops** also aids with decision-making and consensus among stakeholders [240] therefore assisting with “Negotiation and Prioritization of Requirements (Prio)”. For that challenge Questionnaires might be used as well.
  - **Questionnaires** also could help
  - User Stories also helps with “Expertise from team or use (Exp)” as it is a way to express user needs in an informal way [171] and improves collaboration among stakeholders [39]
  - **Questionnaires** are also resourceful in retrieving non-functional requirements, by handling multiple choices [32] and focused questions easy to tabulate [231]

d. Project do not have previous documentation and customers do not have time and do not know how to explain it. As long as it does not jeopardize the work, user is willing to demonstrate by using the legacy software that is in place. Also, non-functional requirements, such as lack of security or performance are old complains and are some drivers for the development project. Finally, it is not clear which currently functionalities the new software should cover.

Again, there are some categories of challenges that can be identified by this last example: “Documentation (Doc)”, as previous documentation is not available; “Expertise from team or use (Exp)”, as user does not not know to explain it; Quality or non-functional requirements are some of the reasons that the project was designed leading to the “Requirements Elicitation work (REw)”; also there isn’t a firm definition about which functions the software should have which implies that “Scope and Volatility of the Requirements (Vol)” might be present as well. One important information lies into the example, the wiliness of the user to demonstrate how he currently works.

- **Observation or Ethnography + User Stories + Interview or Prototyping**
  - **Observation** can help with “Documentation (Doc)” as it is useful to review definitions and requirements [101], [32] and in identifying needs of the user [209]. The same applies to Ethnography which can be resourceful for unstructured data [216] and in-depth findings [215]
  - **User Stories** can be helpful with “Expertise from team or use (Exp)” as it represents a way to express user needs [171] and improves collaboration of users [39]

- **Interviews** or **Prototyping** are two techniques that can assist with “Scope and Volatility of the Requirements (Vol)” depending on the source of the problem. If the problem is more focused in avoid misinterpretation of the requirements [226], [224] then Interview could be the option whereas Prototyping could be used in case there is a necessity to same development and interface definition time [109], [32]
- **Ethnography** could be used for “Requirements Elicitation work (REw)” as it is useful in identifying the user motivation to use the system [209]

Finally, in this fictional scenario, the user can demonstrate the functions by using the current software, but as long as it does not interfere in the work. This situation happens in day by day projects, and that is one reason to not use, for instance, Observation, specially the active observation, as one of its downsides, as presented in table 4.4 the person being observed should accept this situation [164]. Also active observations interrupt the users operations as the engineer interacts with the user.

In fact, by referencing the the tables 3.14 and 3.17, the reader can see that there are a few options of techniques that could be combined in agreement to their characteristics and accordingly with the preference or knowledge of the engineer responsible for the RE activity.

RQ.3. Summary: Research found that combinations elevates the quality of requirements elicitation phase. In literature the most referenced are “Prototyping and User Stories”; “Prototyping and Questionnaires”; “User Stories and Interview”; and “Interview and Prototyping”. Also, the technique of Interview was mentioned as a possibility for combination with almost any other technique. In addition, this section presented the table 3.17 containing the techniques that could assist for each challenge with a few possibilities being demonstrated Section 3.3.3 based on the strengths of each technique that are presented in Chapter 4.

3.4 Survey

3.4.1 Survey Protocol

The survey follows the protocol suggested by Linaker et at [249] which aims to clarifies the objectives, population, design and procedures taken into consideration when creating and executing the survey.

Research Objective

The objective of the opinion-based survey was to confront the data gathered from the systematic literature review trying to find either a confirmation or a different trend on the use of techniques. We view the poll as a complementary action for the literature review. In the SLR detailed in the previous section 3.1 we described the objective of identify requirement elicitation (RE) techniques and challenges related to the RE tasks in order to suggest some combinations that could help overcome the challenges. This survey was designed with the intention to verify which of the identified techniques are actually known and used by the community as well as which challenges are perceived.

In other words, this research is not intended to be exhaustive in relation to the findings, but rather to provide additional information to readers about elicitation techniques. For all these reasons, we understand that the research would only corroborate or contradict the results identified in the SLR as a way of complementing the study.

Target Population

The target for this survey are all the community of engineers and analysts that work with Requirements Elicitation (RE), in special the ones that work with Agile Software Development (ASD).

We did not select the so-called specialists to answer the poll, instead we left the survey open to the agile software community. The reason for this is because we understand that any selection of experts to answer the survey would be a subjective selection and, consequently, biased as we are appointing experts based only on our circle of relationship and areas of knowledge. Still, we emphasized that the survey was targeted at technology practitioners.

Sampling Frame

Although the questionnaire can be applied to anyone in the interest area, i.e. engineers, analysts, managers or other related to requirements area, since participants were recruited primarily through personal contacts, most of the participants are from Brazil due to relationships on or out social medias, university or work that were used to disclose the questionnaire.

Therefore, after we have identified some elicitation techniques described in the literature, we continued this work, and double-checked if the identified requirements elicitation techniques are used or known, in the industry by the agile software development teams, at least for the near community by conducting this online survey with the information and communications technology teams.

Questionnaire Design

The survey was presented in two main languages, English and Portuguese having a total of fifteen (15) questions. The questionnaire was built and distributed through the Google Forms platform and it was opened by around ninety days and the burden time to provide the answers was about twenty (20) to twenty five (25) minutes.

The target audience had to consent that participation was anonymous, voluntary, and with the exclusive purpose of contributing to the success of the research, in addition to the fact that the responses collected could be stored in perpetuity, which could be used anytime for journal publications, conferences, and blog posts. Moreover, they could leave the survey any time before clicking the send button without any discomfort since the process of responding was unsupervised. Finally, an email was provided in case participants had any problems or questions to the researchers

It initiates with the qualification of the respondent by asking for the location (country, state and/or city), gender, age, level of education and employment status. There are no plans to extract respondents or analyze sex or location correlations with responses. These are just informational questions to have a more accurate view of the respondents' profile.

The survey also requests a confirmation of the area of expertise and experience of the respondent, as well, as it checks whether the participant is or not involved in Agile Projects. On this hand, it is interesting to have this information and correlate experience and age with the specific answers about knowledge of the techniques presented in the survey. There is a total of eight (8) questions to qualify the respondent as shown in Table 3.18 Survey: Qualification questions. For a complete view of the questions and answering options, please refer to the appendix A.

Number	Questions to qualify respondents
Q1	Please state the Country, State and City that you work.
Q2	What is your gender?
Q3	What is your age range?
Q4	What is your level of education?
Q5	What is your employment status?
Q6	How many years of experience do you have in Software Development?
Q7	Do you work or is involved with Agile Projects?
Q8	What is your main area of expertise (or main role)?

Table 3.18: Survey: Background questions

The second part of the poll is focused on assessing the felling of the respondent in terms of level of difficulty being considered for the requirements elicitation phase. Also, it requires some evaluation from the participant about thirty (30) of the most elicitation techniques usually employed in this phase.

Continuing the second part the participant is asked to provide with his or her view about requirement elicitation challenges, strengths (pros) and weakness (cons) of the cited techniques. It is also possible to contribute with suggestions of techniques one is familiar with and are not listed in the poll. Finalizing the poll there is a final question when one can state this contact number or email in case one wants to take part of any more detailed interview. The last four (4) questions are not mandatory. One important remark, to make sure we could have a clear view from the respondents we added the words gathering and specification as synonym of elicitation. Once again, the complete form of the survey is at appendix A.

Number	**Questions about elicitation**
Q9	How difficult do you consider gathering or elicit the correct requirements for a project, system or application?
Q10	Please state which of the following tools and practices you are familiar and the level of knowledge.
Q11	What are the main challenges on requirements specification / gathering / elicitation phase ?
Q12	From your knowledge, please state the pros and cons of the tools, practices or techniques you most use?
Q13	Do you use or is familiar with any technique, tool or practice not listed? Please describe the pros and cons of it.
Q14	Based on your answers, is there any tool, practice or technique that your knowledge is not good or excellent and you would like to learn and use?
Q15	We will send a more detailed questionnaire to some of the respondents for further developments. In case you have interest in having a more detailed discussion please state your name and e-mail.

Table 3.19: Survey questions - elicitation

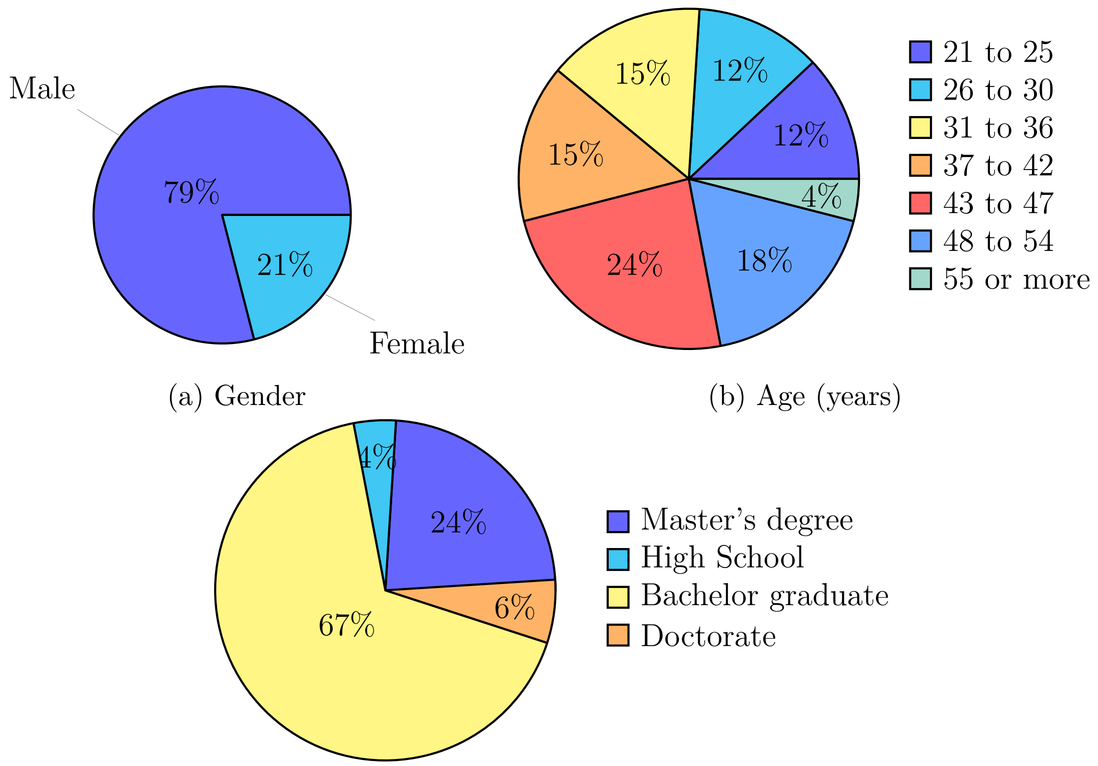
3.4.2 Answers and Analysis

Once the systematic literature review was finalized the opinion-based survey was closed as well and the results had been analyzed.

Respondents profile

The initial questions were defined to qualify the respondents in terms country or location they live, gender, age, employment status and level of education. Due our own location and community relationship, the qualification showed us, as expected, that most of the respondents are from Brazil, however, it is interesting to highlight that two answers came from different countries, one from United States of America and another from Portugal.

Regarding the other qualification questions most of the respondents are male as a gender and the great majority of the participants are a full time employee working with Agile Projects. We also had most of the answers in a specific option on the level of education and years of experience. Related to the others qualification questions, the numbers were less prone to a specific direction with more than one option sharing the higher numbers. The Figure 3.9 "Profile of the respondents" gives a better view of the numbers. The Figure 3.9 presents the statistics for questions Q.2, Q.3 and Q.4 presented in Table 3.18.



(a) Gender

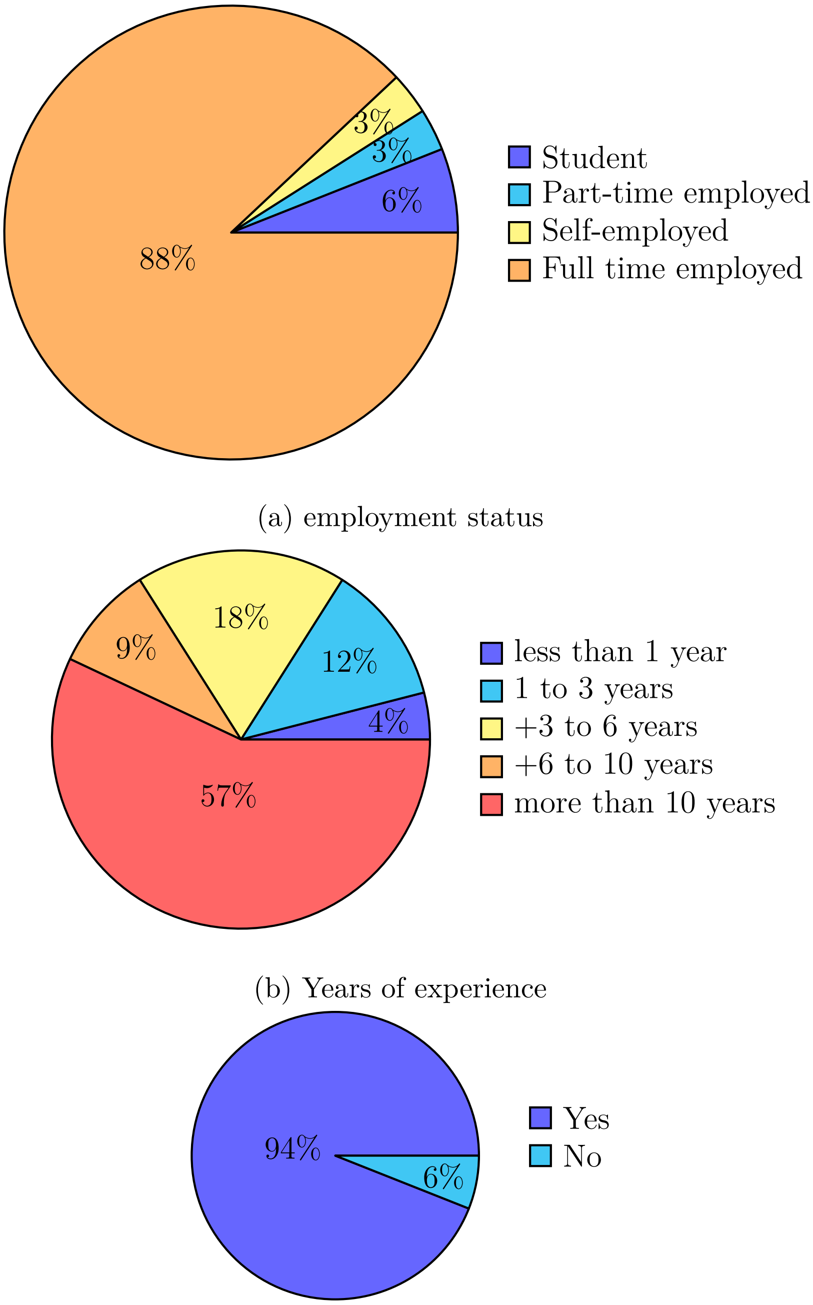
(b) Age (years)

(c) Educational level

Figure 3.9: Profile of survey respondents.

According to the statistics 79% of the respondents are men, while 21% are women. No one self-declared as non-binary and not declared at all, as for question Q.2 (3.18), shown in Figure 3.9 (a). Also, in terms of age range, related to Q.3 (3.18), we had a well-balanced percentage with 24% of the participants being from 43 to 47 years old followed by 18% from 48 to 54 years old, a number of 15% for the ranges of 37 to 42 and 31 to 36 years old whereas the ranges of 21 to 25 and 26 to 30 years old being 12% of the respondents. Finally, only 4% are participants with 55 or more years old, as shown in Figure 3.9 (b).

Regarding to the level of education, question Q.4 (3.18), an important item for the poll qualification, the results showed around 67% as being graduated, while 24% declared themselves with master degree. Interesting to mention that no one declared as a doctor



(a) employment status

(b) Years of experience

(c) Working with Agile Projects

Figure 3.10: Profile of survey respondents.

or post-doctor degree, although there are many within the community with this profile, as shown in Figure 3.9 (c).

The next four questions were related to professional profile. We initiate the assessment by questioning the respondent employment status (question Q.5 - 3.18), with the range of possible options being: student, unemployed, self-employed, part time or full time employed. Also the option “other” was included to cover any other possibility. The results showed that 88% are full time employed, 6% had declared as Student whereas 3% presented themselves as either self-employed or part-time employed, as shown in Figure 3.10 (a).

The qualification continued with the checking about the level of experience in Software development (question Q.6 - 3.18), which showed that the most of the respondents (57%) have more than ten (10) years of experience (Figure 3.10 (b)), and the great majority (94%) is related to Agile projects, as shown in Figure 3.10 (c), for question Q.7 - 3.18).

Finally, the last qualification question and one of great importance, are related to the area of expertise the respondent is involved or experienced in. The Figure 3.11 illustrates the results for question Q.8 (3.18). In terms of numbers, 39% as developers or involved in the development area. Also, the same percentage of 39% is experienced or involved in management area, whereas 12% present themselves as analysts or requirement engineers and 4% are within database administration or modeling area. Finally, 6% are involved in other areas, such as, data science, as shown in Figure 3.11.

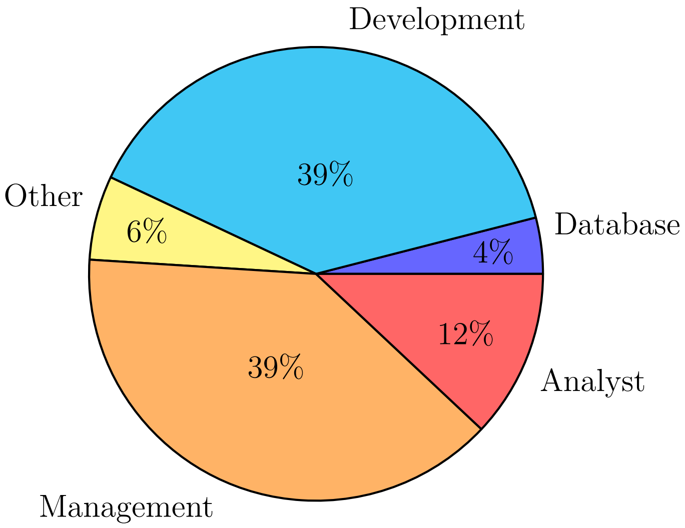


Figure 3.11: Area of expertise of the respondents

Analyzing the Challenges

Another piece of the poll relates to the requirements elicitation challenges. The main idea is to give a glimpse of the more cited challenges to cross with the ones encountered during the Systematic Literature Review. As requirements elicitation is considered, in many studies, as one of the most challenging phases of software development, we asked respondents to present the level of difficulty perceived by them for the correct elicitation or collection of requirements in a project, system or application. The Figure 3.12 presents the statistics for the question Q.9 from Table 3.19.

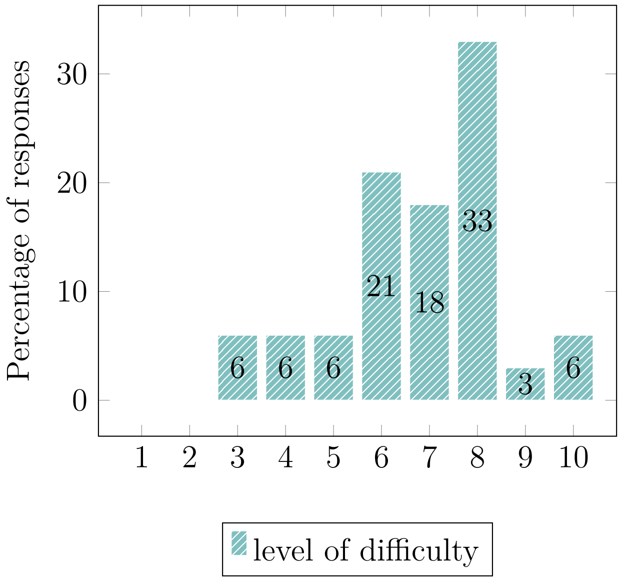


Figure 3.12: Degree of difficulty perceived by the respondents in obtaining requirements.

In a simple and straight analysis, we can cluster the results in three levels, assuming that any level eight or higher shall be considered as very difficult as well as levels one to four shall be considered the opposite view as not difficult and levels five to seven as regular or not so hard to perform. One can see the requirements elicitation phase is considered by 42% as very difficult, whereas 45% considered as regular and only 12% analyzed this phase as easy to accomplish. The level of difficulty perceived by respondents and each grade can be seen in the Figure 3.12. Also, we granted an open question (Q.11 from Table 3.19) to give the respondents a chance to provide what challenges were, in their view, the most relevant. Here is a compiled list with the five most relevant:

- lack of consistency or ability of the customer to explain what he really wants.
- lack of sponsorship.
- difficulties to extract the requirements from the customer.
- understand and translate the problem to be solved.
- uncover the requirements that customer does not inform.

These answers are, in essence, similar to the ones retrieved from the systematic literature review, which mostly described issues such as: poor communication [109], [173], [155], [181], [182], [159], [168], stakeholder engagement and availability [155], [160], [160], [158], [168], [60], lack of a strong sponsor [181], difficulties in translating requirements [150], [176], [168], prioritization problems [173], [109], [174], [160], poor documentation [109], [167], [174], [155], [176], [160], [181], [182].

