NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

SALVADOR ROSA MENDES

Master of Science

A SUSTAINABILITY-DRIVEN APPROACH FOR AGILE SOFTWARE REQUIREMENTS

MASTER IN COMPUTER SCIENCE

NOVA University Lisbon
March, 2022

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF
COMPUTER SCIENCE

A SUSTAINABILITY-DRIVEN APPROACH FOR AGILE SOFTWARE REQUIREMENTS

SALVADOR ROSA MENDES

Master of Science

Adviser: João Baptista da Silva Araújo Junior
Associate Professor, FCT NOVA University of Lisbon

Co-adviser: Ana Maria Diniz Moreira
Associate Professor with Habilitation, FCT NOVA University of Lisbon

MASTER IN COMPUTER SCIENCE
SPECIALIZATION IN SOFTWARE SUSTAINABILITY

NOVA University Lisbon
March, 2022

A Sustainability-Driven Approach for Agile Software Requirements

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v6.9.10) [21].

Acknowledgements

To my adviser, João Baptista da Silva Araújo Junior, and my co-adviser Ana Maria Diniz Moreira, I am thankful for the opportunity and proposal to work on this topic and also for all of the support you've given me all throughout, even at times when I shunned from asking for help. This is no doubt an experience I will not forget.

To my dear friend and the creator of the sustainability catalogue used as a base for my work, Diogo Albuquerque, many thanks for the availability throughout this whole journey, alongside advice and materials given that helped me achieve my goals.

Lastly, a big thank you to my family and girlfriend for the patience and emotional support and help in keeping me focused on my objectives even when motivation was lacking.

"Conseguimos! Portugal, Lisboa, esperávamos, desejávamos, conseguimos, vitória! " (Presidente Marcelo Rebelo de Sousa)

Abstract

The SDGs (sustainable development goals) demand for the involvement of all sectors of our society, from industry to academia. Therefore, the race towards sustainability has become increasingly more important and widely covered. Sustainability involves five different dimensions: environmental, economic, social, technical and individual. These dimensions are also of interest for software development since they have an impact not only on each other but also on the base requirements of a system. Therefore, well-informed design decisions require improved support to reason on such intra- and inter-relationships and impacts, early in the development. We are interested in reasoning about these relationships in an agile context where user stories play a major role, but do not handle sustainability requirements in a methodical way. For this purpose, our first step was to understand how developers from the agile community handled sustainability. The second step was to perform a systematic mapping study where we furthered our understanding of sustainability, and software development with agile methodologies. With this study we also gained awareness concerning any approaches that have been proposed regarding the integration of sustainability in an agile context. This study is the foundation for the third part of our work: a solution that consists in extending and adapting an existing catalogue of sustainability requirements into a catalogue of sustainability user stories for later reuse during the early stages of an agile software development process. We validated the approach with a case study (U-Bike) and performed an online survey for qualitative evaluation of the proposed approach.

Keywords: sustainable software, sustainability requirements, agile methodologies, user stories.

Resumo

Os ODS (objetivos de desenvolvimento sustentável) exigem o envolvimento de todos os setores da nossa sociedade, da indústria à academia. Por isso, a corrida pela sustentabilidade tem se tornado cada vez mais importante e amplamente divulgada. A sustentabilidade envolve cinco dimensões diferentes: ambiental, econômica, social, técnica e individual. Essas dimensões também são de interesse para o desenvolvimento de software, pois impactam não apenas umas nas outras, mas também nos requisitos básicos de um sistema. Portanto, decisões de projeto bem informadas requerem suporte aprimorado para raciocinar sobre tais relacionamentos e impactos intra e inter-relacionados, logo no início do desenvolvimento. Estamos interessados em raciocinar sobre esses relacionamentos em um contexto ágil, onde as *user stories* desempenham um papel importante, mas não tratam os requisitos de sustentabilidade de maneira metódica. Para isso, nosso primeiro passo foi entender como os desenvolvedores da comunidade ágil lidavam com a sustentabilidade. A segunda etapa foi realizar um estudo de mapeamento sistemático onde ampliamos nosso entendimento sobre sustentabilidade e desenvolvimento de software com metodologias ágeis. Com este estudo também tomamos conhecimento de quaisquer abordagens que tenham sido propostas em relação à integração da sustentabilidade em um contexto ágil. Este estudo é a base para a terceira parte do nosso trabalho: uma solução que consiste em estender e adaptar um catálogo existente de requisitos de sustentabilidade em um catálogo de *user stories* de sustentabilidade para posterior reutilização durante os estágios iniciais de um processo de desenvolvimento ágil de software. Validamos a abordagem com um estudo de caso (U-Bike) e realizamos uma pesquisa online para avaliação qualitativa da abordagem proposta.