Analyzing the Techniques

Level of knowledge

The main goal on the question about Level of knowledge (question Q.10 from Table 3.19) was to understand the techniques that were most familiar for the community and in which level they were. The results brought the scenario that Interview was the technique with the best percentage of knowledge of respondents with 80% considering their knowledge as in good or excellent level. It was followed by Prototyping with 76%, and then trailed by Documentation or Data Analysis with 70%, Brainstorming with 66% and User stories accounting for 65% as one can see in the figure 3.13. Also to be highlighted are the results for Use Cases, which 61% of respondents declared good or excellent knowledge of the technique, on the other hand, the Use Cases technique had one of the most balanced results with similar percentages for average, good and excellent, including one of the highest percentages of respondents who reported having knowledge at a level of excellence (28%), also shown in the Figure 3.13.

Among the techniques which had at least one answer, the least and most unfamiliar technique stated was the Laddering with 76% being declared as poor knowledge. This technique was followed by Blueprint with 62%, JAD and QFD with 57% and, surprisingly, by Ethnography, a traditional technique, with 56%. All results for least known techniques can be seen in the Figure 3.14.

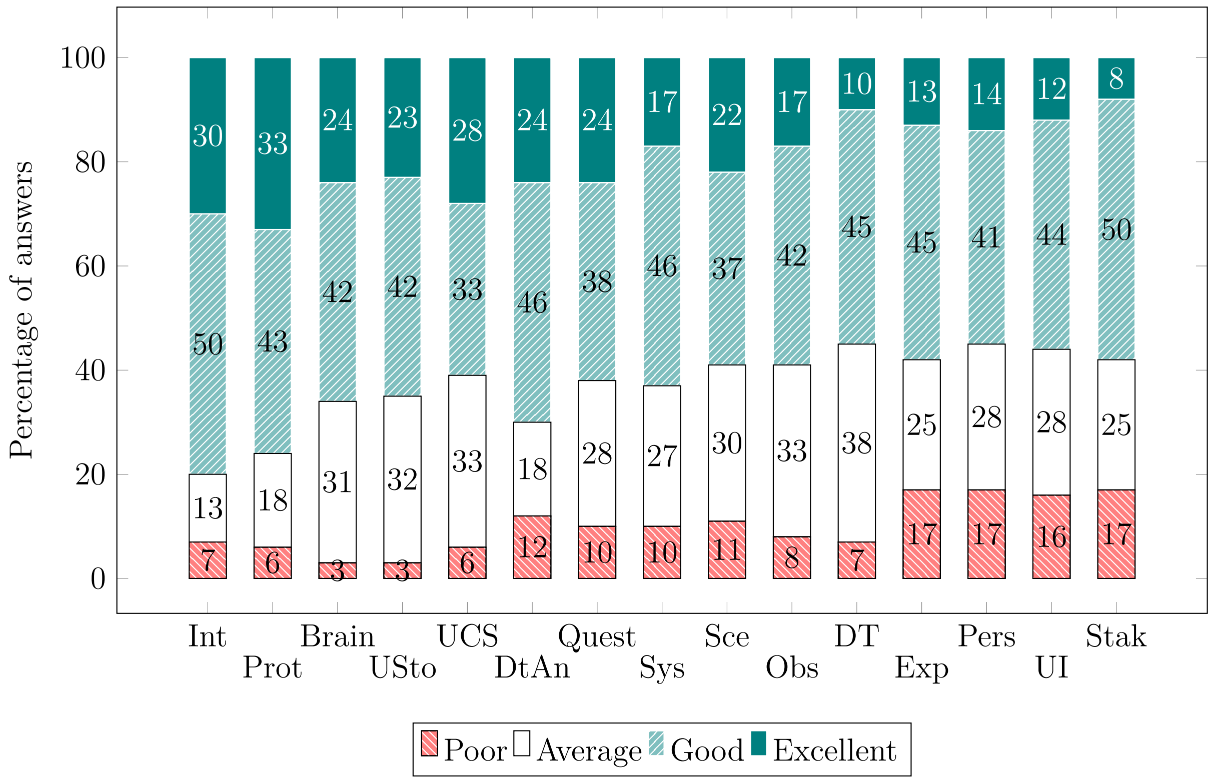
These results confirmed the tendency that the traditional techniques are the more familiar ones, at least in the community of the participants of this poll, with the exception of Ethnography which is being around for decades by now, however, at least in this community of respondents, is perceived as non-familiar technique.

Declared Pros and Cons

In another question (Q.12 - Table 3.19) it was requested to the respondents to declare which characteristics they view as the pros and cons of any of the techniques they were familiar with. Many answers were focused on Interview, Use Cases and User Stories techniques, so Table 3.20 presents a compiled view for these techniques.

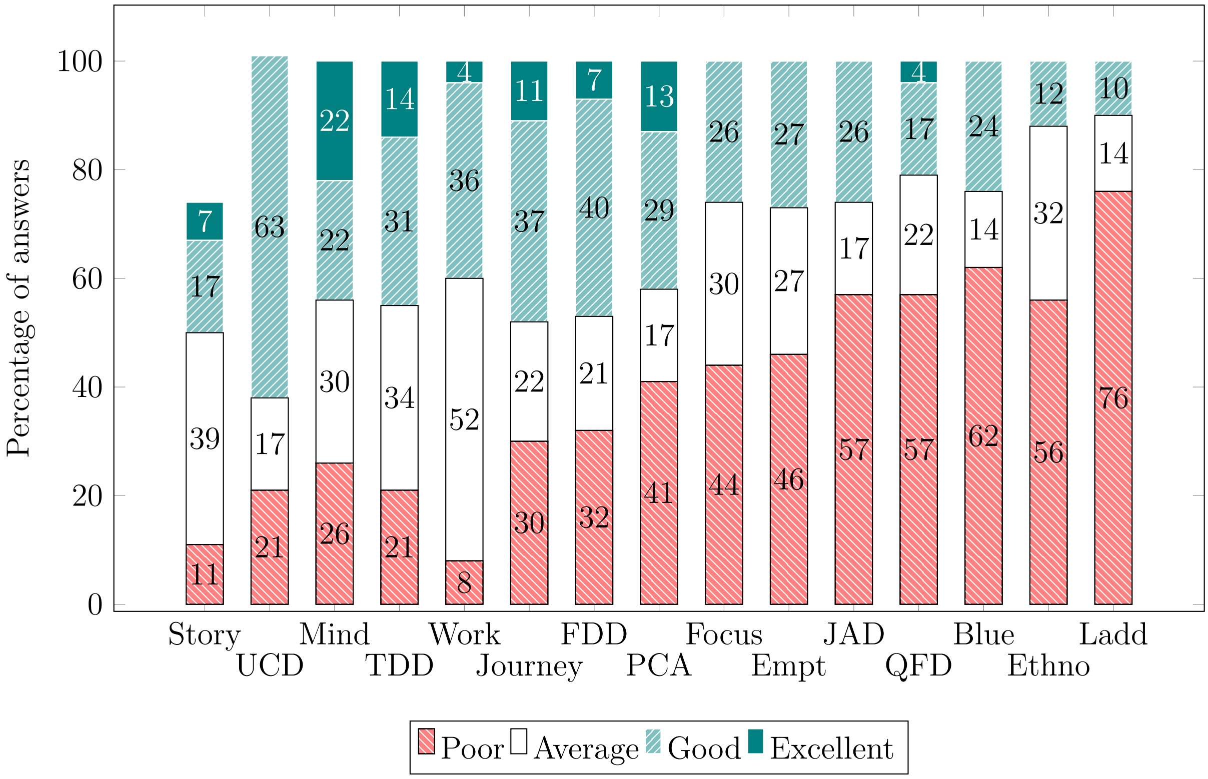
Going further

The poll also searched to uncover two other interesting scenarios. First, it requested to the participants to inform whether there was any other technique not listed in the poll that they were familiar with (Q.13 - Table 3.19). Although a few techniques are worth



Int=Interview; Prot=Prototyping; Brain=Brainstorming; USto=User Story; UCS=Use Cases; DtAn=Data and Document Analysis; Quest=Questionnaires; Syst=Analysis of Legacy Systems; Sce=Scenarios; Obs=Observation; DT=Design Thinking; Exp=Exploratory Research; Pers=Persona; UI=User Interface Analysis; Stak=Stakeholders Analysis.

Figure 3.13: RE Techniques - Level of knowledge of respondents - part 1



Story=Storyboards; UCD=User-Centered Design; Mind=Mind-map; TDD=Test-Driven Design; Work=Workshop; Journey=User Journey; FDD=Feature-Driven Design; PCA=Principal Component Analysis; Focus=Focus Groups; Empt=Empathy Map; QFD=Quality Function Deployment; Blue=Blueprint; Ethno=Ethnography; Ladd=Laddering.

Figure 3.14: RE Techniques - Level of knowledge of respondents - part 2

Technique	Pros	Cons
Interview	can be planned to be simple and straight forward useful to direct initial requirements	might be time consuming cannot elicit implicit or intuitive requirements depends heavily on the ability of the interviewer
Prototyping	very visual and straight forward useful for interfaces and for testing real-feel of the solutions	risk of customer oversimplifies the time for the real solution
Document or Data Analysis	less expensive that collect the data once access is granted does not require aligned agendas with customer	bias in selecting documents time to find useful information throughout the documents difficulties in accessing the data
Brainstorming	when people are willing to participate it gathers much more volume of requirements for simple requirements it can help consensus among stakeholders	not all stakeholders have the profile and good will to participate it requires someone experienced to conduct the sessions
User Stories	simple format makes easy to document and comprehend the requirement	sometimes shallow and imprecise making hard to discover the real requirement quite time consuming and repetitive
Use Cases	useful to confront users in case of questions or discordance on the delivery requirements	time consuming and usually hard for customers to read and understand not adequate for complex systems as long as become more extensive it is harder to be updated

Table 3.20: Pros and cons of most cited techniques

to mention, this question uncovered a common difficulty in the software development community, which is the mixing of the purposes of existing tools and techniques. Some answers pointed out practices of Scrum, Lean and Kanban, others pointed DevOps and Process Automation. To avoid bypassing these cited practices the list below gives a short description about them.

- **Lean manufacturing** aims in delivering maximum value to the customer by reducing waste, controlling variability, maximizing the flow of information, focusing on the whole process, and not on local improvements [250].

- **Scrum** is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value. It is not about increasing productivity or making the software development process more efficient [251].

- **Kanban** is an evolutionary change method that utilizes a pull system, visualization, and other tools to catalyze the introduction of Lean ideas, allowing teams to respond to dynamic changes, increase quality, reduce waste and improve predictability [250].
- **Business Process Automation (BPA)** is viewed as the automation of complex business processes and functions beyond conventional data manipulation and record-keeping activities [252].
- **DevOps** is defined as a paradigm, method or set of practices (and principles) that enables communication and collaboration for an efficient team working between developers and operators [253].

On the contrary, some newsworthy techniques were cited such as: MVP driven, Scrapping, Product Vision Box, Elevator Statement and Business Process Modelling (BPM). These techniques are beyond the scope of this study since they were not discovered either during the Systematic Literature Review or during initial tasks of this study.

Another interesting question (Q.14 - Table 3.19) to be mentioned is about which technique the participant sees their knowledge as poor or average and would be interested in improving their ability to work in it. A few techniques were mentioned showing the community really recognizes their gaps. The most mentioned techniques were: 1. Design Thinking; Persona; Mind Mapping; Test-Driven Design; Laddering; Joint Application Development (JAD). Also, Design Thinking concentrated around 20% of the answers on this matter.

3.5 Threats to Validity

This Section describes the threats to validity from this systematic literature review, and respective mitigation strategies.

- Search string: As well stated by Jarzebowski and Weichbroth [150], the limitation of any systematic literature review is the possible selection bias caused by a non-optimal search string. It is not different in this study. We made an effort to define the search string in an iterative manner working with other researchers to validity the string including running searches and assessing the results. The search string included several synonyms and alternative terms used by software engineers. We cannot, however completely exclude the possibility that some authors of relevant papers could have used other, less common terms and as a result, such papers were not found, therefore it is possible that not all interesting papers were covered by the defined search string.

- Language: Another possible weakness of this study' approach is that we focused on papers written in English, Portuguese or Spanish languages and it is possible that relevant papers written in languages other than these were not captured.

- Inclusion/Exclusion criteria: Another possible limitation concerns invalid choices regarding a article's inclusion/exclusion. The papers went through a two-round process of review and analysis. First this researcher analyzed the papers, excluding only the studies that were absolutely clear that were either duplicated or not related to Agile Software Development (ASD). In the second round, any study that was initially a question mark whether to include or exclude was entirely reanalyzed. Still, if there was any question about the contribution of the paper, it was then promoted to the quality assessment phase. Although, this researcher had extra care in analyzing the papers, an addition of a second researcher would be valuable to diminish the possible bias of the application of inclusion/exclusion criteria.

- Validity: As a systematic literature review aim to a specific topic, it is always possible that these results cannot be generalized since there might be terms and written papers that were not retrieved either in the automatic search string nor in the manual investigation. Even though we followed Kitchenham's [106] suggested process, there is some limitation regarding validity: (1) the data extraction process was carried out by a single researcher (even though guidance was retrieved from the masters advisor who is an experienced researcher); (2) the search did not consider all existing journals and conferences that could might carry relevant information.

- Poll Survey: Sample size: due to the limited number of participants, there is a limitation in the conclusion of the results and, as such, they are considered indicative and not conclusive.

  Location of respondents: most of the participants are from Brazil, which makes the answers very difficult to generalize to other profiles or countries due to the aspect of the locality or region of the participants. The survey was distributed among researchers and professionals linked to the community, university and work of the researcher and his relations, therefore most of the respondents are from Brazil and it cannot be expanded to represent the whole community.

Questionnaire: it is not possible to guarantee that all participants have understood in the same way all the questions. To minimize this threat we initially submitted the questionnaire to the advisor and to reviewers outside of the information technology area.

Combinations: due to time constraints to work in real projects the combinations were not tested, and the validation was performed by a Focus Group 4.3, therefore some of the combinations might not perform as expected in real world.

Survey Summary: The outcome confirmed that the more traditional and known techniques are Interview, Prototyping, Brainstorming and User Stories whereas Laddering, Ethnography, Blueprint and Quality Function Deployment (QFD) are the least familiar to the community. In addition, it uncovered a common difficulty in the software development community, which is confusing techniques with practices, such as Scrum, Lean, Kanban or DevOps. Finally, the survey indicated a few different techniques beyond the scope of the guide such as Elevator Statement, Product Vision Box, MVP driven, Scrapping and Business Process Modelling.

3.6 Chapter Summary

In this Chapter a Systematic Literature Review was described and conducted to identify the requirements elicitation techniques used by the Software Development teams and Information technology and communications professionals involved in the Agile Software Development. The SLR was composed of automatic and manual searches, performed in digital libraries, journals and conferences. We started SLR by creating a search string performed on indexing databases. The string returned 478 papers, of which, after analyzing the title, authors and year of publication and abstracts, 89 were considered duplicates and another 291 were rejected. On the other hand, after a manual investigation in journals and conferences, 18 papers were added to this study. Finally, after quality assessment 54 studies were selected. These papers revealed that the most investigated techniques are: Interview, Brainstorming, Ethnography, Observation, Prototyping, Questionnaires, Scenarios, Use Case and User Stories.

Furthermore, a few other techniques are mentioned, such as, Data or Documentation Analysis, Design Thinking, Feature-Driven Design (FDD), Goal-based Analysis, Joint Application Design (JAD), Journey Maps, Laddering, Legacy system analysis, Mind Mapping, Participatory Design, Persona, Rational Unified Process (RUP), Quality Factor Development (QFD), Storyboards, Survey and Workshop.

Also, the uncovered challenges were distributed in categories to simplify the reading and understanding. The categories are: Documentation (Doc), Communication (Comm), Translation of Requirements (Trl), Stakeholder Engagement and Availability (Engg), Scope and Volatility of the Requirements (Vol), Complexity and Scalability of the Projects (Cplx), Negotiation and Prioritization of Requirements (Prio), Expertise from team or user (Exp), Requirements Elicitation work (REw), Architecture and Software Quality (Qlty), Culture and Environment (Cult) and Other, assumptions, team issues (Oth) with the first four being the most cited within the selected studies.

Moreover, to complement the analysis, a poll survey was held within the community to confront the data gathered in the SLR in terms of the views of Agile challenges, requirements elicitation techniques and its strengths and weakness, as well, as to identify views and attempts of using different combinations to overcome the known challenges and issues of Agile model. Specialists were not nominated or selected to grant their view, instead, in order to diminish the bias regarding the results, we spread the poll to reach a wide spectrum of profiles being the respondents. The results reinforced a similar list of challenges and how the community is familiar with traditional techniques.

Chapter 4

Conceptual analysis - Description of Elicitation Techniques

This Chapter presents the main aspects involving the decision-making in the development of this study, its conception and presentation. Also this chapter is outlined by the description of the techniques found either in the literature review or found as a result of the survey. Nevertheless, there is no certain that one technique would be suffice for all kinds of projects. The quality and shortcoming of every strategy and tools depends upon the unique circumstance and circumstance [126].

4.1 Scope of this study

Among the most cited challenges, according to Table 3.11 (Chapter 3), are the lack of documentation, lack of customer involvement, difficulty of agile processes in dealing with complex projects, difficulties in dealing with non-functional requirements and inadequate prioritization of requirements. None of these challenges, and many others, can be solved using agile practices alone. On the contrary, it requires the sensitivity of the team to use different techniques, which, when combined, can help to understand different parts of the software and product context in order to be able to recognize, delimit and describe the requirements from the most diverse points of view necessary for the delivery of products. a complete software that adheres to what the customer expects.

Instead of creating yet another framework, we understand that it is necessary to present requirements elicitation techniques, with their characteristics, strengths and weaknesses so that they can be used by requirements engineering practitioners in order to achieve a complete description of the expected product or service.

Based on what we have learned from all the analyzed studies we understand that jump straight into development and collect requirements as you go jeopardizes the requirements elicitation. Under no circumstances, this means a return to traditional models since the need for agility is urgent nowadays, but rather a small step back at the beginning of projects seeking to obtain a greater clarity of objectives and, as a result, less errors and refactoring to accommodate requirements that were not properly analyzed at the beginning of the projects.

There are a few guides already created and published over the world wide web, as one can see in Table 3.16, so instead of actually publish another guide we focused in analyzing strengths and weakness and direct some examples that describe techniques and examples to be used by Agile Software Development community.

So, this work intends specially to provide some basics from pros and cons of requirements elicitation techniques in order to assist analysts and engineers.

4.1.1 Conceptual Proposal

When deciding which techniques to include in this study, we first carried a SLR and a Poll Survey to understand which ones were being discussed in the literature and also are noted in the industry as well. After the investigation we define a list of techniques to go deep in analyzing strengths and weakness which can be seen in the section 4.2. Techniques Description.

As already mentioned in RQ.3 (Section 3.3.3), the main idea was to analyzed each of the techniques elected in this study for its strengths and weakness and based on this information (pros and cons) apply the techniques to the challenge categories that they would be able to assist the community. Although there are two Tables (3.17, 3.14) in the Chapter 3 (Section 3.1) we understand that the Figure 4.1 presents a simplified view about which technique could help in each of the identified challenges.

Figure 4.1 to check which technique or techniques would be able to assist according to the project or company environment one is dealing. Figure shows the information in two perspectives, one it is possible to check which challenges one technique can assist, by verifying the presence of the icon in each category box; and two, check which techniques can assist in one specific challenge. By combining both views, the reader is able to notice that, for instance, “Workshop” can assist with “Stakeholder Engagement and Availability” and “Scope and Volatility of the Requirements”.

Moreover, the Figure 4.1 do not show combinations of techniques, instead it focus on the relation between challenges and the techniques that can help practitioners to overcome them.

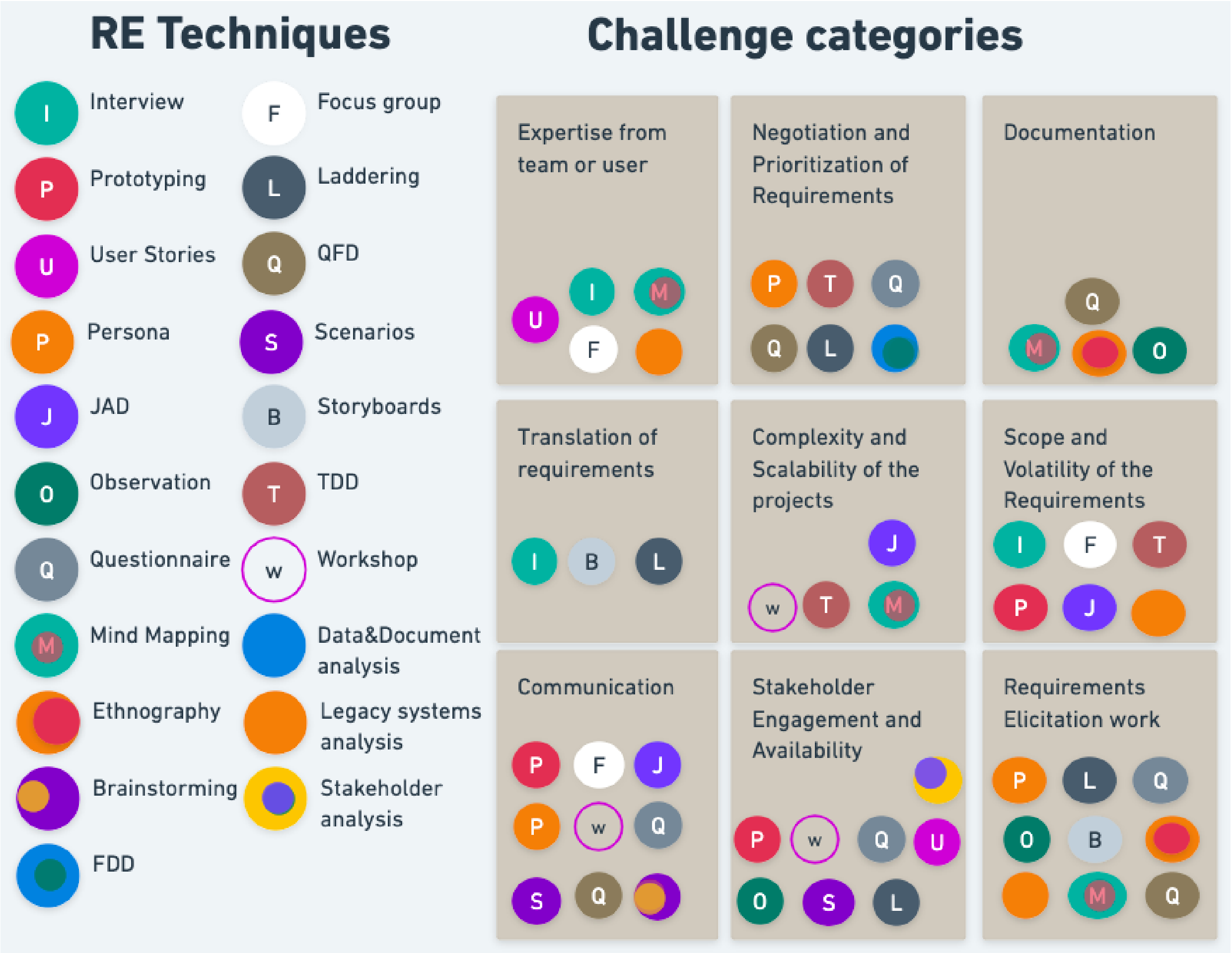


Figure 4.1: Techniques and the challenges they can cover

4.2 Techniques Description

4.2.1 Basic concepts

Throughout the requirement engineering elicitation process there are several techniques designed to help the practitioners in each specific phase [254], [255], [256], [257], [32], [158], [159], [39], [65], [258], [46]. The most suitable technique selection requires in-depth knowledge of the problem domain, techniques available, and requirement sources (domain experts, organization, market, users, and customers) from where requirement collection is carried out [259].

From the complete list of requirements elicitation techniques identified in either the Systematic Literature Review (SLR) or through the Survey done with practitioners we have selected the most referenced ones to in-depth studies, therefore in the following sections we describe "Analysis of Legacy Systems", "Brainstorming", "Data and Document Analysis", "Ethnography", "Feature-driven design (FDD)", "Focus Groups", "Interview", "Joint Application Development (JAD)", "Laddering", "Mind Mapping", "Observation", "Persona", "Prototyping", "Quality Function Deployment (QFD)", "Questionnaires", "Scenarios", "Stakeholders Analysis", "Storyboards", "User Stories" and "Workshop".

4.2.2 Analysis of Legacy Systems

Also called, Legacy Systems Analysis or Old system analysis is when the data is gathered for a system to replace an existing one. It is useful technique to collect the depth knowledge of system [207]. When using this technique it is necessary to be very careful to not over analyze the existing (or legacy) system as there is a risk in trying to replicate the old system making the new system becoming too constrained [207]. Analysis of the Legacy systems helps to understand the exact situation of the system and its entire process [208]. Table 4.1 shows some of the strengths or pros, as well as the weakness or cons investigated and collected in this research.

Strengths (pros)	**Weakness (cons)**
can uncover a large amount of knowledge [207]	over analyzing the existing system and the new system become too constrained [207]
great foundation to for further development [208]	risk in constrain the new system based on the old one [207]

Table 4.1: Analysis of legacy systems pros and cons

4.2.3 Brainstorming

Brainstorming is a meeting in which each participant can freely express its idea of the system requirements [211]. A brainstorming session typically occurs in a panel format and includes a leader (or facilitator), recorder, and around 10 members [210]. It enables a group of people to take advantage of conventional an logical thinking, as well as embracing spontaneity [65].

It is a way to attune the user's mind to the requirements [191] as it is an open discussion forum which is organized where all stakeholders share their ideas, inputs, suggestions in the form of requirements and, these requirements are then documented for future analysis [136]. Table 4.2 shows some of the strengths or pros, as well as the weakness or cons investigated and collected in this research.

Strengths (pros)	**Weakness (cons)**
allows the discovery of new and innovative solutions to existing problems [209]	lack of participant preparation for brainstorming [210]
allow consensus and improves the working atmosphere [209]	lack of advanced tools or creative approaches to complement the brainstorming process [210]
stimulates creativity [211]	many of brainstorming ideas may not be quality ideas [209]
help to answer specific questions [210]	not for all users. Might consider user experience for the correct use of the brainstorming [211]

Table 4.2: Brainstorming pros and cons

4.2.4 Data and Document analysis

It is a systematic procedure for reviewing or evaluating documents—both printed and electronic (computer-based and Internet-transmitted) material. Like other analytical methods in qualitative research, document analysis requires that data be examined and interpreted in order to elicit meaning, gain understanding, and develop empirical knowledge [260].

Documentation analysis is the process of analyze the documents related to the problem domain to gather the information, which is flow with in the organization. It is a useful technique to find in-depth knowledge about a particular task [207].

Table 4.3 shows some of the strengths or pros, as well as the weakness or cons investigated and collected in this research.

Strengths (pros)	**Weakness (cons)**
Cost-effectiveness [212].	Documentation is not available [213].
Lack of obtrusiveness and reactivity [212].	Biased selection [213], [214].
investigator's presence does not alter what is being studied [214].	Documents are produced for specific purpose so might have insufficient details [212].

Table 4.3: Data and Document analysis pros and cons

4.2.5 Ethnography

It is an observational technique where an analyst studies a culture or an environment in order to deduce requirement [65]. Ethnography emphasizes the perspective of seeing things from the side of those under study [215].

A single requirement engineer may participate in a given environment in order to understand given cultural activities and way of life of its environment [65], however the requirement engineer should be able to listen, show interest in what people say and do to complete the task [215]. The Table 4.4 shows some of the strengths or pros, as well as the weakness or cons investigated and collected in this research.

Strengths (pros)	**Weakness (cons)**
it provides insight to a user' own motivation to use the system [209]	time consuming for larger projects or complex requirements [101]
helps in identifying needs of the user who even the users may not be aware [209]	researcher needs to have a deep knowledge of the problem domain [215]
useful for unstructured data [216]	interacts with a user or social group [216]
useful to provide in-depth findings [215]	it is difficult to analyze the social requirements of the people [209], [217]

Table 4.4: Ethnography pros and cons

4.2.6 Feature-driven design (FDD)

FDD uses short interactions to develop a functional software and it mainly focuses on design and building phases of software development [219], [221]. There are five sub processes that have clearly defined entry and exit criteria in FDD which are: Develop an Overall Model, Build a Features List, Plan by Feature, Design by Feature and Build by Feature [221], [218], [219].

The Table 4.5 shows some of the strengths or pros, as well as the weakness or cons investigated and collected in this research.

Strengths (pros)	**Weakness (cons)**
focus in improving quality [218]	does not provide guidance about whole development process rather it mainly focuses on design and building phases of software development [219]
helps to get early feedback [218]	lacks the ability to response rapidly changing requirements [220]
produces workable and tangible independent feature on every iteration [221]	greatly relies on staff's experience and capabilities for successful execution of the development process [220]

Table 4.5: Feature-driven design (FDD) pros and cons

4.2.7 Focus Groups

Focus Groups is "non-standard" technique of information gathering, based on an apparently informal discussion among a group of people [222]. The debate is lead by a moderator who directs the discussion according the purposes outlined on the participants' characteristics [223], [222]. There is also the role of the Observer, who is interested in the non-verbal behavior emerging from the interaction and integrates verbal information rising from the conversation [209], [222].

The Table 4.6 shows some of the strengths or pros, as well as the weakness or cons investigated and collected in this research.

Strengths (pros)	**Weakness (cons)**
data is gathered quickly [209].	volume of data can be hard to summarize or classify [223].
easy-to-organize and inexpensive technique [222].	can be hijacked by dominant participant [223].
can tackle sensitive points [223]	rely heavily in the experience and skills of the moderator [223], [222].
encourage participants that avoid individual interviews to provide their view [223].	for sensitive topics, it can be hard to get honest insights [209].
examines issues in a holistic manner [223].	does not perform well with very heterogeneous groups [222].

Table 4.6: Focus Groups pros and cons

4.2.8 Interview

Interviews are generically defined as an interaction among two or more people with direct contact for at least one party to learn something from the other [223]. The interviews were considered effective in either global software development (GSD) or traditional environments because these enable analysts to obtain more detailed information depending on the quality of the formulated questions [39].

The are different types of Interviews and usually the more accepted are:

- Structured - follows a rigid script
- Semi-structured - allows views and opinions sill follows a script
- Non-structured - presents a theme and rely on spontaneous generation of questions during the interview

Also, according to the type of the interview, it can use close or open questions and quantitative and qualitative data [223]. The Table 4.7 shows some of the strengths or pros, as well as the weakness or cons investigated and collected in this research.

Strengths (pros)	**Weakness (cons)**
flexible [224], [223]	time consuming to set up and conduct [225], [223]
possibility of high response rate [225], [224]	risk of interviewer bias [225], [223], [224]
possibility to correct and avoid misinterpretation [226], [224]	dependent on the skill of the interviewer and capacity and cooperation of the respondents [223]
Need only fewer participants to provide useful and relevant insights [224]	can generate data difficult to analyze [223]
can involve reality [226]	respondent must be wised selected [224], [226]

Table 4.7: Interview pros and cons

In addition, Interviews can easily incorporate other elicitation processes [223].

4.2.9 Joint Application Development (JAD)

JAD is a technique where groups of stakeholders and management work together towards building a project [65]. It involves all the available stakeholders investigating through general discussion both the problems to be solved, and the available solutions to those problems [99]. The various participants from same or different domains are directly involved with a highly structured interview is employed over a period of three to six months. [65].

The Table 4.8 shows some of the strengths or pros, as well as the weakness or cons investigated and collected in this research.

Strengths (pros)	Weakness (cons)
rapid decision making about the problem and its solution [217].	demands engineer with high skills in the technique [65], [32].
handles well the changeability in requirements [217], [39].	possibility of creating scope creep in a project if the customer's requirements are not managed well [127].
provides a direct communication and cooperation among stakeholders, useful for conflict resolution [39], [217], [32].	the selection of people to participate in the workshops may alter or bias the results.[209].
promotes user feedback [32].	requires a huge experience in problem domain [217], [126].
useful for non-functional requirements [39].	cost and time consuming [32], [65], [126].
effective for understanding social issues and knowledge from the domain [39].	when not properly planned can lead to wastage of resources and time [32].

Table 4.8: Joint application development (JAD) pros and cons

4.2.10 Laddering

Laddering is a sort of interview where questions are arranged in a hierarchical format [32]. A series of simple questions are asked from the stakeholders which are answered in a clear way by them [217].

It is usually divided in three phases: create, review and motivation and it is similar to a structured interview [60]. Stakeholder domain information is vital for the success of this technique [217]. The Table 4.9 shows some of the strengths or pros, as well as the weakness or cons investigated and collected in this research.

Strengths (pros)	Weakness (cons)
close contact with stakeholders [217]	it is long and time consuming technique [32].
help prioritization of stakeholders needs [217]	technique becomes complex when requirements are in large number [32], [217].
organize requirements in hierarchical format that is easy to understand [217]	it assumes that knowledge is possible to be hierarchically arranged [60]
can be used in collect the tacit requirements. [60].	maintenance is hard in case of volatility of requirements [217], [32].

Table 4.9: Laddering pros and cons

4.2.11 Mind Mapping

Mind Mapping is a technique where a diagram is used to connect ideas and concepts based on a central or main idea, word or concept [175]. This technique is going to branch factors to the main tasks to make sure none of the important aspects of any of the tasks is ignored, and the requirements tracked in this way are easy to retract [109]

Mind Mapping encourages people to think of, organize and represent information within a radial hierarchy, by locating the most important concept at the center of a given diagram and relate it to other concepts [183]. In this diagram, if we think the most important are remembered first, the elements are ranked intuitively in accordance with the importance of the concepts related to a domain [175].

The Table 4.10 shows some of the strengths or pros, as well as the weakness or cons investigated and collected in this research.

Strengths (pros)	**Weakness (cons)**
assists on information visualization and organization [158].	relatively time consuming [227].
helps associate or group ideas [175].	people tend to write lists instead of diagrams [227].
promotes creativity and innovation [175].	find an effective keyword for a subject is not easy for all types of users [228].
makes it easier for the human mind to process information better, by reducing the cognitive load [175].	links being made are limited to simple associations [229].
promotes people to work together and increases productivity and memory retention [183].	limited in dealing with more complex relationships and can become inconsistent in terms of details [229].

Table 4.10: Mind Mapping pros and cons

4.2.12 Observation

In the Observation technique, the requirements engineer observes the user‘s environment without interfering in their work, in a passive way, in other words, not interacting with user [146]. It is a sort of Social analysis [32]. It involves investigation of user’s work and taking notes on the activities that take place [207].

Observation is almost always complemented with other forms of data collection such as interviews or document analysis [146]. Requirements Engineer observes the environment of the customer and the tasks being done to carry the process [217]. The Table 4.11 shows some of the strengths or pros, as well as the weakness or cons investigated and collected in this research.

Strengths (pros)	Weakness (cons)
helps in identifying needs of the user who even the users may not be aware [209]	end users adjust themselves when being observed making the process artificial [217]
non-verbal and contextual cues can be analyzed [101].	observer must be accepted by the people being studied [164].
useful to confirm or reviews of definitions and requirements [101], [32].	expensive to perform in terms of the time required [209].
inexpensive method [32].	suffer from ambiguous interpretation as users motives are hidden and observer viewpoints may differ [101].
Helpful in work measurements i.e. how long particular task takes to be done [32].	it is difficult for analyst to understand why some decisions are made [32].
can uncover dependencies between stakeholders [169].	requires sensitivity and responsiveness to perceive a rich picture about the work context [67].

Table 4.11: Observation pros and cons

4.2.13 Persona

Persona is a description of a fictitious person that represents a larger part of the target group [158] therefore acting as representatives of real users during the system development [261]. This technique gathers, analyses and synthesizes information related to the users that are to interact with the software system [230].

Persona assists understanding how and when tasks are performed [262]. The intention of Persona technique is to support the project team to gain common understanding concerning user and stakeholder, as well as their needs and behaviors by utilizing them as actors in user stories [171].

The idea on this technique is that it is better to understand and completely satisfy the necessities of the critical users instead of poorly meet the needs of many [122]. Moreover, Persona can be used as the foundation and inspirational approach to develop the application ideas that are initially discussed [170]. The Table 4.12 shows some of the strengths or pros, as well as the weakness or cons investigated and collected in this research.

Strengths (pros)	**Weakness (cons)**
helps understand the user profile and goals [171], [77].	communication breakdown to the development team leads to failure [230].
resulting in more prescriptive scenarios and visual designs on how users interacts with the system [171].	project their own goals, motivations, skills and mental models on the software system development [230].
helps prioritize the requirements [230], [122].	no specification on how to document the output of activities can lead to non-proper description [230].

Table 4.12: Persona pros and cons

4.2.14 Prototyping

A prototype is a physical or digital embodiment of critical elements of the intended design while Prototyping is the process of creating it [263]. Prototyping is useful in agile requirements engineering (RE) as it enables obtaining feedback on requirements in a light-weight fashion [184]. Also, paper prototypes and informal drawings are very useful in discussions with users [264]. As an iterative tool it can enhance communication, enable learning, and inform decision-making at any point in the design process [263].

Prototype is a model of the software application that supports the evaluation of design alternatives and communication [167]. A prototype can range from a simple sketch through a mock-up or simple drawing to an incomplete version of the production software [184]. The Table 4.13 shows some of the strengths or pros, as well as the weakness or cons investigated and collected in this research.

Strengths (pros)	**Weakness (cons)**
helps user engagement and promotes innovation [32], [170].	developer misunderstanding of user objectives [109].
Support customer communication and allows early user feedback for requirements refinement [184], [170]. [32], [209].	risk of missing the big picture and thus the overall business needs [184].
Users and team get better understanding of the system. [32], [65].	user often resist changes if they had become used to a specific kind of the system [170].
can save development and interface definition time [32], [109], [109].	can be time consuming for complex or larger requirements and projects [209], [32].
useful when there is a great deal of uncertainty about the requirements [65], [209].	Effort and cost estimation may get high [32].
supports the evaluation of design alternatives and communication. [167].	it can be time consuming and costly [32], [159], [109], [39].

Table 4.13: Prototyping pros and cons

4.2.15 Questionnaires

Questionnaires are a technique of requirement elicitation formed by a document in which some objective questions are design sometimes along with their possible multiple options [136]. Questionnaires yield responses that are usually easy to tabulate or score, and the resulting data are easy to analyze [231].

To get precise results, the questionnaire should be clear, concise and structured to obtain genuine user requirements, objective and constraints [159]. Stakeholders need to fill in the answers which are relevant to them as per their domain knowledge [136]. The Table 4.14 shows some of the strengths or pros, as well as the weakness or cons investigated and collected in this research.

Strengths (pros)	Weakness (cons)
simple and low cost [159], [209].	no mechanism for participants to clarify or correct misunderstandings [217], [159].
questionnaires can be administered anonymously [231].	inflexibility to the stakeholder's language, interests, views [209].
can encompass a large number of stakeholders [39], [32].	Questions can be misinterpreted. [32].
yield responses that are usually easy to tabulate or score [231].	question ambiguities may arise [32].
Easy to handle as it is possible to use multiple choice or true false questions [32].	weak on getting further clarification regarding specific problem or requirement [32].
useful for statistics as the same question is asked to a large number of people [32].	feedback is not received [32].
economical technique [32].	questionnaire response rate is often low [231].
quickly applicable to many stakeholders [209].	may provide only a snapshot rather than rich picture [231].

Table 4.14: Questionnaires pros and cons

4.2.16 Quality Function Deployment (QFD)

QFD is a method for developing a design quality to satisfy customer's needs [207] focuses on delivering "value" by seeking out both spoken and unspoken customer requirements, translating their demands into design targets and communicating them throughout an organization [233].

It is an approach to convert the customer's requirements into product design. In QFD, the customer determines the importance of each and every requirement on a scale ranging from 1 to 5 [136]. The Table 4.15 shows some of the strengths or pros, as well as the weakness or cons investigated and collected in this research.

Strengths (pros)	**Weakness (cons)**
helps prioritizing customer requirements in order of importance from the customer viewpoint [233]	Applicable for limited number of requirements [136]
Establishment and maintenance of documentation due to the fact that information is stored in the matrices [233]	requires a lot of knowledge and skills of the team carrying out the method [233]
constant improvement of product quality [232]	complex matrices and analyses is time-consuming as methodology assume dealing with large amounts of data [232], [233]
increased possibility for breakthrough innovation [233]	Applicable for limited number of requirements. [136]
improvement of cross-functional team members and intra-organization communication [232], [233]	Difficulty of cooperation between multi-disciplinary teams [233]
Fewer design and service costs due to the reduction of irrelevant processes [233]	difficulties in the implementation of numerous conclusions resulting from the analysis into measurable service features [233], [232]

Table 4.15: Quality Function Deployment (QFD) pros and cons

4.2.17 Scenarios

Technique that gives a narrative description of user processes including actions and interactions [65]. Scenario is a textual representation of a problem and describes the interaction between user and system in a specific context.[167].

Scenarios requires an incremental and interactive approach to their development. User scenarios techniques are considered contextual as it explains the theories and context behind why a particular system function [65]. The Table 4.16 shows some of the strengths or pros, as well as the weakness or cons investigated and collected in this research.

Strengths (pros)	Weakness (cons)
ensures that end-user's view is considered for requirements elicitation [32].	write effective scenarios requires much practice and experience. [209]
useful for understanding and validating requirements [209], [32].	does not consider internal structure of the system [65].
simple and understandable for users [32], [209].	it is difficult to draw useful scenarios [32], [101].
useful for understanding and validating requirements [209].	it is not suitable for all types of projects [32].
When well-developed, helps organizations to be proactive and specific for the desired system [32].	do not cover all process or necessary complete view of the system [32].
take into account the normal flow, exceptional behavior, alternative paths [32].	rely on communication which can lead to misinterpretation [101].

Table 4.16: Scenarios pros and cons

4.2.18 Stakeholder analysis

A project stakeholder can be defined as "any individual or group who can affect or be affected by the project process or the project outcomes" [235].