Palavras-chave: software sustentável, requisitos de sustentabilidade, metodologias ágeis, user stories.

Contents

List of Figures

List of Tables

1 Introduction

This project was influenced by the ever growing need to make life as sustainable as possible due to the noticeable impacts human lifestyle has caused on the planet. Nowadays sustainability is becoming increasingly more documented and well researched but has yet to make a high impact in the software development industry. For that reason and due to the high adoption rate of agile methodologies in industry we hope to provide useful work that allows advances in the sustainability of software developed through the agile process. This introductory chapter will present the context and motivation for this MSc dissertation work, whilst discussing the problems and challenges encountered, the objectives of this work and the expected contributions. Finally there will be an overview of the document's structure.

1.1 Context and motivation

Sustainability is "a process of change in which the exploitation of resources, the direction of investments, the orientation of technological development and institutional change are all in harmony and enhance both current and future potential to meet human needs and aspirations" [16]. In recent years, Sustainability has been a very discussed topic worldwide. Implementation norms of sustainability, like sustainability reports, international treaties and protocols, have been put in place all over the world in order to start mitigating the impacts of human lifestyle and safeguard the prosperity of future generations and the planet. It is especially important and time sensitive to research and implement sustainability measures since our current lifestyle isn't maintainable for long and has already caused irreversible damage to the planet and existing ecosystems, as such, for it to be effective, we must also embrace it as a part of our daily life.

Sustainability is a broad topic and as such it needs to be divided into groups to better identify the problems and challenges that come from each sustainability dimension. Sustainability is classically divided into three main dimensions: economic, environmental and social[4]. However nowadays, some authors also consider the technical dimension

and the individual as part of sustainability [13]. These dimensions have inter and intra-relations meaning that a requirement that is part of a dimension can have a relationship to another requirement from that same dimension or from a different dimension or even both since requirements aren't always mutually exclusive to a single dimension. This is important to understand because the trade-offs between the requirements have to be well pondered in order to make the correct choices of the requirements to implement for the given project so that it maximizes its sustainability.

The motivation for this dissertation is linked with the concept of sustainability but applied to agile software development. This is relevant because we now live in the digital age and software controls and fills our daily lives. Therefore, if we want to create a sustainable way of life on this planet, the software and technology we use must also be sustainable. However, for software to be truly sustainable, its development process has to be as well. Agile development processes are becoming more and more used throughout the industry[9], whether through Scrum or Kanban. This dissertation aims to provide an agile approach to deal with sustainability in the early stages, and even support it during the whole development process. The sustainability integration in these early stages will be done primarily with the usage of user stories that will be adapted to describe sustainability requirements. In short, user stories are a key component of agile methodologies and an informal and general description of a software feature that in turn represents the stakeholders needs and wants for the software. Therefore, it is critical that sustainability is introduced in the early stages because poor software requirements are a major cause for project failure[1].

1.2 Problem statement

The problem that prompted this dissertation is the **lack of approaches to support the development of sustainable software through agile methods**. This problem has been confirmed by the results of a systematic mapping study performed as part of the state of the art research for this work. The section that follows divides this problem into two challenges.

Problem 1: The need to address sustainability in agile software development

As a means to correctly implement sustainability we need to handle it from the earliest stages we can, seeing as the further along we are in the development process the harder and more costly changes to the software become. So sustainability and the impacts the software has have to be pondered and analysed as soon as possible. In agile this means tackling the problem in the form of user stories, that will reflect the sustainability goals and requirements necessary for the system. It is also a must to understand how the agile processes work to be able to introduce sustainability, whether in the form of a parallel

process that complements and supports the already existing agile one or by altering the existing events that compose agile methodologies to encompass sustainability.

Challenge 1: What is the current state of the art?