Stakeholder analysis is an approach for identifying and getting knowledge about actors (individuals or organizations) regarding intentions, interests, influence and the interrelations they have. Aim of understanding the policy context and processes [234]. The Table 4.17 shows some of the strengths or pros, as well as the weakness or cons investigated and collected in this research.

4.2.19 Storyboards

Storyboards (or Story boarding) consists of a sequence of pictures, which show significant steps of the workflow [171]. It can be used as effective media to capture and explore the user experience by translating the story and script into scenes through who, what, when, where, and how using images and text [170].

Storyboards use images, text, audio, video, animation diagram to visualize the concept to the stakeholders [64]. It focus not on the mechanics (technical steps) to be taken, but rather what the product will do/not do when interfacing with the system [51]. The Table 4.18 shows some of the strengths or pros, as well as the weakness or cons investigated and collected in this research.

Strengths (pros)	Weakness (cons)
it is a rational and transactional process [265].	preexisting relationships with other stakeholders can influence their responses [234].
keeps the parties focused on few parameters [235].	large number of stakeholders can make too complex and cluttered the stakeholders mapping [234].
useful to identify assumptions in which the success or failure of project outcomes depend [234].	gives a picture that is too simple and therefore not helpful for sufficiently predicting the stakeholder's coming behavior [235].
helps when circumstances changes for a detailed understanding and communication between the stakeholder and the project representatives may be helpful [235].	identification and categorization of stakeholders can be subjective in terms of their relative power, influence, and legitimacy [236].
simple profiling is less complex and time-consuming [234].	creating a richer insight may be very time consuming [235].
help strength ties with stakeholders [236].	If the project representatives don't perceive the stakeholder analysis as helpful, they may have very limited motivation to undertake it carefully [235].

Table 4.17: Stakeholder analysis pros and cons

Strengths (pros)	Weakness (cons)
promotes participation and engagement [237].	requires time and talent to be realistic [239].
can be used to visualize the workflow and user-system interaction [171].	story can be misinterpreted or out of context [238].
facilitate communication [67].	viewers may be critical on quality and not focus on the action and environment [239].
allows stakeholders to come into common understanding of about the functionalities of the system [64].	lacks flexibility as it cannot guarantee an image is precisely relevant to a novel input story [238].
helps identify potential consequences of using the application [237], [170].	images can be extracted from different sources making storyboard sequence visually inconsistent in styles and characters [238].

Table 4.18: Story Boarding pros and cons

4.2.20 User stories

User stories are a description of a function or feature that is written from the perspective of the user [158] and reduce the focus on requirements specification activities [173]. They consist of a written text, conversation about it and acceptance criteria [158]. Also, are one of the most popular alternatives to traditional requirements specifications [262].

User stories are utilized in order to describe the functional requirements of a system from users viewpoint [171]. User stories usually have three parts: the card, conversation and confirmation. The story is written on a small index card with the second step being a conversation, because the card is small and has no space for the details, to develop understanding of the necessity. The confirmation refers to acceptance criteria which are often written on the back of the card [177]. The Story Card is a physical representation for the written text and details from a user story [158].

The Table 4.19 shows some of the strengths or pros, as well as the weakness or cons investigated and collected in this research.

Strengths (pros)	**Weakness (cons)**
effective in improving the collaboration and feedback among stakeholders and development team [39]	usually results in incomplete or ambiguous requirements [177], [186]
easily adaptable to volatile requirements [39]	shows communication and planning issues for big stories [167]
represents a way to express user needs in an informal way [171]	lacks in identifying specific user and its needs [171]
easy to estimate and prioritize [32]	lack in recognizes the environment in which the specified feature of the story is applied [171]

Table 4.19: User Stories pros and cons

4.2.21 Workshop

Workshop, project's stakeholders come together for a deliberation to gather the requirements for a system under development. Participation is limited to the stakeholders that are directly affected by the system. Workshop is also a collaborative technique and better used for collecting multiple viewpoints [65].

Workshop is a structured meeting in which a carefully selected group of stakeholders and content experts work together to define, create, refine and reach closure on deliverable that represent user requirements [240]. The Table 4.20 shows some of the strengths or pros, as well as the weakness or cons investigated and collected in this research.

Strengths (pros)	Weakness (cons)
useful to elicit requirements for complex and large systems [209]	unfeasible for small projects [209]
can bridge communication among stakeholders, decision-making, and mutual understanding [240]	require a greater time commitment from each participant; considerable cost [209]
effective way to bring together customers, users and software suppliers to improve the quality of software products [240]	very costly technique in terms of time and money [217]

Table 4.20: Workshops pros and cons

4.3 Evaluation of the Mapping - Analysis through Focus Group

This section describes the evaluation performed using the Focus Group technique. The results show evidence about the applicability of the proposed mapping and the suggestions presented.

4.3.1 Planning

The first step was the definition of the method to be used to validate the study. The decision was to use the Focus Group technique. Focus Groups is a qualitative data collection method and in-depth group interview with a moderator leading the discussion [266]. This technique helps identifying which things are important for the stakeholders [191]. For more details about this technique please refer to the section 4.2.7 that describes the advantages and disadvantages of this technique.

We understand this Focus Group there is no right or wrong, and all opinions are valid, as the objective is not to reach a consensus, but to understand and discuss the professionals' perceptions and ideas. In addition, the meeting was recorded as a way to obtain a richer detail of the discussions, and although qualitative answers are presented and reported the identities of the participants and individual opinions are kept confidential. The questions initially defined are presented in the Table 4.21.

The goal of the Focal Groups sessions is to capture the perceptions into using requirements elicitation techniques and how they could be combined to improve results.

Questions

To reach the goal, questions were asked openly, avoiding "yes" or "no" so that it is possible to compare experiences and opinions on the subject and to understand the perceptions of the invited specialists.

ID	Question
Q1	What requirements elicitation techniques do you often use and why?
Q2	What advantages do you identify in using this(these) technique(s)?
Q3	Have you combined techniques to elicit requirements? If yes, which ones and why? What advantages did you identify in combining these techniques?
Q4	This study presents the advantages and disadvantages of different techniques, in your perception how do you think this study can help the software development community? What are your suggestions for improvements?
Q5	Regarding the combinations, what do you think about these suggested combinations?
Q6	What techniques would you use to solve the hypothetical scenario described and why?

Table 4.21: Focus Group Sessions - Evaluation Questions

Since the time for the Focus Group session are scarce, for Question 5 (Q5) 4.21 we selected three combinations for the discussion. The first (Prototyping and Questionnaires) and second (User Stories and Prototyping) combinations were selected because we found the large number of references during our research while the third (Mind Mapping and Persona) was a personal selection of the researcher aiming to i) provoke the discussion of techniques not much explored in combinations and ii) verify whether non-trivial combinations would have any acceptance by the experts.

a. User Stories and Prototyping [267], [268], [269], [270], [271], [272]

b. Prototyping and Questionnaires [241], [242], [243], [101], [244], [44]

c. Mind Mapping and Persona

In addition, to address the Question 6 (Q6) (section 4.21), the participant was invited to analyze the hypothetical situation described below. This hypothetical situation was based on similar scenarios that happens in the organization that the author work.

- Legacy system existing for 10 years and the new one shall be created based on the original;
- User has been using the system for 5 years and cannot explain the functionalities (it is not known if he does not want to or does not know how);
- New system will cover the existing functionalities and new complementary functionalities, as another department with many users (more than 30) will be added to have the whole process within the same software/system;

- Motivations for the new system:
    - Improve the interface (the current one is difficult to use);
    - Security (there is none in the current one);
    - Process chain completion (activities of the other area currently in spreadsheets).

Sessions

In order to be able to accommodate all participants, we plan for a maximum of three possible sessions at different times and periods. The sessions are schedule with the duration of about 1 hour and a half so that each invited specialist can demonstrate his vision on the techniques used in requirements elicitation, as well as the possible combinations of techniques that could be done to help practitioners to have better results in the requirements elicitation phase.

It is very difficult to get the experts together at a specific time for the analysis sessions. To reduce this difficulty, before making the invitation, we contacted the participants and suggested some options for times and days, and we hold two different sessions to accommodate the participants. Sessions were also scheduled in two different days aiming in having the availability of the majority of specialists.

Once invitation was sent, even though the time was the most suitable for the vast majority, the specialists have their own responsibilities, so a long session could impact the availability and enthusiasm of the participants to participate of the debate, therefore, we opted for shorter sessions, with around one (1) hour long, to retain the specialists throughout the whole session.

Participants

Participants were selected from groups of professionals with experience in software development and requirements. Only professionals working in Agile Software Development and with Requirements were invited to the Focal Groups discussions. First Session was defined as 1 and Second Session as 2. Participants were enumerated from 1 to 19 in alphabetical order based on the session they joined. Table 4.22 presents the complete list of participants along with their current position and the years of experience they declared working with Requirements.

Participants in the first session are from different departments of the public agency that the principal investigator works for. To avoid bias, none of them are close to the researcher and were selected based on their familiarity with the topic of this study. The participants

ID	Session	Role	Level	Experience
P1	1	Project Manager and Scrum Master	Postgraduate degree in Project Management	9 years
P2	1	Project Manager	Postgraduate degree in Project Management and SW Engineering	12 years
P3	1	Requirements Engineer	Postgraduate degree in Requirements Engineering	13 years
P4	1	Project and Contract Manager	Postgraduate degree in Project Management and Public Management	+20 years
P5	1	System Analyst	Postgraduate degree in Mobile development	9 years
P6	1	Governance Coordinator	Bachelor's degree in Computing Science	8 years
P7	1	Requirements Engineer	Postgraduate degree in Distributed Software Architecture	10 years
P8	1	Developer	Bachelor's degree in Economics	10 years
P9	1	Requirements Engineer	Master Degree in Electric Engineering	19 years
P10	1	Scrum Master	Postgraduate degree in Project Management	14 years
P11	1	Business solutions Lead	Postgraduate degree in Database Management and Business Intelligence	+20 years
P12	1	Development Manager	Master Degree Electric Engineering	10 years
P13	1	Project Manager	Postgraduate degree in Project Management	5 years
P14	1	Requirements Engineer	Postgraduate degree in Software Engineering	9 years
P15	2	Senior Analyst and Developer	Master Degree in Applied Computing	18 years
P16	2	Business Analyst	Master Degree in Computing and SW Engineering	16 years
P17	2	Scrum Master	Master Degree in Computing	4 years
P18	2	Information Technology Coordinator	Master Degree in Applied Computing	8 years
P19	2	User Experience/Interface Developer	Bachelor's degree in Biotechnology	2 years

Table 4.22: Focus Group Sessions - Participants

of the second session are experienced professionals from different organizations and the relationship between them is just having the same supervisor in the doctoral degree course.

4.3.2 Execution

Each of the two sessions was initiated with the statement of its purpose and the list of techniques that would be discussed (Interviews, Prototyping, Brainstorming, Mind Mapping, User Stories, Observation, Ethnography, Questionnaires, Persona, JAD, Scenarios). Moderator also asked the confirmation whether or not it was necessary to present a brief description of any technique while presenting a short description of each of the techniques selected for the discussion. Sessions were held in Portuguese, as it was the mother language of the participants. Also, rules of engagement were stated.

- Participants during the meeting will not be anonymous, that is, you will see and hear the other people. Personal critics are not allowed.

- The moderator will be asking the questions and stimulating the discussion. The goal is for people to debate.

- Moderator can and will interfere if the discussion gets too heated, because the focus should remain only in the differences of opinions.

- There will be no transcription of individual opinions or identification of any participant during the report, so different opinions are welcomed.

- Moderator will invite participants to give their impressions as to enrich the discussion.

The debate followed with a open and provocative question from the moderator the professionals discussing the techniques they knew the ones they mostly used during their projects, also asking to explain and justify the selection of the techniques. In addition, moderator inquired which advantages the participants could identify in using the techniques they have mentioned. These were the questions 1 and 2 from the Table 4.21.

The discussion was followed by a brief discussions about combinations referenced in academia and industry in introducing the third question (Table 4.21) about combinations. In addition, moderator inquired about the Guide with question 4 (Table 4.21) by asking for the participants to freely state their opinion about the guide and which improvements could be made for future versions.

For question 5 (Table 4.21) the three previously defined combinations section 4.3.1 were presented to the participants which, again, could freely state their opinion about the validity and whether or not they were familiar (either in theory or in practice) and their opinion about these sort of technique combinations. Finally, a hypothetical scenario was presented to debate which possible techniques and combinations would be suitable for dealing with the presented situation section 4.3.1 along with their perceptions of what pros and cons they would have.

4.3.3 Reporting and results

Techniques Assessment

For the first question (Table 4.21) about the techniques used, the participants, in the first session, listed Observation, Persona, Scenarios, User Stories, Mind Map, Questionnaires, Brainstorming, Process Mapping, Prototyping and Interviews as the techniques used in the regulatory agency of Public Administration, the last three being the most used and cited during the focus group session. From their side, participants from second session mentioned Interviews, Prototyping, User Stories, Brainstorming, Persona, Observation, Test-Driven Design and Business Process Modeling. Natural Language was also mentioned as technique, although is much more a method of discussion than a technique.

For both sessions, the first and second questions Table 4.21 were answered together as participants mentioned the techniques and also the justification of the usage, meaning, the advantages they actually notice by using it. Some interesting comments are worth to highlight.

Participant P5 mentioned that Brainstorming "stimulates new solutions to the customer and is helpful for specific problems" whereas participant P3 mentioned using Brainstorming, unstructured Interviews and Prototyping where "user likes to see how the screen will look making easier to discuss how the business rules and field rules will be". This participant also mentioned that User Stories are used in the Public Administration Agency.

In addition, this participant suggested that Business Process Modeling (BPM) is one larger technique that is mostly applied in the beginning of the projects. This was also corroborated by P2, P4, P9 and P12 which informed that there is a standardized process at the Regulatory Agency that states the techniques that users are more familiar and suggest their use even though there is room for the requirements analyst to bring different techniques to enrich the discussions. He finalized informing that Business Process Modeling (BPM) is probably the technique most acceptable and used in the Agency.

Furthermore, participant P2 mentioned although they do not use Persona technique as described in literature, they do define users and system profiles during the Business Process Modeling (BPM) and continued that "Persona is an added value in other techniques". Also, when the user is not certain of what to be done they try to pull the Brainstorming technique to understand what they really need in the system and how the application should process. To facilitate, this participant uses the structure of user stories that gives advantage to write down the "focus of the system for the users" and depending on the situation, it goes on Prototyping to "try to materialize the ideas" although Prototyping "usually places when he is dealing with organizations that already have a defined interface pattern". It closed his answer by mentioning that is familiar with Ethnography and usually sees it being applied in private companies, it likes Mind Map "for consolidating ideas from User Stories" and recognizes Scenario being "heavily used in Behavioral-Driven Development (BDD)

From his side, participant P11 mentioned good experiences with User Stories (for documentation) and Prototyping which was defined as "very suitable for the development of Business Intelligence panels" which are very demanded in the organization. It is, he mentioned "by far one of the best fits" with the addition that the prototype usually comes out with a minimally functional model.

In his turn, participant P13 elected Brainstorming, "to have a basic understanding of what will be discussed" as "user has difficulties in condensing ideas" and Questionnaires to close on the details of the requirements. Also mentioned that with Prototyping "user perceives the materialization of ideas, not through writing, which is helpful to come into agreements".

Moreover, participant P6 brought a view from Governance mentioning his team is responsible for performing feasibility analysis which he compared as "something like an initial requirements survey". For his duties, his team uses a mix of Interviews with Questionnaires, with the later technique being under a "guided completion" to improve the capture of needs and giving the "advantage of not separating business from Information Technology (IT) worlds" as he highlighted that "most of the users do not know the IT world". Still, participant added that he does not use Observation Technique however, after reading the Guide for the meeting he was "interested in practicing to see the user on a daily basis" as the finalized that "meetings are just cuts/pictures of what the user does". On the other hand, since there are cases when IT allocates some analysts in the business areas, participant P12 has a sense that Observation does happen with a "low maturity and ad doc way".

Participant P4 elaborated about the standardized process of Regulatory Agency by informing it was defined in 2015 as a initial methodology that includes Prototyping, Questionnaires, Interviews all being encompassed by Business Process Modeling (BPM). Other techniques that are not suggested in the standardized process, such as, Mind Mapping and User Stories are highly used as well. P7 completed stating that the reason for the Agency to use Mind Map and Prototyping, for instance is the complexity of the systems, full of flows and much related to the niche of the organization.

Participant P14 who uses Interviews and Prototyping preferred to comment a disadvantage of the Prototyping technique mentioning “users understand that everything is ready and then get frustrated with the final version”.

Participant P15 mentioned that Interviews and User Stories or Brainstorming (“when people know the process”) are widely used in his organization with also the technique of Mind Mapping. On his turn, participant P18 cited User Stories and sometimes Persona and added that “Meeting” technique by meeting with the whole team are highly used on this organization. This participant also mentioned Test-Driven Design (TDD) and referred that based on the environment as the development team has great knowledge on the business side he did not see advantages in using Prototyping for Agile Software Development.

Moreover, participant P17 mentioned that never practiced Observation, Ethnography and Joint Application Development (JAD) while already applied Persona, although not “as pure as literature prescribes” and believes that Prototyping helps “deciding together with the client what is really necessary to implement” but, on the other hand, Prototyping is “time consuming”and sometimes “result is not viable to implement”. For such cases, it uses Brainstorming to reach consensus what is necessary in terms of interfaces and the development cost. In addition, understands that User Stories are only applicable in Agile and not in traditional methodologies. Finally, this participant mentioned that Natural Language as a method of discussion for specification is widely used in his organization as an easy way to communicate the objectives.

Participant P16 mentioned using Interviews, User Stories and Persona when dealing with a new product whereas Observation, Ethnography, Test-Driven Design (TDD) and Persona when dealing with any sort of maintenance. Furthermore, participant highlighted that the most used technique in his reality is Business Process Modeling (BPM). P19 added that Business Process Modeling (BPM) “helps structuring, refactoring and understand the flow of data”.

In terms of question Q2 (4.21), in summary, advantages of Prototyping are related to assisting client deciding what is really necessary to implement as user perceives the materialization of ideas which make easier to discuss business rules while Brainstorming helps reaching consensus on basic understanding of what will be addressed by the system with a plus of stimulating new solutions. Interviews and Questionnaires help guide users through the IT world with the later also assisting in close the details of the requirements. User Stories can maintain the focus of the system for the users and helps on documentation.

Persona helps defining users and system profiles. Mind Map simplify the complexity of the systems and helps consolidating ideas from User Stories. Observation is excellent when user has trouble to explain the requirement while Natural Language as it is easy to communicate the objectives. Finally, Business Process Modeling assists in understanding the flow of data. Also it is probably the technique most acceptable and used in the Agency (participants of session 1) also being mentioned by three of the participants of session 2.

Techniques Combination

About combinations, on question 3 (Table 4.21), the vast majority of the participants mentioned they see as impossible to not use more than one technique due organization, users and the complexity of the systems to be developed as the participant P7 added that "most systems use more than one technique to try to understand all the specifics about the systems to be created" and participant P2 mentioned that "combination is possible depending on the technicians involved". P3 also mentioned that always work with a mix of techniques to better understand the requirements. Others agreed with this information. Table 4.24 has more details of the most mentioned combinations.

Those from session 2 made a remark that Mind Mapping and Users Stories (P15) and Interview and User Stories (P15 and P18) were combinations already used and proven on their projects. Participant P18 cited User Stories and Persona as a combination already used as well. Participant P17 mentioned that User Stories, Prototyping and Brainstorming are interchangeably used.

Regarding combinations, as informed prior and at the beginning of the sessions, three combinations (List 4.3.1) were presented to be evaluated and discussed, being the first two among the most referenced in literature and the third an alternative selection of this researcher in order to verify if it would be familiar or not.

As expected, the third combination was not familiar to any of those present in both sessions, although part of the participants showed curiosity about whether such a combination would be useful and feasible in a real environment whereas for the first combination had the majority of the comments as it was recognized as the most familiar with several participants stating they use or already used such combination (User Stories and Prototyping).

Participant P11 mentioned it is the one combination he and his team uses as User Stories gives "something more formal to apply on the rigidity of the Public Administration" whereas the Prototyping provides a "faster user engagement" which was followed by participants P14, P9 and P2. In addition, participant P6 mentioned that User Stories is "a technique that uses writing from the user's perspective" while Prototyping is "more visual and give a vision of the solution"therefore both "complement each other". P3 on his turn mentioned that this combination also helps the development team and finalized informing that also used and view "as interesting" the combination Prototyping and Questionnaires.

Participant P15 made a remark that User Stories and Prototyping are “great complement” and is one combination he has the habit of using with success. Also was with great value when he had a team member experienced in front end only and the Prototyping was used to integrate this member and accelerate the project. On the other hand, participant P19 already used to work with web sites improvements where the Questionnaires were used to get the perceptions from the users while Prototyping being used to refine the user interface and experience.

Moreover, Participant P16 mentioned not being familiar with much combinations and was willing to try some options in real world as feels like it could be useful. Finally, participant P18 mentioned that he operates in a small team with small number of stakeholders, so Prototyping is not helpful on his environment therefore none of these combinations are applied on his organization however other combinations such as User Stories and Questionnaires have already been taken in some projects. P17 cited usage of User Stories and Prototyping in a “truly agile project” which as long as development went further the Prototyping was no longer necessary and was replaced by Natural Language procedure. Participant also elaborated that usually works with Interviews and Mind Mapping using Natural Language discussions or Natural Language as a way o easy on Prototyping as main options for eliciting requirements.

Guide analysis

In continuation of the debate, participants were asked about the Mapping of this study (question 4 - Table 4.21). Most relevant statement are as follows:

As for the suggestions of improvements, two items were intentionally not fully covered in this work, usage examples of the displayed techniques, mentioned in item I 4.23 and detailed and step-by-step description the each technique, mentioned in item III 4.23 due to there are some web guides on the net that can supply this information as presented in 3.16. Item II 4.23 on the other hand, refers to a recommendation system based on the findings of this study and is something to be explored in a future work.

Hypothetical scenario

Finally, for closing the session the hypothetical situation 4.3.1 of question 6 4.21 was presented and the participants were asked to discuss on which techniques or combinations of techniques they would apply to overcome the set situation. Most relevant suggestions are as follows.

ID	**Positive views**
i	“pros and cons help a lot, assess your need and identify which technique can help you” - (P2)
ii	“useful as a reference to the professional’s work” - (P3)
iii	“guide instructs a lot and leaves it up to the analyst to frame in the real case which technique to use to perform the work. Several parameters to choose one technique or another, if you have a limited time, looking at the guide it can direct you to a technique that gives a faster response.” - (P13)
iv	“very useful after environment is understood” - (P14)
v	“interesting for practitioners with experience because they are already familiar with the techniques and pros and cons help them to decide which one to use” - (P17)
vi	“good for new or inexperienced people as well” - (P18)
vii	“will definitely be used on his work” - (P16)
ID	**Suggestions of Improvements**
I	“missed some examples or case studies” - (P3)
II	“it would be interesting in a future work create some checklist or recommendation system to help engineers to select the more appropriate technique according to the environment” - (P9)
III	“inexperienced would not know the details of techniques, so might not take much advantage of the guide” - (P17)

Table 4.23: Guide - Positive and Improvements remarks

- P4 suggest using Business Process Modeling (BPM) to “identify the chain of activities and map the information for each activity” and then work with other techniques for completing the elicitation.

- P3 suggested Business Process Modeling (BPM) as well, and went further indicating Prototyping and User Stories to complete the elicitation process.

- P7 started from the premise that user only knew his specific part of the process therefore he would use Observation in combination with Mind Mapping. However if use had a complete view of the process participant would apply Business Process Modeling (BPM) technique.

- P6 believed that each item could use a different technique. Suggested that for “user doesn’t know or doesn’t want to say”afirmation, he would go with Observation or Brainstorming to help “giving ideas to the new system”. Prototyping would be used to focus on improving the interface and User Stories would be interesting to cover the piece of process.

- P15 mentioned that usually works with Business Process Modeling (BPM) for most of the scenarios however this one could be done by using Questionnaires, Interviews and Persona.
- P16 highlighted that Business Process Modeling (BPM) is the technique that could rescue the team in this sort of scenario uses Interviews as well, as he mentioned that "it is the role of requirements analyst to overcome difficulties and extract information from the user". Suggested also that User Stories could be a good complement for this proposed case.
- P18 remark that in case of "extensive documentation and access to the code" a analysis of legacy system would be beneficial.

On a final remark, participant P19 stated that for security issues usually developers and others involved in previous releases and phases of the project can be a great source for identifying security and interface issues and solutions.

Results

After analyzing all the collected data from the evaluation Focal Group sessions we reach the conclusion that the objective of the Guide was accomplished with the study being able to assist most of the practitioners either inexperienced or experienced ones taking into account that goal was to analyze strengths and weakness of the techniques and examples can be found in the references from table 3.16.

The table below 4.24 summarizes the main findings from the sessions. In relation of techniques and combinations they were organized from the most remembered and discussed by the specialists to the least ones.

From the total of nineteen (19) participants over two sessions, the techniques listed as most used Interview, Prototyping and BPM were mentioned by seven (7) participants while User Stories was mentioned by six (6).

Regarding combinations, most of the participants, pointed that User Stories and Prototyping, Brainstorming and User Stories, User Stories and Interviews along with a variety of BPM-related combinations as of their preference.

About the new findings that came from the evaluation sessions, Natural Language and Meeting were methods cited by one participant that are used as a way to easy the process of eliciting the requirements, however the most important new finding from the Focus Group sessions and something to highlight is the technique of Business Process Modelling (BPM) which was not retrieved during the research by the search string 3.1.3.

Business process modeling contributes to deeper understanding and systematization of the network of activities that composes a business [273] as a common goal for process

Subject	Descriptions
Techniques	Interviews, Prototyping, Business Process Modelling (BPM) User Stories Persona, Brainstorming Questionnaires Mind Mapping, Observation, Test-Driven Design (TDD) Scenarios, Ethnography
Combinations	Business Process Modeling and Interviews Business Process Modeling and Questionnaires Business Process Modeling and User Stories Business Process Modeling, Persona and User Stories Business Process Modeling, Brainstorming and User Stories Mind Mapping and User Stories User Stories and Interviews User Stories and Prototyping User Stories and Persona Brainstorming and User Stories Interviews and Mind Mapping
New Findings	Business Process Modelling (BPM) Meeting, Natural Language *

Table 4.24: Summary of Focus Group Evaluation Results

modeling efforts is to increase knowledge and shared awareness within the organization [274].

This technique was highly commented during the evaluation sessions including some participants informing the technique had their preference to start much of the elicitation discussions in the projects they work, therefore BPM might be a good candidate to further investigation and addition to the future versions of this guide.

4.3.4 Evaluation risks

A possible risk, inherent to the process, is the fact that some of the participants know each other therefore there is possibility of avoid exposing their opinions or entering into discussions, tending to agree with the specialists who provide the first answers. Such a situation can cause a bias in the answers. To reduce this situation, we reinforce the idea that there is no right or wrong in the answers and emphasize that we seek different opinions to oppose ideas.

Also there is a risk of putting pressure on the participants and causing them to have unnatural attitudes or behaviors. Some rules of engagement, that can be seen at section 4.3.2, were stated in an attempt to minimize this risk.

4.4 Chapter Summary

In this chapter we have presented we described some Requirements Elicitation (RE) Techniques that are common and useful for Agile Software Development (ASD) and focus on presenting strengths (pros) and weakness (cons) of these techniques also known in literature as advantages and disadvantages (or limitations) of the RE techniques.

Moreover at the beginning of the chapter we also presented an image (Figure 4.1) suggesting which of the studied techniques could assist with the uncovered and described challenge categories, which were explored in previous chapter (Section 3.3.2). The Figure 4.1 contains the list of techniques that could be applied for each category to surpass the challenge the category encompass and shows that challenges related to communication, user engagement and requirements elicitation activities are the ones best served in terms of Requirements Elicitation Techniques that could assist.

The list, combinations found either in the literature or in practical evaluations and other suggestions are the result of this study and are based on the strengths and weaknesses depicted in the previous sections of this chapter 4. Finally, we described Focus Group evaluation sessions held with selected specialists who analyzed and provide their experience about the Requirements Elicitation (RE) Techniques, combinations of techniques used in real projects, discussed a hypothetical scenario and evaluated the guide of this study. The general outcome was (i) the guide is a good reference to new and experienced practitioners, (ii) most of the specialists uses User Stories along with Interviews, Prototyping or Brainstorming to elicit their systems, (iii) Business Process Modelling (BPM) is one technique greatly accepted when dealing with new systems as they need to understand the business rules that applies. The detailed results were presented in the section Evaluation of the Mapping 4.3.

Chapter 5

Final Remarks

We initiated this study by presenting a Theoretical Background (Chapter 2) to support our discussion highlighting among others, requirements, agile software development and the elicitation techniques. In this study, a SLR was conducted to investigate the requirements elicitation techniques used by the Agile Software Development teams.

The SLR was instrumented with automatic and manual searches, performed in digital libraries, journals and conferences. Some of the studies were also Systematic Literature Reviews, therefore a snowballing was made to unfold the original studies covered by those SLR. From a total of 341, we have selected 54 papers. The objective was to identify the existing requirements elicitation techniques that are discussed for agile software development. By identifying these techniques, we sought to verify which were also mentioned as used in the industry. We also seek to investigate the challenges in ASD, as well as what are the strengths or positives and weaknesses or negatives of the identified techniques.

In addition, we conducted a survey with practitioners of agile software development, from engineers and analysts to developers and managers, to compare the results of the survey with the elements collected from the SLR. Finally, with the analysis of techniques, agile development challenges and strengths and weaknesses of the identified techniques, we analyze possible combinations of the latter that could be useful to overcome the challenges encountered and help the community and have a more complex requirements elicitation phase. robust and efficient.

In seeking to answer the RQ.1 (Section 3.3.1), the analysis revealed that the most referenced techniques in the literature are: Prototyping, Interview, User Stories, Joint Application Development (JAD), Ethnography, Brainstorming, Scenarios, Observation, Questionnaires, Mind Mapping and Workshops. On the other hand, the most referenced techniques in industry are: Interview, User Stories, Prototyping, Questionnaires, Persona, Observation, Ethnography, Mind Mapping and Scenarios.

By those lists, we verify that Prototyping, User Stories and Interview are well consolidated on both worlds, whereas Scenarios are much remembered in the literature but not so much in the industry. Also, there is much room for investigating new techniques in literature such as, Persona which have high acceptance in industry.

There are many difficulties (or challenges) in the requirements elicitation process such as communication barriers, knowledge and cognitive limitations, human behavior and a technical issue [77]. In regard of challenges, which is discussed in the RQ.2 (Section 3.3.2), we realize that grouping them into categories would make easy for the reader to recognize, therefore, based on the selected papers we identified the following categories which the table 3.13 presenting the details.

- Documentation (Doc)
- Stakeholder Engagement and Availability (Engg)
- Communication (Comm)
- Scope and Volatility of Requirements (Vol)
- Complexity and Scalability of Projects (Cplx)
- Negotiation and Prioritization of Requirements (Prio)
- Translation of Requirements (Trl)
- Expertise from team or user (Exp)
- Architecture and Software Quality (Qlty)
- Culture and Environment (Cult)
- Requirements Elicitation Work (REw)
- Project issues and Others (Oth)

Although all are relevant at some point during the Agile Development process, we focused on the ones related to Requirements Elicitation (RE) work thus we did not take into account some categories for the analysis and proposition of the combinations. Moreover, the categories of challenges "Architecture and Software Quality (Qlty)", "Culture and Environment (Cult)" and "Project issues and Others (Oth)" were not considered in the RQ.3 discussion (Section 3.3.3).

Furthermore, from this list, we identified that challenges related to "Communication", "Documentation", "Translation of Requirements" and "Stakeholder Engagement and Availability" are the ones most mentioned in the selected studies.

Based on the analysis, we presented the techniques that could help surpass the identified challenges. The Table 3.14 is the result of analysis made throughout this study where the strengths and weakness (or advantages and limitations) of the elicitation techniques were investigated and confronted with the identified challenges. Other views of the techniques that could assist in surpassing the challenges by each of the investigated elicitation techniques can be seen either in the table 3.17 or in the figure 4.1.

Complementary, we analyzed technique combinations to help overcome more than one of those challenges as discussed in the RQ.3 (Section 3.3.3). A few combinations were identified within the literature with “Prototyping and User Stories” as well as “Prototyping and Questionnaires” followed by “User Stories and Interview” being the most referenced ones.

In addition, two validation sessions were performed with most of the participants, mentioning that User Stories and Prototyping, Brainstorming and User Stories, User Stories and Interviews along with a variety of Business Process Modelling-related combinations as of their preference.

Specialists from both sessions mainly believe that combining techniques is fundamental to elicit high quality requirements and the guide would be of great value to either experienced or new practitioners to help them select the best options among Requirements Elicitation Techniques.

Although the way of each technique is applied depends on the engineer, team, customer, type of project and organization involved, the findings of this study can bring some light to the software development community by improving the knowledge either on the RE techniques or the how to use them in a combination format.

5.1 Future work

In the near future, we plan to build an online version of this mapping and perform a few studies testing the applicability and effectiveness of the requirements elicitation combinations with practitioners of agile software development. Also, once improvements are identified we plan to increment with other techniques that are currently not presented on this guide.

In addition, there are plans to perform quantitative studies focusing in build a recommendation system to help analysts and engineers to better assess their environments and receive suggestions of techniques to be used for their identified challenges. We do believe this can bring greater benefit to ASD teams.

Appendix A

Requirements Elicitation Survey Form

1. Please state the Country, State and City that you work?Por favor, informe o seu País, Estado e Cidade que você trabalha? *

.

text

2. What is your gender?Qual o seu gênero? *

Descrição

○ Female / Feminino

○ Male / Masculino

○ Non-binary / Náo binário

○ Prefer not to state / Prefere não informar

3. What is your age range?Qual sua faixa de idade? *

Descrição

○ less than 21 years / menos de 21 anos

○ 21 to 25 years / 21 a 25 anos

○ 26 to 30 years / 26 a 30 anos

○ 31 to 36 years / 31 a 36 anos

○ 37 to 42 years / 37 a 42 anos

○ 43 to 47 years / 43 a 47 anos

○ 48 to 54 years / 48 a 54 anos

○ 55 years or more / 55 anos ou mais

4. What is your level of education?Qual o seu nível de educação? *

select the maximum completed level / selecione o máximo nível completo

○ None to fundamentals grade (8th or 9th) / Nível fundamental

○ High school or secondary technical or equivalent / Ensino médio ou técnico ou equivalente

○ Bachelor's degree / Bacharel (universitário completo)

○ Master's degree / Mestrado completo

○ Doctorate degree / Doutorado completo

○ Post-doctorate degree / Pós-doutorado

5. What is your employment status?Qual a sua situação profissional? *

owners can select self-employed / proprietários podem selecionar autônomo ou auto-emprego

○ Unemployed / Desempregado

○ Self-employed or freelancer / Autônomo ou Auto-empregado

○ Part time employed / Empregado em parte do tempo

○ Full time employed / Empregado em tempo integral

○ Student / Estudante

○ Other / Outros

6. How many years of experience do you have in Software Development? Quantos anos de experiência você tem em Desenvolvimento de Software? *

any task related to analysis, development, test, science data, data base or management / qualquer tarefa relacionada com análise, desenvolvimento, teste, ciência de dados ou banco de dados ou gerenciamento.

○ Less than 1 year / menos de 1 ano

○ more than 1 to 3 years / entre 1 e 3 anos

○ more than 3 to 6 years / entre 3 e 6 anos

○ more than 6 to 10 years / entre 6 e 10 anos

○ more than 10 years / mais de 10 anos

7. Do you work or is involved with Agile Projects? Você trabalha ou está envolvido em Projetos Ágeis? *

involved = studying, researching. / envolvido = estudando, pesquisando

○ Yes / Sim

○

○ No / Não

8. What is your main area of expertise (or main role) ? Qual a sua maior área de envolvimento ou experiência? *

select the option that most fits to your working or expertise area / selecione a opção que mais de enquadra ao trabalho ou área de experiência

○ Management (manager) / Gerenciamento (Gerente ou Gestão)

○ Development (developer) / Desenvolvedor

○ Test (tester) / Testador

○ Analyst / Engineer (requirements) / Analista ou Engenheiro de requisitos

○ Database (administration or data modeler) / Banco de dados (DBA ou administrador de dados)

○ Full Stack / Desenvolvedor multifuncional ou multiplataforma

○ Others (data science, etc.) / Outros (Ciência de dados, etc.)

9. How difficult do you consider gathering or elicit the correct requirements for a project, system or application? Quão difícil você considera ser a tarefa de elicitar ou coletar requisitos corretos de um projeto, sistema ou aplicação? *

Descrição

1	2	3	4	5	6	7	8	9	10

Not at all Extremely

10. Please state which of the following tools and practices you are familiar and the level of knowledge Por favor selecione quais as práticas abaixo você tem conhecimento e qual o nível *

Please evaluate at least 5 practices or tools / Por favor, avalie pelo menos 5 práticas ou ferramentas

	Poor / Pouco ou nenhum	Average / Médio conhecimento	Good / Bom conhecimento	Excellent / Excelente
Analysis of Legacy systems / Análise de sistemas legados	□	□	□	□
Blueprint	□	□	□	□
Brainstorming	□	□	□	□
Design Thinking	□	□	□	□
Documentation or Data Analysis / Análise de dados ou documentos	□	□	□	□

Empathy Map / Mapa de empatia	□	□	□	□
Ethnography / Etnografia	□	□	□	□
Exploratory Research / Pesquisa exploratória	□	□	□	□
Feature-Driven Design	□	□	□	□
Focus Groups / Grupos focais	□	□	□	□
Interview / Entrevistas	□	□	□	□
JAD (Joint Application Development)	□	□	□	□

Laddering / Escada	□	□	□	□
Mind Mapping / mapa mental	□	□	□	□
Observation / Observação	□	□	□	□
Persona	□	□	□	□
Principal Component Analysis	□	□	□	□
Prototyping / Prototipação	□	□	□	□
Questionnaires / Questionários	□	□	□	□
QFD (Quality Function Deployment)	□	□	□	□

Scenarios / Cenários	□	□	□	□
Stakeholder Analysis / Análise de partes interessadas	□	□	□	□
Story Boarding	□	□	□	□
Test-Driven Design	□	□	□	□
Use Cases / Casos de uso	□	□	□	□
User-Centered Design	□	□	□	□

User Interface Analysis / Análise de interface do usuário	□	□	□	□
User Journey / Jornada de usuário	□	□	□	□
User Stories / Estórias de usuário	□	□	□	□
Workshop	□	□	□	□

11. What are the main challenges on requirements specification / gathering / elicitation phase ?Para você, quais são os principais desafios na fase de especificação / coleta / elicitação de requisitos? *

at least 2 challenges. In case of specific for a tool or technique, please state the name of the tool/technique / Apresente pelo menos 2 desafios, caso sejam específicos de uma técnica que conheça, informe o nome.

12. From your knowledge, please state the pros and cons of the tools, practices or techniques you most use? A partir do seu conhecimento, por favor descreva prós e contras das ferramentas, práticas ou técnicas que você mais usa.

if more than one, relate the technique to the pros and cons / Se mais de uma, relacione a técnica com os prós e contras

13. Do you use or is familiar with any technique, tool or practice not listed? Please describe the pros and cons of it. Você usa ou tem conhecimento que alguma técnica, ferramenta ou prática não listada? Por favor informe a técnicas e descreva os prós e contras.

Descrição

14. Based on your answers, is there any tool, practice ou technique that your knowledge is not good or excellent and you would like to learn and use? Baseado nas suas respostas, existe alguma técnica, ferramenta ou prática que seu conhecimento não seja bom ou excelente e que você gostaria de aprender e usar?

Descrição

15. We will send a more detailed questionnaire to some of the respondents for further developments. In case you have interest in having a more detailed discussion please state your name and e-mail. Enviaremos um questionário mais detalhado para alguns dos respondentes para devenvolvimentos futuros. Caso tenha interesse em ter uma discussão mais detalhada, por favor informe seu nome e e-mail.