To overcome this challenge we needed to perform something more than a simple search that would very likely miss too many relevant documents. So, we completed a systematic mapping study on sustainability in agile methodologies by using well constructed search questions. The whole process is described in detail in Section 3.1. In a brief explanation, the study consisted in an automated search using query strings, based on the research questions, through DBLP, that compiles several accredited online digital libraries like IEEExplore and ACM. This automated search returned quite a few results which were then analysed and selected for the final data extraction process with the help of a set of inclusion and exclusion criteria. After this, a manual search was also performed in Google scholar to gather possible relevant documents that might have been missed in the automated search. This search consisted in querying the database with different combinations of the key terms of the original search queries. As a second part of this challenge, we surveyed software developers in the agile industry in Portugal to understand how they handle sustainability in their projects (4).

Challenge 2: Providing an approach for specifying sustainability requirements in an agile development context. We are interested in providing an approach that encourages practitioners of agile development to specify sustainability requirements. The approach needs to be appealing to those practitioners and easily integrated to their current practices. To overcome this challenge it would be useful to reuse preexisting information, for example from a catalogue, and from there to derive requirements for sustainability in a form that is understood and used by those practitioners, such as user stories.

1.3 Objectives

The objective of this dissertation is to present an approach based on agile methodologies to aid developers in producing sustainable software. This approach will allow the developers to quickly and efficiently select sustainability qualities for their projects. In turn with the help of the tool support transform those sustainability qualities into sustainability user stories that can be used as artefacts in agile development. The approach will allow user customization making it flexible and adaptable to as many application domains and development methodologies as possible, as long as they are based on agile. The ease of use and flexibility of the tool support and user stories will hopefully encourage more developers to include sustainability qualities in their future projects.

1.4 Contributions

This dissertation aims to contribute with an approach to support developers using agile methodologies to specify sustainability requirements. We want to provide this through the use of a sustainability catalogue, which with improvements, will allow the user to derive general-purpose sustainability user stories, user stories being one of the most important artefacts in agile development.

1.5 Structure of the document

The remain of this document is organised into the following chapters:

- **Background:** this chapter contains an overview of requirements engineering, agile methodologies and sustainability engineering.
- **State of art:** this chapter presents the major part of the research work performed during the preparation period of this dissertation. It details the research methodology used along with the results and their analysis. Other related work is also discussed.
- **Survey:** this chapter presents the results of a survey made with the aim of assessing the state of sustainability in agile software development in Portugal.
- **The Approach:** this chapter presents the conceptualization of our solution. This encompasses the process created to derive the user stories and the features that comprise our tool support.
- **Applying the Process To a Case Study:** this chapter focuses on the application of the developed solution to a real world example in order to assess its usefulness.
- **Evaluation:** this chapter covers the process and survey done to obtain feedback from IT professionals and evaluate our work.
- **Conclusions:** this chapter summarizes our work and presents our conclusions. It also introduces some ideas for future work to be developed.

2

Background

In this chapter we describe the subjects covered in this dissertation, such as requirements engineering, agile methodologies, and sustainability engineering. For requirements engineering we present the main concepts, what activities are associated with it and some key techniques used. For the agile methodologies we introduce some well-known methodologies (e.g., Kanban and Scrum), detailing each of the variants' respective events and activities. Regarding sustainability engineering, we present the main concepts, its challenges and possible solutions. Lastly, we will describe the sustainability connection and relevance to software engineering.

2.1 Requirements Engineering

Requirements engineering is one of the most important parts of software development and is closely related to project failures [39] if not done properly and to its full extent. At the early stages of a new project, it is a must that requirements be clearly identified and detailed along with its objectives in order to attend to the stakeholders needs. To better understand requirements engineering, a proper definition of requirements must be presented. A requirement "is a physical or functional need that a particular design, product or process aims to satisfy" [37].

A requirement can be functional or non-functional [37]. A functional requirement specifies a service a system should provide. Its non-functional counterpart represents how the system should operate, describing its quality characteristics and limitations. Now armed with a clear idea of what a requirement is we can define requirements engineering as "the process of defining, documenting and maintaining requirements in the engineering design process" [49]. Requirements engineering can be a complex process caused by the need to identify and analyze all the requirements and constraints presented by the stakeholders, where its activities are elicitation, analysis and negotiation, documentation, validation, and management [36].

Elicitation consists in meeting with the stakeholders to elicit and learn about the business logic to better understand their objectives, needs and limitations. In the end of this

activity all system requirements are identified and should be rigorously and systematically described. After identified and documented, requirements must be analyzed and discussed (with developers and stakeholders) to find out possible problems such as ambiguities, omissions or conflicts. During the Analysis and Negotiation stage, solutions are presented along with their benefits and trade-offs, and various alternatives are discussed until an agreement can be reached with the stakeholders.