Descrição

References

[1] Boehm, Barry W, John R Brown, and Myron Lipow: *Quantitative evaluation of software quality.* In *Proceedings of the 2nd international conference on Software engineering*, pages 592–605, 1976. xvii, 14

[2] Chung, Lawrence and Julio Cesar Sampaio do Prado Leite: *On Non-Functional Requirements in Software Engineering*, pages 363–379. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, ISBN 978-3-642-02463-4. `https://doi.org/10.1007/978-3-642-02463-4_19`. xvii, 13, 14

[3] Klima, Matej, Miroslav Bures, Karel Frajtak, Vaclav Rechtberger, Michal Trnka, Xavier Bellekens, Tomas Cerny, and Bestoun S. Ahmed: *Selected code-quality characteristics and metrics for internet of things systems.* IEEE Access, 10:46144–46161, 2022. xvii, 14

[4] Munassar, Nabil Mohammed Ali and A Govardhan: *A comparison between five models of software engineering.* International Journal of Computer Science Issues (IJCSI), 7(5):94, 2010. xvii, 15, 16

[5] Sarker, Iqbal H, Faisal Faruque, Ujjal Hossen, and Atikur Rahman: *A survey of software development process models in software engineering.* International Journal of Software Engineering and Its Applications, 9(11):55–70, 2015. xvii, 15, 17

[6] Wiegers, Karl E and Joy Beatty: *Software Requirements 3.* Microsoft Press, 2013. 1

[7] Ryan, Michael J. and Louis S. Wheatcraft: *On a cohesive set of requirements engineering terms.* Systems Engineering, 20(2):118–130, 2017. `https://onlinelibrary.wiley.com/doi/abs/10.1002/sys.21382`. 1

[8] Macaulay, Linda A.: *Requirements Engineering.* Springer-Verlag, Berlin, Heidelberg, 1996, ISBN 3540760067. 1, 2

[9] Song, Wenyan: *Requirement management for product-service systems: Status review and future trends.* Computers in Industry, 85:11–22, 2017, ISSN 0166-3615. `https://www.sciencedirect.com/science/article/pii/S016636151630272X`. 1

[10] Sai, Vallabhajosyula Bala Naga Venkata and Kiran Kumar Reddi: *Requirement analysis practices are the foremost way to bridge the dissonance between project success.* International Journal of Computer Science Trends and Technology (IJCST), 2017. 1

[11] Kruse, P . J.: . Anforderungen in der Systementwicklung, VDI Fortschrittberichte Reihe 20, Nr. 191, 1996, VDI Verlag GmbH, Düsseldorf, 1995. 1

[12] Aurum, Aybüke and Claes Wohlin: *Requirements engineering: setting the context.* In *Engineering and managing software requirements*, pages 1–15. Springer, 2005. 1

[13] Herzfeldt, Alexander, Robert O. Briggs, Aaron Read, and Helmut Krcmar: *Towards a taxonomy of requirements for hybrid products.* In *2011 44th Hawaii International Conference on System Sciences*, pages 1–10, 2011. 1

[14] Febriani, Elfira: *System entities approach for first step to design system that connecting small and medium enterprises (smes) and researchers.* ICASESS 2019, page 169, 2020. 1

[15] Toller, C. N. K., C. Johansson, and M. Bertoni: *Issues in capturing and utilizing non-functional requirements along the road to servitization.* Proceedings of the Design Society, 2:623–632, 2022. 1

[16] Ilin, Igor, Aleksandr Lepekhin, Anastasia Levina, and Oksana Iliashenko: *Analysis of factors, defining software development approach.* In Murgul, Vera and Zdenka Popovic (editors): *International Scientific Conference Energy Management of Municipal Transportation Facilities and Transport EMMFT 2017*, pages 1306–1314, Cham, 2018. Springer International Publishing, ISBN 978-3-319-70987-1. 1, 2

[17] Bourque, Pierre and Richard E. Fairley (editors): *SWEBOK: Guide to the Software Engineering Body of Knowledge.* IEEE Computer Society, version 3.0 edition, 2014, ISBN 978-0-7695-5166-1. `http://www.swebok.org/`. 2

[18] Hickey, Ann M. and Alan M. Davis: *Requirements elicitation and elicitation technique selection: A model for two knowledge-intensive software development process.* Proceedings of the 36th IEEE International Conference on System Sciences, 2002. 2, 7

[19] Law, Foong Li: *A tacit knowledge sharing model to enhance stakeholder engagement in requirements elicitation/Law Foong Li.* PhD thesis, Universiti Malaya, 2019. 2, 7

[20] Sadiq, Mohd and Mohd Shahid: *Elicitation and prioritization of software requirements.* International Journal of Recent Trends in Engineering, 2(3):138, 2009. 2, 19, 20

[21] Hu, Xuan and Jie Liu: *Requirements knowledge model construction and requirements elicitation method of avionics systems software based on multi-ontology.* In *2022 24th International Conference on Advanced Communication Technology (ICACT)*, pages 1265–1279, 2022. 2

[22] Venters, Colin C., Norbert Seyff, Christoph Becker, Stefanie Betz, Ruzanna Chitchyan, Leticia Duboc, Dan Mcintyre, and Birgit Penzenstadler: *Characterising sustainability requirements: A new species red herring or just an odd fish?* In *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Society Track (ICSE-SEIS)*, pages 3–12, 2017. 2

[23] Hickey, A. M., A. M. Davis, Alan M, and D. Kaiser: *Requirements elicitation techniques: Analyzing the gap between technology availability and technology use. comparative technology transfer and society.* Inf. Softw. Technol., 3(1):279–302, 2003. 2, 4, 5

[24] Elijah, Joseph, Amit Mishra, M Chukwu Udo, Abdulrahman Abdulganiyu, and Abiodun Musa: *Survey on requirement elicitation techniques: it's effect on software engineering.* Int. J. Innov. Res. Comput. Commun. Eng, 5(5):9201–9215, 2017. 2

[25] Corpino, Sabrina and Fabio Nichele: *An ilities-driven methodology for the analysis of gaps of stakeholders needs in space systems conceptual design.* IEEE Systems Journal, 11(4):2182–2191, 2016. 2

[26] Mohd Kasirun, Zarinah: *A survey on requirements elicitation practices among courseware developers.* Malaysian Journal of Computer Science, 18(1):70–77, Jun. 2005. `http://mojes.um.edu.my/index.php/MJCS/article/view/6232`. 2, 3, 4

[27] Davis, Christopher J., Robert M. Fuller, Monica Chiarini Tremblay, and Donald J. Berndt: *Communication challenges in requirements elicitation and the use of the repertory grid technique.* Journal of Computer Information Systems, 46(5):78–86, 2006. `https://doi.org/10.1080/08874417.2006.11645926`. 2

[28] Ferreira Martins, Hugo, Antônio Carvalho de Oliveira Junior, Edna Dias Canedo, Ricardo Ajax Dias Kosloski, Roberto Ávila Paldês, and Edgard Costa Oliveira: *Design thinking: Challenges for software requirements elicitation.* Information, 10(12), 2019, ISSN 2078-2489. `https://www.mdpi.com/2078-2489/10/12/371`. 2, 6

[29] Alvarez, Sebastián, Kevin Duy, Mireya Zapata, Jorge Galarza, Danilo Martinez, and Carlos Puco: *The communication between client-developer in the process of requirements elicitation for a software project.* In Rocha, Álvaro, Hojjat Adeli, Gintautas Dzemyda, Fernando Moreira, and Ana Maria Ramalho Correia (editors): *Trends and Applications in Information Systems and Technologies*, pages 36–45, Cham, 2021. Springer International Publishing, ISBN 978-3-030-72654-6. 2

[30] Cançado, Emille Catarine Rodrigues, Ian Nery Bandeira, Pedro Henrique Teixeira Costa, José Fortes, Daniel Carvalho Moreira, Luis Henrique Vieira Amaral, and Edna Dias Canedo: *Exploring user-centered requirements validation and verification techniques in a social inclusion context.* In *ICEIS (1)*, pages 85–92. SCITEPRESS, 2022. 2

[31] Li, Jinyu, Asif Ullah, Jun Li, Shah Nazir, Habib Ullah Khan, Hanif Ur Rehman, and Amin Ul Haq: *Attributes-based decision making for selection of requirement elicitation techniques using the analytic network process.* Mathematical Problems in Engineering, 2020, 2020. 2

[32] Yousuf, Masooma and M Asger: *Comparison of various requirements elicitation techniques.* International Journal of Computer Applications, 116(4), 2015. 2, 3, 7, 21, 23, 40, 44, 62, 64, 66, 68, 69, 88, 93, 94, 95, 96, 97, 99, 101

[33] Rafiq, Usman, Sohaib Shahid Bajwa, Xiaofeng Wang, and Ilaria Lunesu: *Requirements elicitation techniques applied in software startups.* In *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 141–144, 2017. 2, 5, 19

[34] Canedo, Edna Dias, Ian Nery Bandeira, Angelica Toffano Seidel Calazans, Pedro Henrique Teixeira Costa, Emille Catarine Rodrigues Cançado, and Rodrigo Bonifácio: *Privacy requirements elicitation: a systematic literature review and perception analysis of it practitioners.* Requirements Engineering, pages 1–18, 2022. 2, 7

[35] Parizi, Rafael, Matheus Prestes, Sabrina Marczak, and Tayana Conte: *How has design thinking being used and integrated into software development activities? a systematic mapping.* Journal of Systems and Software, page 111217, 2022. 2

[36] Khan^1, Shadab, Aruna B Dulloo, and Meghna Verma: *Systematic review of requirement elicitation techniques.* India, 2014. 2, 21, 23, 41, 45

[37] Sharma, Shreta and SK Pandey: *Revisiting requirements elicitation techniques.* International Journal of Computer Applications, 75(12), 2013. 2, 7, 21, 23, 41, 45

[38] Iqbal, Tabbassum and Mohammad Suaib: *Requirement elicitation technique:-a review paper.* Int. J. Comput. Math. Sci, 3(9):1–6, 2014. 2, 7, 21, 23, 40, 44

[39] Pacheco, Carla, Ivan García, and Miryam Reyes: *Requirements elicitation techniques: a systematic literature review based on the maturity of the techniques.* IET Software, 12(4):365–378, 2018. `https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-sen.2017.0144`. 2, 3, 4, 7, 22, 24, 38, 40, 44, 46, 62, 66, 68, 88, 92, 93, 96, 97, 101

[40] Mishra, Deepti, Seçil Aydin, Alok Mishra, and Sofiya Ostrovska: *Knowledge management in requirement elicitation: Situational methods view.* Computer Standards & Interfaces, 56:49–61, 2018, ISSN 0920-5489. `https://www.sciencedirect.com/science/article/pii/S092054891630229X`. 2, 6

[41] Hickey, Ann M. and Alan M. Davis: *A unified model of requirements elicitation.* Journal of Management Information Systems, 20(4):65–84, 2004. `https://doi.org/10.1080/07421222.2004.11045786`. 3, 15

[42] Tiwana, A. and M. Keil: *Functionality risk in information systems development: An empirical investigation.* IEEE Transactions on Engineering Management, 53(3):412–425, 2006. 3, 5, 6, 7

[43] Lim, Sachiko, Aron Henriksson, and Jelena Zdravkovic: *Data-Driven Requirements Elicitation: A Systematic Literature Review.* SN Computer Science, 2(1):16, January 2021, ISSN 2661-8907. `https://doi.org/10.1007/s42979-020-00416-4`. 3

[44] Muqeem, Mohd, Ahmad Sultan, Jabeen Nazeer, Md Faizan Farooqui, and Afroj Alam: *Selection of requirement elicitation techniques: A neural network based approach.* International Journal of Advanced Computer Science and Applications, 13(1), 2022. 3, 4, 66, 103

[45] Silva, Danyllo, Taisa Guidini Gonçalves, and Ana Regina C da Rocha: *A requirements engineering process for iot systems.* In *Proceedings of the XVIII Brazilian symposium on software quality*, pages 204–209, 2019. 3

[46] Souza, Anderson, Bruna Ferreira, Natasha Valentim, Lauriane Correa, Sabrina Marczak, and Tayana Conte: *Supporting the teaching of design thinking techniques for requirements elicitation through a recommendation tool.* IET Software, 14(6):693–701, 2020. `https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-sen.2019.0300`. 3, 4, 7, 88

[47] Cheng, Betty H.C. and Joanne M. Atlee: *Research directions in requirements engineering.* In *Future of Software Engineering (FOSE '07)*, pages 285–303, 2007. 3

[48] Gambo, Ishaya P, Abimbola H Soriyan, and Rhoda N Ikono: *A proposed process model for requirements engineering using delphi techniques for prioritisation.* International Journal of Information Technology and Computer Science, 1:73–80, 2015. 3

[49] Liebel, Grischa, Matthias Tichy, Eric Knauss, Oscar Ljungkrantz, and Gerald Stieglbauer: *Organisation and communication problems in automotive requirements engineering.* Requirements Engineering, 23(1):145–167, March 2018, ISSN 1432-010X. `https://doi.org/10.1007/s00766-016-0261-7`. 3

[50] Aldekhail, Maysoon, Azzedine Chikh, and Djamal Ziani: *Software requirements conflict identification: review and recommendations.* International Journal of advanced computer science and applications, 7(10), 2016. 3, 4

[51] Crowder, James A. and Curtis W. Hoff: *Categories of Requirements*, pages 139–153. Springer International Publishing, Cham, 2022, ISBN 978-3-030-91077-8. `https://doi.org/10.1007/978-3-030-91077-8_11`. 3, 11, 13, 99

[52] Freeman, R.: *The stakeholder approach revisited.* Zeitschrift für Wirtschafts- und Unternehmensethik, 5, January 2004. 3

[53] Andersson, Svante and Niclas Wikström: *Why and how are social media used in a b2b context, and which stakeholders are involved?* Journal of Business & Industrial Marketing, 2017. 3

[54] Agbo, Melletus Uchechukwu: *Effects of conflict management on organizational performance (a study of broadcasting corporation of abia state).* Journal of Humanities and Social Science, 20(6):312–329, 2020. 4

[55] Gul, Sameen, Hafsa Shareef Dar, Abdul Rehman Malik, Muhammad Zulqarnain, and Salma Imtiaz: *Gamification and gaming elements for software requirements elicitation: A systematic literature review.* International Journal on Electrical Engineering and Informatics, 13(4):931–950, 2021. 4, 5, 6

[56] Rivero, Luis, Carlos Portela, José Boaro, Pedro Santos, Venicius Rego, Geraldo Braz Junior, Anselmo Paiva, Erika Alves, Milton Oliveira, Renato Moraes, *et al.*: *Lessons learned from applying requirements and design techniques in the development of a machine learning system for predicting lawsuits against power companies.* In *International Conference on Human-Computer Interaction*, pages 227–243. Springer, 2021. 4

[57] Gobov, Denys and Inna Huchenko: *Software requirements elicitation techniques selection method for the project scope management.* In *ITPM*, pages 1–10, 2021. 4

[58] Gupta, Ashok Kumar and Aziz Deraman: *Algorithmic solution for effective selection of elicitation techniques.* In *2019 International Conference on Computer and Information Sciences (ICCIS)*, pages 1–7, 2019. 4

[59] Davis, A. and A. Hickey: *Requirements Researchers: Do We Practice What We Preach*, chapter 30, pages 199–206. John Wiley & Sons, Ltd, 2015, ISBN 9781119134657. `https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119134657.ch30`. 4

[60] Al-Zawahreh, Hanan and Khaled Almakadmeh: *Procedural model of requirements elicitation techniques.* In *Procedural Model of Requirements Elicitation Techniques*, New York, NY, USA, 2015. Association for Computing Machinery, ISBN 9781450334587. `https://doi.org/10.1145/2816839.2816902`. 4, 40, 44, 48, 53, 55, 62, 64, 76, 93

[61] Hussein, Ibrahim Hassan, Jamilah Din, Salmi Baharom, and Muhammed Basheer Jasser: *An approach for selecting the suitable requirement elicitation technique.* Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(3):2083–2087, 2021. 4

[62] Paiva, Ana CR, Ana Rosa Cavalli, Paula Ventura Martins, and Ricardo Pérez-Castillo: *Quality of Information and Communications Technology: 14th International Conference, QUATIC 2021, Algarve, Portugal, September 8–11, 2021, Proceedings*, volume 1439. Springer Nature, 2021. 4, 6

[63] Pereira, Lauriane, Rafael Parizi, Matheus Prestes, Sabrina Marczak, and Tayana Conte: *Towards an understanding of benefits and challenges in the use of design thinking in requirements engineering.* In *Towards an Understanding of Benefits and Challenges in the Use of Design Thinking in Requirements Engineering*, SAC '21, page 1338–1345, New York, NY, USA, 2021. Association for Computing Machinery, ISBN 9781450381048. `https://doi.org/10.1145/3412841.3442008`. 4

[64] Tiwari, Saurabh and Santosh Singh Rathore: *A methodology for the selection of requirement elicitation techniques.* CoRR, abs/1709.08481, 2017. `http://arxiv.org/abs/1709.08481`. 4, 99, 100

[65] Okesola, Olatunji, Kennedy Okokpujie, Rowland Goddy-Worlu, Afolakemi Ogunbanwo, and Olamma Iheanetu: *Qualitative comparisons of elicitation techniques in requirement engineering.* Journal of Engineering and Applied Sciences, 14:565–570, January 2019. 4, 40, 44, 62, 88, 89, 90, 92, 93, 96, 98, 99, 101

[66] Aleryani, Arwa Y: *The impact of the user experience (ux) on the quality of the requirements elicitation.* International Journal of Digital Information and Wireless Communications, 10(1):1–9, 2020. 4

[67] Zhang, Zheying: *Effective requirements development–a comparison of requirements elicitation techniques.* British Computer Society, January 2007. 4, 6, 7, 40, 44, 62, 95, 100

[68] Joseph, Elijah: *Survey on requirement elicitation techniques: It's effect on software engineering.* International Journal of Innovative Research in Computer and Communication Engineering, 5:9201–9215, May 2017. 5

[69] Spoletini, Paola and Alessio Ferrari: *Requirements elicitation: A look at the future through the lenses of the past.* In *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pages 476–477, 2017. 5, 7, 19

[70] various: *Standish chaos report 2016.* Expert Panel 2020, 2016. `www.standish/reports/reports.php`. 5

[71] Ahmad, Nauman: *Effects of Electronic Communication on the Elicitation of Tacit Knowledge in Interview Techniques for Small Software Developments.* PhD thesis, University of Huddersfield, 2021. 5

[72] various: *14 common reasons software projects fail (and how to avoid them).* Expert Panel 2020, 2022. `https://www.forbes.com/sites/forbestechcouncil/2020/03/31/14-common-reasons-software-projects-fail-and-how-to-avoid-them/?sh=3887c898798c`. 5

[73] Davey, Bill and Kevin Parker: *Requirements elicitation problems: A literature analysis.* Issues in Informing Science and Information Technology, 12, January 2015. 5

[74] Robinson, Fred Y.: *Underestimation of jingle fallacy influences on systems acquisitions.* Systems Research and Behavioral Science, 38(4):413–425, 2021. `https://onlinelibrary.wiley.com/doi/abs/10.1002/sres.2672`. 5

[75] Chaudhary, Moazzama Nadeem Ahmad, Nosheen Sabahat, and Saba Khalil Toor: *Res-tool to overcome requirements elicitation barriers in pakistan software industry.* In *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pages 1–6, 2020. 5

[76] Garg, Nikita, Pankaj Agarwal, and Shadab Khan: *Recent advancements in requirement elicitation and prioritization techniques.* In *2015 International Conference on Advances in Computer Engineering and Applications*, pages 237–240, 2015. 6

[77] Jakkaew, Prasara and Tew Hongthong: *Requirements elicitation to develop mobile application for elderly.* In *2017 International Conference on Digital Arts, Media and Technology (ICDAMT)*, pages 464–467, 2017. 6, 62, 96, 117

[78] Ali, Zahid, Muhammad Yaseen, and Salman Ahmed: *Effective communication as critical success factor during requirement elicitation in global software development.* International Journal of Computer Science Engineering (IJCSE), 8(3):108–115, 2019. 6

[79] Coughlan, Jane and Robert D. Macredie: *Effective Communication in Requirements Elicitation: A Comparison of Methodologies.* Requirements Engineering, 7(2):47–60, June 2002, ISSN 1432-010X. `https://doi.org/10.1007/s007660200004`. 6

[80] RAMIREZ, ITZEL MORALES: *Requirements analysis skills: How to train practitioners?* In *The 8th International Workshop on Requirements Engineering Education and Training (REET'18)*, 2018. 6

[81] Tooptompong, Piyanush and Krerk Piromsopa: *Factor analysis of the functional requirement of smes information systems.* 4th International Conference on Business Finance and Management on 13th, 2017. 6

[82] Darlington, M and S.J. Culley: *Current research in the engineering design requirement.* Proceedings of The Institution of Mechanical Engineers Part B-journal of Engineering Manufacture - PROC INST MECH ENG B-J ENG MA, 216:375–388, March 2002. 6

[83] Ali, Naveed and Richard Lai: *A method of requirements elicitation and analysis for global software development.* Journal of Software: Evolution and Process, 29(4):e1830, 2017. 6

[84] Ahmad, Nauman: *Effects of electronic communication on the elicitation of tacit knowledge in interview techniques for small software developments.* `http://eprints.hud.ac.uk/id/eprint/35469/`, ., February 2021. 6

[85] Eltigani, Rasha: *The stakeholder-profile framework for tacit knowledge acquisition in requirements elicitation interviews.* ., May 2022. 6

[86] Siakas, Errikos, Harjinder Rahanu, Elli Georgiadou, and Kerstin Siakas: *Towards reducing communication gaps in multicultural and global requirements elicitation.* In *European Conference on Software Process Improvement*, pages 257–277. Springer, 2021. 6

[87] Kujala, Sari: *User studies : a practical approach to user involvement for gathering user needs and requirements.* Doctoral thesis, ., 2002, ISBN 951-22-5900-1. 6

[88] Jones, Capers: *Strategies for managing requirements creep.* Computer, 29:92 – 94, July 1996. 6

[89] Singh, MP and Rajnish Vyas: *Requirements volatility in software development process.* International Journal of Soft Computing, 2(4):259–264, 2012. 6

[90] Vaidyanathan, Srinivasan and Priya B. Lakshmi: *Challenges in developing software in today's scenario: An analysis at developmental stage level.* Confluence of AI, Machine, and Deep Learning in Cyber Forensics, 10(1):199–222, 2021. 6

[91] Nurmuliani, Nur, Didar Zowghi, and S Powell: *Analysis of requirements volatility during software development life cycle.* In *2004 Australian Software Engineering Conference. Proceedings.*, pages 28–37. IEEE, 2004. 6

[92] Dasanayake, Sandun, Sanja Aaramaa, Jouni Markkula, and Markku Oivo: *Impact of requirements volatility on software architecture: How do software teams keep up with ever-changing requirements?* Journal of software : evolution and process, 31(6):n/a, 2019, ISSN 2047-7473. 6

[93] Jayatilleke, Shalinka and Richard Lai: *A systematic review of requirements change management.* Information and Software Technology, 93:163–185, 2018, ISSN 0950-5849. `https://www.sciencedirect.com/science/article/pii/S0950584917304664`. 6

[94] Nurmuliani, Nur, Didar Zowghi, and S Williams: *Requirements volatility & its impact on change effort: Evidence based research n software development projects.* In *Verified OK*. University of South Australia, 2006. 6

[95] AlSanad, Abeer and Azeddine Chikh: *The impact of software requirement change – a review.* In Rocha, Alvaro, Ana Maria Correia, Sandra Costanzo, and Luis Paulo Reis (editors): *New Contributions in Information Systems and Technologies*, pages 803–812. Springer International Publishing, 2015, ISBN 978-3-319-16486-1. 6

[96] Malaiya, Y.K. and J. Denton: *Requirements volatility and defect density.* In *Proceedings 10th International Symposium on Software Reliability Engineering (Cat. No.PR00443)*, pages 285–294, 1999. 6

[97] Ganna, Anil and Sri Sai Ripughna Rishitosh Sonti: *Analysis of requirements volatility in elicitation process: A systematic literature review & survey.* ., 2020. 6

[98] Carreteiro, Paulo, José Braga de Vasconcelos, Alexandre Barão, and Álvaro Rocha: *A knowledge management approach for software engineering projects development.* In Rocha, Álvaro, Ana Maria Correia, Hojjat Adeli, Luis Paulo Reis, and Marcelo Mendonça Teixeira (editors): *New Advances in Information Systems and Technologies*, pages 59–68. Springer International Publishing, 2016, ISBN 978-3-319-31232-3. 6

[99] Zowghi, Didar and Chad Coulin: *Requirements Elicitation: A Survey of Techniques, Approaches, and Tools*, pages 19–46. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, ISBN 978-3-540-28244-0. `https://doi.org/10.1007/3-540-28244-0_2`. 7, 92

[100] Carrizo, Dante, Oscar Dieste, and Natalia Juristo: *Systematizing requirements elicitation technique selection.* Information and Software Technology, 56(6):644–669, 2014, ISSN 0950-5849. `https://www.sciencedirect.com/science/article/pii/S0950584914000202`. 7

[101] Sutcliffe, Alistair and Pete Sawyer: *Requirements elicitation: Towards the unknown unknowns.* In *2013 21st IEEE International Requirements Engineering Conference (RE)*, pages 92–104. IEEE, 2013. 7, 21, 23, 40, 44, 46, 62, 66, 68, 90, 95, 99, 103

[102] Li, Jinyu, Asif Ullah, Jun Li, Shah Nazir, Habib Ullah Khan, Hanif Ur Rehman, and Amin Ul Haq: *Attributes-based decision making for selection of requirement elicitation techniques using the analytic network process.* Mathematical Problems in Engineering, 2020:2156023, September 2020, ISSN 1024-123X. `https://doi.org/10.1155/2020/2156023`, Publisher: Hindawi. 7, 22, 24

[103] Saad, Aslina and Christian Dawson: *Journal of systems and information technology.* ., 2018. 7

[104] Dave, Dev, Angelica Celestino, Aparna S. Varde, and Vaibhav Anu: *Management of implicit requirements data in large srs documents: Taxonomy and techniques.* SIGMOD Rec., 51(2):18–29, jul 2022, ISSN 0163-5808. `https://doi.org/10.1145/3552490.3552494`. 7

[105] Snyder, Hannah: *Literature review as a research methodology: An overview and guidelines.* Journal of Business Research, 104:333–339, 2019, ISSN 0148-2963. `https://www.sciencedirect.com/science/article/pii/S0148296319304564`. 9

[106] Kitchenham, B. and S Charters: *Guidelines for performing systematic literature reviews in software engineering*, 2007. 9, 26, 27, 30, 32, 33, 34, 82

[107] Templier, Mathieu and Guy Paré: *A framework for guiding and evaluating literature reviews.* Communications of the Association for Information Systems, 37(1):6, 2015. 9

[108] Heeager, Lise Tordrup and Peter Axel Nielsen: *A conceptual model of agile software development in a safety-critical context: A systematic literature review.* Information and Software Technology, 103:22–39, 2018. 9

[109] Saeeda, Hina, Junyu Dong, Yong Wang, and Muhammmad Adil Abid: *A proposed framework for improved software requirements elicitation process in scrum: Implementation by a real-life norway-based it project.* Journal of Software: Evolution and Process, 32(7):e2247, 2020. `https://onlinelibrary.wiley.com/doi/abs/10.1002/smr.2247`, e2247 JSME-19-0129.R1. 11, 12, 13, 17, 18, 20, 39, 43, 50, 55, 62, 64, 69, 76, 94, 96

[110] Aurum, Aybüke and Claes Wohlin: *Aligning requirements with business objectives: a framework for requirements engineering decisions.* ., August 2022. 11

[111] Crowder, James A and Curtis W Hoff: *Mdre: Systems engineering and requirements.* In *Requirements Engineering: Laying a Firm Foundation*, pages 85–97. Springer, 2022. 11

[112] Harwell, Richard, Erik Aslaksen, Roy Mengot, Ivy Hooks, and Ken Ptack: *What is a requirement?* INCOSE International Symposium, 3(1):17–24, 1993. 11

[113] .: *Ieee standard glossary of software engineering terminology.* IEEE Std 610.12-1990, pages 1–84, 1990. 11

[114] .: *Iso/iec/ieee international standard - systems and software engineering–vocabulary.* ISO/IEC/IEEE 24765:2017(E), pages 1–541, 2017. 12

[115] .: *Iso/iec/ieee international standard - systems and software engineering–vocabulary.* ISO/IEC/IEEE 29148:2018(E), pages 1–541, 2018. https://www.iso.org/obp/ui/#iso:std:iso-iec-ieee:29148:ed-2:v1:en? 12

[116] Alsaqaf, Wasim, Maya Daneva, and Roel J. Wieringa: *Agile quality requirements engineering challenges: First results from a case study.* In *ESEM*, pages 454–459. IEEE Computer Society, 2017. 13

[117] Kotonya, G. and I. Sommerville: *Integrating safety analysis and requirements engineering.* In *Proceedings of Joint 4th International Computer Science Conference and 4th Asia Pacific Software Engineering Conference*, pages 259–271, 1997. 13, 15

[118] Glinz, Martin: *On non-functional requirements.* In *15th IEEE International Requirements Engineering Conference (RE 2007)*, pages 21–26, 2007. 13

[119] Pohl, Klaus: *Requirements engineering fundamentals: a study guide for the certified professional for requirements engineering exam-foundation level-IREB compliant.* Rocky Nook, Inc., 2016. 13

[120] .: *Iso/iec/ieee international standard - systems and software engineering–vocabulary.* ISO/IEC/IEEE 24765:2017(E), pages 1–541, 2017. 13

[121] Habibullah, Khan Mohammad and Jennifer Horkoff: *Non-functional requirements for machine learning: Understanding current use and challenges in industry.* In *2021 IEEE 29th International Requirements Engineering Conference (RE)*, pages 13–23, 2021. 13

[122] Fernandes, João M. and Ricardo J. Machado: *Requirements Elicitation*, pages 85–117. Springer International Publishing, Cham, 2016, ISBN 978-3-319-18597-2. https://doi.org/10.1007/978-3-319-18597-2_5. 13, 41, 45, 62, 67, 95, 96

[123] Dias Canedo, Edna and Bruno Cordeiro Mendes: *Software requirements classification using machine learning algorithms.* Entropy, 22(9):1057, 2020. 13

[124] Galster, Matthias and Eva Bucherer: *A taxonomy for identifying and specifying non-functional requirements in service-oriented development.* In *2008 IEEE Congress on Services - Part I*, pages 345–352, 2008. 13

[125] Purnamasari, Fanindia and Surya Hardi: *A study on usability requirement for redesigning student information system.* In *2019 3rd International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM)*, pages 145–148, 2019. 15

[126] Mateen, Ahmed, Khizar Abbas, and Muhammad Azeem Akbar: *Robust approaches, techniques and tools for requirement engineering in agile development.* In *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, pages 100–103. IEEE, 2017. 15, 20, 39, 43, 46, 62, 85, 93

[127] Ruparelia, Nayan B.: *Software development lifecycle models.* SIGSOFT Softw. Eng. Notes, 35(3):8–13, may 2010, ISSN 0163-5948. https://doi.org/10.1145/1764810.1764814. 15, 16, 40, 44, 62, 93

[128] Balaji, Sundramoorthy and M Sundararajan Murugaiyan: *Waterfall vs. v-model vs. agile: A comparative study on sdlc.* International Journal of Information Technology and Business Management, 2(1):26–30, 2012. 15

[129] Alshamrani, Adel and Abdullah Bahattab: *A comparison between three sdlc models waterfall model, spiral model, and incremental/iterative model.* International Journal of Computer Science Issues (IJCSI), 12(1):106, 2015. 15, 16, 17

[130] Doshi, Dhruv, Labdhi Jain, and Kunj Gala: *Review of the spiral model and its applications.* International Journal of Engineering Applied Sciences and Technology, 5(12):2455–2143, 2021. 15

[131] Fowler, Martin, Jim Highsmith, *et al.*: *The agile manifesto.* Software development, 9(8):28–35, 2001. 15, 18

[132] Beck, Kent, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, *et al.*: *The agile manifesto*, 2001. 15, 18

[133] Dingsøyr, Torgeir, Sridhar Nerur, VenuGopal Balijepally, and Nils Brede Moe: *A decade of agile methodologies: Towards explaining agile software development.* Journal of Systems and Software, 85(6):1213–1221, 2012, ISSN 0164-1212. `https://www.sciencedirect.com/science/article/pii/S0164121212000532`, Special Issue: Agile Development. 15, 20

[134] Aldave, Ainhoa, Juan M Vara, David Granada, and Esperanza Marcos: *Leveraging creativity in requirements elicitation within agile software development: A systematic literature review.* Journal of Systems and Software, 157:110396, 2019. 15, 19

[135] Sommerville, Ian: *Software engineering. 10th.* In *Book Software Engineering. 10th, Series Software Engineering.* Addison-Wesley, 2015. 15, 20

[136] Govil, Nikhil and Ashish Sharma: *Information extraction on requirement prioritization approaches in agile software development processes.* In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pages 1097–1100, 2021. 17, 20, 39, 43, 46, 62, 89, 97, 98

[137] Kumar, Gaurav and Pradeep Kumar Bhatia: *Impact of agile methodology on software development process.* International Journal of Computer Technology and Electronics Engineering (IJCTEE), 2(4):46–50, 2012. 18

[138] Rodríguez, Pilar, Jouni Markkula, Markku Oivo, and Kimmo Turula: *Survey on agile and lean usage in finish software industry.* In *Proceedings of the 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 139–148, 2012. 18

[139] Matharu, Gurpreet Singh, Anju Mishra, Harmeet Singh, and Priyanka Upadhyay: *Empirical study of agile software development methodologies: A comparative analysis.* SIGSOFT Softw. Eng. Notes, 40(1):1–6, feb 2015, ISSN 0163-5948. `https://doi.org/10.1145/2693208.2693233`. 18

[140] .: *15th state of agile report, tech. rep.* Digital.ai Software, 2021. https://digital.ai/resource-center/analyst-reports/state-of-agile-report. 18

[141] Read, Aaron and Robert Briggs: *The many lives of an agile story: Design processes, design products, and understandings in a large-scale agile development project.* Hawaii International Conference on System Sciences, 0:5319–5328, January 2012. 19

[142] Fægri, Tor Erlend and Nils Brede Moe: *Re-conceptualizing requirements engineering: findings from a large-scale, agile project.* In *Scientific Workshop Proceedings of the XP2015*, pages 1–5, 2015. 19

[143] Bjarnason, Elizabeth, Michael Unterkalmsteiner, MARKUS BORG, and Emelie Engström: *A multi-case study of agile requirements engineering and the use of test cases as requirements.* Information and Software Technology, 77:61–79, 2016, ISSN 0950-5849. 19, 20

[144] Yu, Xiaodan and Stacie Petter: *Understanding agile software development practices using shared mental models theory.* Information and Software Technology, 56(8):911–921, 2014, ISSN 0950-5849. https://www.sciencedirect.com/science/article/pii/S0950584914000524. 19

[145] Curcio, Karina, T. Navarro, A. Malucelli, and S. Reinehr: *Requirements engineering: A systematic mapping study in agile software development.* J. Syst. Softw., 139:32–50, 2018. 19, 38, 40, 44

[146] Meligy, Ali, Walid Dabour, and Alaa Farhat: *The role of ethnography in agile requirements analysis.* In *Proceedings of the 7th International Conference on Software and Information Engineering*, ICSIE '18, page 27–31, New York, NY, USA, 2018. Association for Computing Machinery, ISBN 9781450364690. https://doi.org/10.1145/3220267.3220273. 19, 21, 39, 43, 64, 94

[147] Ewusi-Mensah, Kweku: *Software development failures.* Mit Press, 2003. 20

[148] Abad, Zahra Shakeri Hossein, Mohammad Noaeen, and Guenther Ruhe: *Requirements engineering visualization: a systematic literature review.* In *2016 IEEE 24th International Requirements Engineering Conference (RE)*, pages 6–15. IEEE, 2016. 20

[149] Weichbroth, Paweł and Marcin Sikorski: *User interface prototyping. techniques, methods and tools.* Studia Ekonomiczne, 1(234):184–198, 2015. 20

[150] Jarzębowicz, Aleksander and Paweł Weichbroth: *A qualitative study on non-functional requirements in agile software development.* IEEE Access, 9:40458–40475, 2021. 20, 22, 25, 38, 39, 43, 50, 55, 76, 81

[151] Thayer, Richard H. and Merlin Dorfman: *System and Software Requirements Engineering.* IEEE Computer Society Press, Washington, DC, USA, 1995, ISBN 0818689218. 20

[152] Kotonya, G and I Sommerville: *I (1998)"requirements engineering—processes and techniques"*. 20

[153] Cockburn, Alistair: *Agile software development joins the" would-be" crowd.* Cutter IT Journal, 15(1):6–12, 2002. 20

[154] Coutinho, Jarbele CS, Wilkerson L Andrade, and Patrícia DL Machado: *Requirements engineering and software testing in agile methodologies: A systematic mapping.* In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering*, pages 322–331, 2019. 20

[155] Medeiros, Juliana, Alexandre ML de Vasconcelos, and Carla Silva: *Integration of agile practices: An approach to improve the quality of software specifications.* In *CIbSE*, page 859, 2015. 20, 39, 51, 76

[156] Sadiq, Mohd. and V. Susheela Devi: *A rough-set based approach for the prioritization of software requirements.* International Journal of Information Technology, 14(1):447–457, February 2022, ISSN 2511-2112. `https://doi.org/10.1007/s41870-021-00749-0`. 20

[157] Inayat, Irum, Siti Salwah Salim, Sabrina Marczak, Maya Daneva, and Shahaboddin Shamshirband: *A systematic literature review on agile requirements engineering practices and challenges.* Computers in human behavior, 51:915–929, 2015. 21, 23

[158] Schön, Eva Maria, Dominique Winter, María José Escalona, and Jörg Thomaschewski: *Key challenges in agile requirements engineering.* In *International Conference on Agile Software Development*, pages 37–51. Springer, Cham, 2017. 22, 24, 40, 53, 55, 62, 67, 76, 88, 94, 95, 100, 101

[159] Zamudio, Lizbeth, José Alfonso Aguilar, Carolina Tripp, and Sanjay Misra: *A requirements engineering techniques review in agile software development methods.* In *International Conference on Computational Science and Its Applications*, pages 683–698. Springer, 2017. 22, 24, 38, 40, 44, 52, 55, 62, 76, 88, 96, 97

[160] Alsaqaf, Wasim, Maya Daneva, and Roel Wieringa: *Quality requirements challenges in the context of large-scale distributed agile: An empirical study.* Information and software technology, 110:39–55, 2019. 22, 24, 38, 40, 51, 55, 76

[161] Palomares, Cristina, Xavier Franch, Carme Quer, Panagiota Chatzipetrou, Lidia López, and Tony Gorschek: *The state-of-practice in requirements elicitation: an extended interview study at 12 companies.* Requirements Engineering, 26(2):273–299, 2021. 22, 24

[162] Anwar, Hamnah, Saif Ur Rehman Khan, Javed Iqbal, and Adnan Akhunzada: *A tacit-knowledge-based requirements elicitation model supporting covid-19 context.* IEEE Access, 10:24481–24508, 2022. 22, 25

[163] Mayer, André Henrique, Cristiano André da Costa, and Rodrigo da Rosa Righi: *Electronic health records in a blockchain: A systematic review.* Health Informatics Journal, 26(2):1273–1288, 2020. `https://doi.org/10.1177/1460458219866350`, PMID: 31566472. 26

[164] Zhang, He, Muhammad Ali Babar, and Paolo Tell: *Identifying relevant studies in software engineering.* Inf. Softw. Technol., 53:625–637, 2011. 27, 62, 69, 95

[165] Petersen, Kai, Sairam Vakkalanka, and Ludwik Kuzniarz: *Guidelines for conducting systematic mapping studies in software engineering: An update.* Information and Software Technology, 64:1–18, 2015, ISSN 0950-5849. `https://www.sciencedirect.com/science/article/pii/S0950584915000646`. 27

[166] Brereton, Pearl, Barbara A. Kitchenham, David Budgen, Mark Turner, and Mohamed Khalil: *Lessons from applying the systematic literature review process within the software engineering domain.* Journal of Systems and Software, 80(4):571–583, 2007, ISSN 0164-1212. `https://www.sciencedirect.com/science/article/pii/S016412120600197X`, Software Performance. 28

[167] Schön, Eva Maria, Jörg Thomaschewski, and María José Escalona: *Agile requirements engineering: A systematic literature review.* Computer Standards & Interfaces, 49:79–91, January 2017. `https://doi.org/10.1016/j.csi.2016.08.011`. 38, 39, 43, 46, 50, 62, 76, 96, 98, 101

[168] Jarzębowicz, Aleksander and Paweł Weichbroth: *A systematic literature review on implementing non-functional requirements in agile software development: Issues and facilitating practices.* In *International Conference on Lean and Agile Software Development*, pages 91–110. Springer, 2021. 38, 40, 53, 76

[169] Medeiros, Juliana, Alexandre Vasconcelos, Carla Silva, and Miguel Goulão: *Requirements specification for developers in agile projects: Evaluation by two industrial case studies.* Information and Software Technology, 117:106194, 2020. 39, 43, 62, 95

[170] Palacin-Silva, Maria, Jayden Khakurel, Ari Happonen, Timo Hynninen, and Jari Porras: *Infusing design thinking into a software engineering capstone course.* In *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T)*, pages 212–221. IEEE, 2017. 39, 43, 48, 62, 95, 96, 99, 100

[171] Sedeño, Jorge, Eva Maria Schön, Carlos Joaquín Torrecilla Salinas, Jörg Thomaschewski, María José Escalona Cuaresma, and Manuel Mejías Risoto: *Modelling agile requirements using context-based persona stories.* In *WEBIST 2017: 13th International Conference on Web Information Systems and Technologies (2017), p 196-203.* ScitePress Digital Library, 2017. 39, 43, 62, 68, 95, 96, 99, 100, 101

[172] Magües, Daniel A, John W Castro, and Silvia T Acuna: *Hci usability techniques in agile development.* In *2016 IEEE International Conference on Automatica (ICA-ACCA)*, pages 1–7. IEEE, 2016. 39, 43, 48

[173] Mendes, Thiago Souto, Mário André de F. Farias, Manoel Mendonça, Henrique Frota Soares, Marcos Kalinowski, and Rodrigo Oliveira Spínola: *Impacts of agile requirements documentation debt on software projects: a retrospective study.* In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pages 1290–1295, 2016. 39, 43, 48, 50, 55, 76, 100

[174] Soares, Henrique F, Nicolli SR Alves, Thiago S Mendes, Manoel Mendonça, and Rodrigo O Spinola: *Investigating the link between user stories and documentation debt on software projects.* In *2015 12th International Conference on Information Technology-New Generations*, pages 385–390. IEEE, 2015. 39, 48, 50, 55, 76

[175] Wanderley, Fernando, Denis Silva da Silveira, João Araujo, and Maria Lencastre: *Generating feature model from creative requirements using model driven design.* In *Proceedings of the 16th International Software Product Line Conference-Volume 2*, pages 18–25, 2012. 39, 43, 48, 62, 67, 93, 94

[176] Butt, Saad Masood and Wan Fatimah Wan Ahmad: *Handling requirements using flexreq model.* In *2012 IEEE International Conference on Computer Science and Automation Engineering*, pages 661–664. IEEE, 2012. 39, 43, 51, 55, 76

[177] Abdullah, Nik Nailah Binti, Shinichi Honiden, Helen Sharp, Bashar Nuseibeh, and David Notkin: *Communication patterns of agile requirements engineering.* In *Proceedings of the 1st workshop on agile requirements engineering*, pages 1–4, 2011. 40, 43, 62, 101

[178] Borg, Rodrick and Martin Kropp: *Automated acceptance test refactoring.* In *Proceedings of the 4th Workshop on Refactoring Tools*, pages 15–21, 2011. 40, 43

[179] Chand, M Gopi, K Narendar Reddy, A Ananda Rao, and J Kiran Kumar: *An approach to requirements elicitation and analysis using goal.* In *2010 2nd International Conference on Software Technology and Engineering*, volume 1, pages V1–218. IEEE, 2010. 40, 43, 46

[180] Medeiros, Juliana, Alexandre Vasconcelos, Carla Silva, and Miguel Goulão: *Quality of software requirements specification in agile projects: A cross-case analysis of six companies.* Journal of Systems and Software, 142:171–194, 2018. 40, 44, 46, 48