The Documentation stage has the objective of transforming all of the system requirements into a clear and easy to understand format for the stakeholders and technical teams, while still guaranteeing completeness and accuracy. The Validation stage is where all the documents and models created for the system are validated to make sure they clearly and accurately convey the needs of the stakeholders in a way that leaves no room for misinterpretation or error.

The Management stage is one that is ongoing throughout the whole development process since it aims to deal with maintaining the requirements and related documentation accurate and updated on the stakeholders needs and changes, if any. This is an important activity to ensure no other development process fails or is slowed down by incorrect or inconsistent views of the system's requirements.

The next sections discuss the elicitation, prioritization and trade-off analysis processes of requirements engineering in more detail.

2.1.1 Elicitation

Requirements elicitation is the activity that focuses on finding out and understanding the system's requirements needed from stakeholders and every other group of system users, like admins, customers, etc. There are various techniques to fulfil requirement elicitation but only ones that have been proven to be effective [38, 24] will be presented in more detail below.

- **Interviews** are a simple technique and way of obtaining information directly from the source, usually already known facts like existing documentation or procedures, by usually having an interview meeting with the stakeholders or users of the system in order to acquire relevant information. Interviews have the benefit of privacy, if necessary, by interviewing only one person at a time. There are two types of interviews: structured, where questions are prepared beforehand, or unstructured [10]. A downside of interviews is that it is only as effective as the interviewer [36], meaning that if the person has less experience or familiarity with the domain less accurate information or even less information may be gathered.

- **Surveys/Questionnaires** are a paper or online way of gathering information about the requirements by sending a questionnaire to one or more people from the group(s)

of users or stakeholders [10]. The questionnaire may have open or close ended questions, but regardless its type, the questions must be clear and to the point. This technique has the upside of being the least expensive in terms of money and time.

- **Scenarios/Use cases** are the description of the possible functions the system might have to offer and how that system is going to behave [10]. They are used to better understand the general requirements of the system and help identifying ambiguous requirements.

  User stories are a scenario-based requirements description, that will be discussed in more detail in Section 2.2.2.

- **Workshops** are structured group meetings with set objectives, during which a group of people, preferable from different sectors within the organization in order to increase the completeness of the information gathered, engage with each other and discuss to achieve the set of objectives for the meeting. Workshops are useful in resolving conflicts, if the interested parties for such conflict are present and actively participate in the discussion [24]. This technique has the same downside as interviews where it is very time consuming and can be hard to schedule due to the involvement of more than one person.

- **Brainstorming** is another group technique that focuses on developing new ideas about the project at hand. It is an unstructured meeting since the ideas are achieved by the participants spontaneously and in no particular order [24]. This has the advantage of promoting creative thinking which in turn can lead to new and more advanced alternatives to solve a problem or implement a feature.

- **Focus Groups** are a semi structured group technique that mixes properties from interviews and workshops, where participants are incited by an interviewer [10] to have an open discussion about the problems at hand. This has the objective of eliciting requirements and has the benefit of promoting teamwork that is effective in solving conflicts.

- **Prototyping** consists in the development of a simplified and minimalist version of the system-to-be to augment the understanding and elicitation of the requirements needed for the final system [10]. Prototypes are developed mostly using unrefined requirements or existing examples of similar systems and are mainly used for projects of complex systems and human computer interfaces.

2.1.2 Prioritization

Decisions are an integral part of software development. They go from deciding what requirements are needed for the system to which of those requirements will be implemented first, since it is not feasible to implement them all in a single increment. This

task's complexity increases the bigger the number of stakeholders involved. This is due to the associate increase of choices available to the developers. A way to facilitate decision making is to use requirements prioritization. This strategy consists in determining the order of importance the requirements have for the system implementation and for the stakeholders, hence being helpful to identify the most important requirements to best fulfill the stakeholders' needs and satisfaction [17]. The prioritization technique is very important to aid the development teams combat the limitations usually associated with a project, like limited human resources, limited budget and deadlines they have to follow. The developers can adapt each development increment to the current constraints and pending requirements by having a different set of requirements for each. As said previously, this process can become complicated if there are several stakeholders involved, especially if they are from different business areas, as each one might have a different perspective on what is most important for the project and the development process and therefore might become harder to reach a consensus.

In order for the development team to adhere to the constraints imposed on the project's resources, there are five main factors that need to be considered when analyzing a requirement priority [22]. These factors are:

- **Importance:** this is probably the most relevant factor out of the five. That said, one has to be careful and properly define and explain its definition to the stakeholders since this is a subjective concept and everyone can have it's own meaning for the word.
- **Time:** this refers to the required time needed to correctly implement the requirement. This factor is influenced by the experience of the team and the number of people working on it.
- **Cost:** this refers to the monetary cost to correctly implement the requirement. This factor is influenced by staff hours and extra resources needed.
- **Penalty:** this is the money it would cost if the requirement is not correctly implemented. This is not directly related to the importance of the requirement since one with a low importance can have a high penalty cost.
- **Risk:** this refers to the probability that the project will fail to comply with the set resources constraints. The project's risk is calculated by adding up all the risks for each of the project's requirements. This factor is associated with unrealistic schedules and budgets, too many changes to the already existing requirements, or by having more features than necessary.

To assist in the prioritization of requirements, there are a plethora of techniques, each with unique strengths and weaknesses allowing developers to tailor their prioritization process to the projects' characteristics and teams' needs. It is hard to point out which

technique is the best or the worst since they are different from each other and optimal in specific scenarios. A valid way to combat the weaknesses of each technique is to combine two or more. The techniques are split into three categories depending on what type of scale they use:

- **Nominal** scale assigns requirements to priority groups, where every requirement within that group has the same priority level;
- **Ordinal** scale orders all of the requirements in a priority list;
- **Ratio** scale works with a relative difference in priorities between the requirements. The methods used for this must be carefully selected, to make sure all of the requirements have a priority value associated.

2.1.3 Trade-off Analysis

Trade-offs are, by definition in the Oxford dictionary, "a situation in which you accept something bad in order to have something good", or in other words better fitting to software requirements, consider some requirements in detriment of those from another requirement or requirements.

This is an often complex process but one frequently necessary to achieve the stakeholders' maximum level of satisfaction. The complexity comes from having to fully identify the dependencies between the different requirements, which requirements affect which and with what magnitude, having to know the cost, time and importance of each requirement and understanding the whole context on how the requirements interact with each other to create the final build. Only after knowing and understanding all the previously stated items can we begin to analyse trade-offs between requirements and the advantages and disadvantages associated with each of the possibilities and after make a comprehensive decision about the trade-offs.

2.2 Agile Methodologies

Agile is a fairly recent software development and project management methodology that promotes development through continuous iterations rather than all at once, it is one of the simplest and most effective ways of converting the vision of a project into a finalized product.

2.2.1 Agile Development

As said before, the development of software in an Agile setting happens in iterations and unlike other models, like waterfall, the development and testing are activities executed in parallel and happen in every iteration of the software's life cycle. Agile is known for

employing continuous planning, learning and improvement, focusing on team collaboration and early delivery. All this results in a process that is flexible and responsive to change. There are four core values of agile that must always be present in mind when using or talking about agile [43]:

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

Two of most adopted agile development methodologies are Kanban and Scrum [9]. They are very different in the way they approach the product backlogs, user stories and development iterations, all of which will be explained in Sections 2.2.3 and 2.2.4. But before we can understand what Scrum and Kanban are, and the differences between them, we must first understand what user stories are and what they are used for.

2.2.2 User Stories

User stories are an informal and concise explanation of a system feature [40]. This explanation is written from the perspective of the end user for that feature. Here we see once again in typical agile fashion, user stories putting people at the center of the discussion. User stories are part of a hierarchy of agile artefacts used to describe requirements of the project. The other artefacts are [44]:

- **Epics** are large bodies of work that can be broken down into a number of smaller tasks called user stories, usually involving multiple development teams on multiple projects and are rarely delivered in a single sprint.
- **Initiatives** are a collection of epics that drive towards a common goal. Initiatives are usually big goals that can't be achieved by any epic alone.
- **Themes** are labels that track high-level organizational goals, helpful in understanding and keeping track of what work contributes to what organizational goal.