[181] Gregory, Peggy, Leonor Barroca, Helen Sharp, Advait Deshpande, and Katie Taylor: *The challenges that challenge: Engaging with agile practitioners' concerns.* Information and Software Technology, 77:92–104, 2016. 40, 48, 52, 76

[182] Hess, Anne, Philipp Diebold, and Norbert Seyff: *Understanding information needs of agile teams to improve requirements communication.* Journal of Industrial Information Integration, 14:3–15, 2019. 40, 44, 48, 52, 55, 76

[183] Mahmud, Imran and Vito Veneziano: *Mind-mapping: An effective technique to facilitate requirements engineering in agile software development.* In *14th International Conference on Computer and Information Technology (ICCIT 2011)*, pages 157–162. IEEE, 2011. 40, 44, 48, 94

[184] Bjarnason, Elizabeth, Franz Lang, and Alexander Mjöberg: *A model of software prototyping based on a systematic map.* In *Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 1–11, 2021. 40, 44, 48, 62, 96

[185] Rodriguez, Sebastian, John Thangarajah, and Michael Winikoff: *User and system stories: an agile approach for managing requirements in aose.* In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1064–1072, 2021. 40, 44, 48

[186] Amna, Anis R and Geert Poels: *Ambiguity in user stories: A systematic literature review.* Information and Software Technology, 145:106824, 2022. 41, 45, 62, 101

[187] Mohagheghi, Parastoo and Mario Ek Aparicio: *An industry experience report on managing product quality requirements in a large organization.* Information and Software Technology, 88:96–109, 2017. 41, 53, 55

[188] Dingsøyr, Torgeir, Nils Brede Moe, Tor Erlend Fægri, and Eva Amdahl Seim: *Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation.* Empirical Software Engineering, 23:490–520, 2018. 41, 53, 55

[189] Bjarnason, Elizabeth, Krzysztof Wnuk, and Björn Regnell: *A case study on benefits and side-effects of agile practices in large-scale requirements engineering.* In *proceedings of the 1st workshop on agile requirements engineering*, pages 1–5, 2011. 41, 54, 55

[190] Käpyaho, Marja and Marjo Kauppinen: *Agile requirements engineering with prototyping: A case study.* In *2015 IEEE 23rd International requirements engineering conference (RE)*, pages 334–343. IEEE, 2015. 41, 45, 48, 54, 55

[191] Younas, M, DNA Jawawi, I Ghani, and R Kazmi: *Non-functional requirements elicitation guideline for agile methods.* Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 9(3-4):137–142, 2017. 41, 45, 48, 89, 102

[192] Lombriser, Philipp, Fabiano Dalpiaz, Garm Lucassen, and Sjaak Brinkkemper: *Gamified requirements engineering: model and experimentation.* In *Requirements Engineering: Foundation for Software Quality: 22nd International Working Conference, REFSQ 2016, Gothenburg, Sweden, March 14-17, 2016, Proceedings 22*, pages 171–187. Springer, 2016. 41, 45, 48

[193] Arif, Mohd and Saoud Sarwar: *Identification of requirements using goal oriented requirements elicitation process.* International Journal of Computer Applications, 120(15), 2015. 41, 45

[194] Chua, Bee Bee, Danilo Valeros Bernardo, and June Verner: *Understanding the use of elicitation approaches for effective requirements gathering.* In *2010 Fifth International Conference on Software Engineering Advances*, pages 325–330. IEEE, 2010. 41, 48, 54, 55

[195] Rizwan Beg, Md, Md Muqeem, and Md Faizan Farooqui: *Extending application of non-verbal communication to effective requirement elicitation.* In *Advances in Computing and Information Technology: Proceedings of the Second International Conference on Advances in Computing and Information Technology (ACITY) July 13-15, 2012, Chennai, India-Volume 2*, pages 623–630. Springer, 2013. 41, 45, 48

[196] Zapata, Sergio, Estela Torres, Cesar A Collazos, Fáber D Giraldo, and Gustavo Sevilla: *Distributed elicitation of software requirements: an experimental case from argentina and colombia.* In *2013 8th Computing Colombian Conference (8CCC)*, pages 1–7. IEEE, 2013. 41, 45, 48

[197] Farinha, Carla and Miguel Mira da Silva: *Requirements elicitation with web-based focus groups.* In *Information Systems Development: Reflections, Challenges and New Directions*, pages 443–455. Springer, 2013. 41, 45, 48

[198] Vijayan, Jaya and G Raju: *A new approach to requirements elicitation using paper prototype.* International Journal of Advanced Science and Technology, 28:9–16, 2011. 41, 45, 48, 54

[199] Kumar, Manoj, Manish Shukla, and Sonali Agarwal: *A hybrid approach of requirement engineering in agile software development.* In *2013 International Conference on Machine Intelligence and Research Advancement*, pages 515–519. IEEE, 2013. 41, 45

[200] Kostopoulou, Stergiani, Marie Doyle, Ha Thu Truong, Jean Christophe Jacquier, Joe Houghton, Alanna Swan, Grace OŚullivan, Cécilia Travers, and Alan Morgan: *Building learning partnerships for innovation and transformation in education: An example from a creative and innovative programme for educators at a leading irish university.* All Ireland Journal of Higher Education, 14(1), 2022. 46

[201] Dziersk, Mark: *Design thinking... what is that.* Fast Company, 20, 2006. 46

[202] Liedtka, Jeanne: *Learning to use design thinking tools for successful innovation.* Strategy & Leadership, 39(5):13–19, 2011. 46

[203] Collins, Hilary: *Can design thinking still add value?* Design management review, 24(2):35–39, 2013. 46

[204] Retna, Kala S: *Thinking about "design thinking": A study of teacher experiences.* Asia Pacific Journal of Education, 36(sup1):5–19, 2016. 46

[205] Pressman, Andrew: *Design thinking: A guide to creative problem solving for everyone.* Routledge, 2018. 46

[206] Lambert, Rachel, Kara Imm, Rachel Schuck, Sunghee Choi, and Avery McNiff: *" udl is the what, design thinking is the how:" designing for differentiation in mathematics.* Mathematics Teacher Education and Development, 23(3):54–77, 2021. 46

[207] Tiwari, Saurabh, Santosh Singh Rathore, and Atul Gupta: *Selecting requirement elicitation techniques for software projects.* In *2012 CSI Sixth International Conference on Software Engineering (CONSEG)*, pages 1–10. IEEE, 2012. 62, 88, 89, 94, 97

[208] Srinivas, M, G Ramakrishna, K Rajasekhara Rao, and E Suresh Babu: *Analysis of legacy system in software application development: A comparative survey.* International Journal of Electrical and Computer Engineering, 6(1):292, 2016. 62, 88

[209] Canché, Maximiliano and José A Pino: *Requirements elicitation for collaborative systems: A systematic review.* In *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 297–304. IEEE, 2021. 62, 66, 67, 68, 69, 89, 90, 91, 93, 95, 96, 97, 99, 102

[210] Shirey, Maria R.: *Brainstorming for Breakthrough Thinking.* JONA: The Journal of Nursing Administration, 41(12), 2011, ISSN 0002-0443. `https://journals.lww.com/jonajournal/Fulltext/2011/12000/Brainstorming_for_Breakthrough_Thinking.1.aspx`. 62, 89

[211] ALFLEN, Naiara Crislaine and Edmir Parada Vasques PRADO: *Requirements elicitation techniques for software development: a systematic review of literature.* AtoZ: novas práticas em informação e conhecimento; v. 10, n. 1 (2021): jan./abr.; 39-49, 24(2):49–39, 2021. 62, 89

[212] Serbinski, Kendra: *Maple syrup and climate change in ontario: Assessing transdisciplinary research across multiple, related projects.* ., 2019. 62, 90

[213] Lundin, Hans and Lars Geschwind: *Exploring tuition fees as a policy instrument of internationalisation in a welfare state–the case of sweden.* European Journal of Higher Education, pages 1–19, 2021. 62, 90

[214] Salibi-Hachem, Souzan: *Critical Document Analysis of Equivalency Decree 4443 (2001) and Decree 848 (2018): An examination of equivalency criteria for upper secondary school qualifications in the UAE.* PhD thesis, The British University in Dubai (BUiD), 2019. 62, 90

[215] Nixon, Amuomo and Collins Otieno Odoyo: *Ethnography, its strengths, weaknesses and its application in information technology and communication as a research design.* Comput Sci Inf Tech, 8:50–6, 2020. 62, 68, 90

[216] Herbert, Steve: *For ethnography.* Progress in human geography, 24(4):550–568, 2000. 62, 68, 90

[217] Arif, Qadeem Khan Shams-ul and SAK Gahyyur: *Requirements engineering processes, tools/technologies, & methodologies.* International Journal of reviews in computing, 2(6):41–56, 2009. 62, 90, 93, 94, 95, 97, 102

[218] Anwer, Faiza, Shabib Aftab, Usman Waheed, and Syed Shah Muhammad: *Agile software development models tdd, fdd, dsdm, and crystal methods: A survey.* International journal of multidisciplinary sciences and engineering, 8(2):1–10, 2017. 62, 90, 91

[219] Goyal, Sadhna: *Major seminar on feature driven development.* Jennifer Schiller Chair of Applied Software Engineering, 2008. 62, 90, 91

[220] Nawaz, Zahid, Shabib Aftab, and Faiza Anwer: *Simplified fdd process model.* International Journal of Modern Education and Computer Science, 9(9):53, 2017. 62, 91

[221] Tirumala, SS, Shahid Ali, and Anjan Babu: *A hybrid agile model using scrum and feature driven development.* International Journal of Computer Applications, 156(5):1–5, 2016. 62, 90, 91

[222] Acocella, Ivana: *The focus groups in social research: advantages and disadvantages.* Quality & Quantity, 46(4):1125–1136, June 2012, ISSN 1573-7845. `https://doi.org/10.1007/s11135-011-9600-4`. 62, 91

[223] Brown, Jennifer: *Interviews, focus groups, and delphi techniques.* In *Advanced Research Methods for Applied Psychology*, pages 95–106. Routledge, 2018. 62, 91, 92

[224] Queirós, André, Daniel Faria, and Fernando Almeida: *Strengths and limitations of qualitative and quantitative research methods.* European Journal of Education Studies, 0(0), 2017, ISSN 25011111. `https://oapub.org/edu/index.php/ejes/article/view/1017`. 62, 66, 69, 92

[225] Adhabi, Essa and Christina Blash Anozie: *Literature review for the type of interview in qualitative research.* International Journal of Education, 9(3):86–97, 2017. 62, 92

[226] Alshenqeeti, Hamza: *Interviewing as a data collection method: A critical review.* English linguistics research, 3(1):39–45, 2014. 62, 66, 69, 92

[227] Nurlaila, Alma Prima: *The use of mind mapping technique in writing descriptive text.* Journal of English and Education, 1(2):9–15, 2013. 62, 94

[228] Yunus, Melor Md, Chan Hua Chien, *et al.*: *The use of mind mapping strategy in malaysian university english test (muet) writing.* Creative Education, 7(04):619, 2016. 62, 94

[229] Davies, Martin: *Concept mapping, mind mapping and argument mapping: what are the differences and do they matter?* Higher Education, 62(3):279–301, September 2011, ISSN 1573-174X. `https://doi.org/10.1007/s10734-010-9387-6`. 62, 94

[230] Acuña, Silvia T, John W Castro, and Natalia Juristo: *A hci technique for improving requirements elicitation.* Information and Software Technology, 54(12):1357–1375, 2012. 62, 67, 95, 96

[231] Patten, Mildred: *Questionnaire research: A practical guide.* Routledge, 2016. 62, 68, 97

[232] Wolniak, Radosław: *The use of qfd method advantages and limitation.* Production Engineering Archives, 18, 2018. 62, 98

[233] Andronikidis, Andreas, Andreas C Georgiou, Katerina Gotzamani, and Konstantina Kamvysi: *The application of quality function deployment in service quality management.* The TQM journal, 2009. 62, 97, 98

[234] Brugha, Ruairí and Zsuzsa Varvasovszky: *Stakeholder analysis: a review.* Health policy and planning, 15(3):239–246, 2000. 62, 99, 100

[235] Eskerod, Pernille and Tina Larsen: *Advancing project stakeholder analysis by the concept 'shadows of the context'.* International Journal of Project Management, 36(1):161–169, 2018. 62, 99, 100

[236] Prell, Christina, Klaus Hubacek, and Mark Reed: *Stakeholder analysis and social network analysis in natural resource management.* In *Handbook of applied system science*, pages 367–383. Routledge, 2016. 62, 100

[237] Cross, Ruth and Louise Warwick-Booth: *Using storyboards in participatory research.* Nurse researcher, 23(3), 2016. 62, 100

[238] Chen, Shizhe, Bei Liu, Jianlong Fu, Ruihua Song, Qin Jin, Pingping Lin, Xiaoyu Qi, Chunting Wang, and Jin Zhou: *Neural storyboard artist: Visualizing stories with coherent image sequences.* In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2236–2244, 2019. 62, 100

[239] Sova, Ron and Deborah Hinderer Sova: *Storyboards: a dynamic storytelling tool.* In *Proceedings of the 2006 UPA conference on Usability through Storytelling*, pages 1–12, 2006. 62, 100

[240] Jaatun, Martin Gilje, Inger Anne Tøndel, Nils Brede Moe, Daniela Soares Cruzes, Karin Bernsmed, and Børge Haugset: *Accountability requirements for the cloud.* In *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 375–382, 2017. 62, 68, 101, 102

[241] Hickey, Ann and Douglas Dean: *Prototyping for requirements elicitation and validation: A participative prototype evaluation methodology.* IEEE, 1998. 66, 103

[242] Tuunanen, Tuure: *A new perspective on requirements elicitation methods.* Journal of Information Technology Theory and Application (JITTA), 5(3):7, 2003. 66, 103

[243] Anwar, Fares and Rozilawati Razali: *A practical guide to requirements elicitation techniques selection-an empirical study.* Middle-East Journal of Scientific Research, 11(8):1059–1067, 2012. 66, 103

[244] Fotrousi, Farnaz, Samuel A Fricker, and Markus Fiedler: *Quality requirements elicitation based on inquiry of quality-impact relationships.* In *2014 IEEE 22nd international requirements engineering conference (RE)*, pages 303–312. IEEE, 2014. 66, 103

[245] Bergqvist, Peppe, Henrik Eneroth, Joel Hinz, Robert Krevers, Gustav Ladén, and Anders Mellbratt: *Categorization and visualization of news items.* Cognitive Science Program, University of Linkoping, 2006. 66

[246] Coorevits, Lynn, Dimitri Schuurman, Kathy Oelbrandt, and Sara Logghe: *Bringing personas to life: user experience design through interactive coupled open innovation.* Persona Studies, 2(1):97–114, 2016. 66

[247] Singh, Vandana: *Personas and scenarios as a methodology for information sciences.* Qualitative and Quantitative Methods in Libraries, 7(1):123–134, 2018. 66

[248] Vallet, Flore, Jakob Puchinger, Alexandra Millonig, Guillaume Lamé, and Isabelle Nicolaï: *Tangible futures: Combining scenario thinking and personas-a pilot study on urban mobility.* Futures, 117:102513, 2020. 66

[249] Linaker, Johan, Sardar Muhammad Sulaman, Martin Höst, and Rafael Maiani de Mello: *Guidelines for conducting surveys in software engineering v. 1.1.* Lund University, 50, 2015. 69

[250] Ahmad, Muhammad Ovais, Denis Dennehy, Kieran Conboy, and Markku Oivo: *Kanban in software engineering: A systematic mapping study.* Journal of Systems and Software, 137:96–113, 2018, ISSN 0164-1212. `https://www.sciencedirect.com/science/article/pii/S0164121217302820`. 80, 81

[251] Fowler, Frederik M.: *What Is Scrum?*, pages 3–8. Apress, Berkeley, CA, 2019, ISBN 978-1-4842-4164-6. `https://doi.org/10.1007/978-1-4842-4164-6_1`. 81

[252] Romao, Mário, Joao Costa, and Carlos J Costa: *Robotic process automation: A case study in the banking industry.* In *2019 14th Iberian Conference on information systems and technologies (CISTI)*, pages 1–6. IEEE, 2019. 81

[253] Jabbari, Ramtin, Nauman bin Ali, Kai Petersen, and Binish Tanveer: *What is devops? a systematic mapping study on definitions and practices.* In *Proceedings of the Scientific Workshop Proceedings of XP2016*, pages 1–11, 2016. 81

[254] Hsia, Pei, Alan M Davis, and David Chenho Kung: *Status report: requirements engineering.* IEEE software, 10(6):75–79, 1993. 88

[255] Kotonya, Gerald and Ian Sommerville: *Requirements engineering: processes and techniques.* Wiley Publishing, 1998. 88

[256] Nuseibeh, Bashar and Steve Easterbrook: *Requirements engineering: a roadmap.* In *Proceedings of the Conference on the Future of Software Engineering*, pages 35–46, 2000. 88

[257] Paetsch, Frauke, Armin Eberlein, and Frank Maurer: *Requirements engineering and agile software development.* In *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.*, pages 308–313. IEEE, 2003. 88

[258] Parizi, Rafael, Marina Moreira da Silva, Igor Couto, Kendra Trindade, Matheus Plautz Prestes, Sabrina dos Santos Marczak, Tayana Conte, and Heloisa Candello: *Design thinking in software requirements: What techniques to use? a proposal for a recommendation tool.* In *Proceedings of the XXIII Ibero-American Conference on Software Engineering-CIbSE 2020, 2020, Estados Unidos.*, 2020. 88

[259] Li, Jinyu, Asif Ullah, Jun Li, Shah Nazir, Habib Ullah Khan, Hanif Ur Rehman, and Amin Ul Haq: *Attributes-based decision making for selection of requirement elicitation techniques using the analytic network process.* Mathematical Problems in Engineering, 2020:2156023, 2020. `https://doi.org/10.1155/2020/2156023`. 88

[260] Bowen, Glenn A.: *Document Analysis as a Qualitative Research Method*. Qualitative Research Journal, 9(2):27–40, January 2009, ISSN 1443-9883. `https://doi.org/10.3316/QRJ0902027`, visited on 2022-11-09, Publisher: Emerald Group Publishing Limited. 89

[261] Adlin, Tamara and John Pruitt: *The essential persona lifecycle: Your guide to building and using personas*. Morgan Kaufmann, 2010. 95

[262] Hudson, William: *User stories don't help users: Introducing persona stories*. interactions, 20(6):50–53, 2013. 95, 101

[263] Lauff, Carlye A, Daria Kotys-Schwartz, and Mark E Rentschler: *What is a prototype? what are the roles of prototypes in companies?* Journal of Mechanical Design, 140(6), 2018. 96

[264] Obendorf, Hartmut, Axel Schmolitzky, and Matthias Finck: *Xpnue–defining and teaching a fusion of extreme programming and usability engineering*. In *Inventivity: Teaching theory, design and innovation in HCI, Proceedings of of HCIEd2006-1 First Joint BCS/IFIP WG13. 1/ICS/EU CONVIVIO HCI Educators' Workshop, 23-24 March 2006, Limerick, Ireland*, pages 103–108. Citeseer, 2006. 96

[265] Węgrzyn, Joanna and Anna Wojewnik-Filipkowska: *Stakeholder analysis and their attitude towards ppp success*. Sustainability, 14(3):1570, Jan 2022, ISSN 2071-1050. `http://dx.doi.org/10.3390/su14031570`. 100

[266] Gundumogula, Manju: *Importance of focus groups in qualitative research*. The International Journal of Humanities & Social Studies, 8(11), 2020. 102

[267] Basso, Fábio Paulo, Raquel Mainardi Pillat, Fabricia Roos-Frantz, and Rafael Z Frantz: *Combining mde and scrum on the rapid prototyping of web information systems*. International Journal of Web Engineering and Technology, 10(3):214–244, 2015. 103

[268] Lai, Sen Tarng: *A user story quality measurement model for reducing agile software development risk*. International Journal of Software Engineering & Applications, 8(2):75–86, 2017. 103

[269] Marques, Anna Beatriz, Alex Felipe Costa, Ismayle Santos, and Rossana Andrade: *Enriching user stories with usability features in a remote agile project: a case study*. In *Proceedings of the XXI Brazilian Symposium on Software Quality*, pages 1–10, 2022. 103

[270] Gilson, Fabian, Matthias Galster, and François Georis: *Generating use case scenarios from user stories*. In *Proceedings of the international conference on software and system processes*, pages 31–40, 2020. 103

[271] Keshk, Nader, Mohammad El-Ramly, and Akram Salah: *A proposal for enhancing agile requirements engineering with prototyping and enriched user stories*. In *Proceedings of the Federated Africa and Middle East Conference on Software Engineering*, pages 59–63, 2022. 103

[272] Misztal-Radecka, Joanna and Bipin Indurkhya: *Persona prototypes for improving the qualitative evaluation of recommendation systems.* In *Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, pages 206–212, 2020. 103

[273] Simões, David, Pedro Antunes, and Luis Carriço: *Eliciting and modeling business process stories.* Business & Information Systems Engineering, 60:115–132, 2018. 113

[274] Bandara, Wasana, Guy G Gable, and Michael Rosemann: *Factors and measures of business process modelling: model building through a multiple case study.* European Journal of Information Systems, 14(4):347–360, 2005. 114