The main goal of user stories is to provide a guideline of how a piece of software will deliver value back to the customer. Customers are not meant in the traditional sense as they do not have to necessarily be external end users and can be members of the organization of which the project belongs to. User stories are usually a few sentences long in simple and informal language that serves to outline the desired goal for the feature, further detail is added after with requirements once they are agreed upon. The key benefits of utilizing user stories and not just system requirements are [40]:

- **The focus is kept on the user**, instead of having just a list of tasks that need to be done and teams worrying about simply delivering those items. Keeping a list of user stories makes the teams focused on the fact they are solving problems for the real users.

- **They promote collaboration**, because once the end goal is set, the team works together to find a solution to solve the problem and meet the end user's goal.

- **They encourage creative thinking and solutions**, since the problems at end are usually specific to a user and its needs; there is not a complete solution to the problem and so the team must work together to find a solution.

- **They create momentum**, each user story is a small challenge for the team to solve and with each story completed a small win is achieved giving the team momentum and a confidence boost into the next one.

Bellow there is a brief explanation of the core set of conditions and rules to follow in order to write a user story [40].

- **Definition of "Done"** - A user story is usually completed when the end user can fulfil the task described, meaning we must be careful to be clear on what the end goal is.

- **Outline sub-tasks or tasks** - Clearly and fully define what steps are necessary to complete the task at hand and who is responsible for them.

- **User personas** - If there are multiple end users it might be beneficial to create multiple stories. Personas are useful to achieve this as they are fictional users that represent the different end users of the system alongside their different preferences and needs.

- **Ordered steps** - If a system's process is large it can be useful to create a user story for each step of that process.

- **Listen to feedback** - Talk to the users and stakeholders to better understand their needs and listen to feedback. There's no reason to guess a user story's goal when you can hear it from the source.

- **Time** - Time is a scarce resource and many development teams rely solely on their frameworks to estimate the time it will take to complete a story. Since stories should be achievable in one sprint, large stories must be divided into smaller stories or maybe consider the original story an epic.

2.2.3 Scrum

The Scrum framework [46, 35] is based on continuous learning and adjustment; it concedes the fact that the teams do not know everything at the start of the project and will evolve through experience. For this, Scrum is structured in a way that helps teams naturally adapt to changing conditions and requirements, having short release cycles and re-prioritization built into the process. Although Scrum is structured it is not rigid and as such any team that uses it can tailor the process to their needs and preferences. There is much discussion about what the best Scrum approach is, but key factors like clear communication, transparency and a dedication for continuous improvement are always present in any of the successful approaches. Scrum is composed of three artefacts – product backlog, sprint backlog, and increment – and six ceremonies – backlog grooming, sprint planning, sprint, daily scrum, sprint review, and sprint retrospective.

Product Backlog. The product backlog is a collection of all the work that needs to be done, this collection is maintained by the product owner and/or by the product manager. This collection is dynamic and changes overtime in response to the changing needs and conditions the market the product is inserted in has. To answer to the changes to the collection the tasks within it have to be re-prioritized, revisited and maintained by the product owner to make sure they are still compliant.

Sprint Backlog. The sprint backlog is the list of user stories or bug fixes chosen by the development team to be implemented in the current sprint cycle. Before each sprint begins a sprint planning meeting, discussed further ahead, takes place in which the team discusses and chooses which are the stories or bugs from the product backlog they will implement in the sprint. The sprint backlog is flexible and can evolve during the sprint, but the implementation and overall achievement of the original items cannot be compromised by these changes.

Increment or sprint goal. An increment is the usable product that results from the work done in a sprint. This can be demonstrated in a demo after each sprint if the team chooses to do so but it is not a mandatory practice.

User stories in Scrum. In scrum user stories chosen are added to sprints and "burned down" or completed during that sprint. This burn down of the stories helps teams get more accurate estimations and improve their sprint planning.

The Scrum ceremonies are described as follows.

Backlog grooming. The responsibility for this activity falls into the product owner's hands whose main job is to make sure the product is heading into his goal for it, while

being aware of the market and the customer changes that might occur. Hence, the product owner, in order to maintain this list and correctly prioritize it, s/he uses the feedback provided by the users and the development team.

Sprint planning. This activity's aim is to define what are the user stories that will be tackled in the sprint and the time the sprint will take. This activity is the responsibility of the whole development team, the Scrum master and sometimes the product owner. Before the user stories can be chosen, a goal for the sprint has to be agreed and only then can user stories that align with that goal be pulled from the backlog.

Sprint. This is the time period in which the development team implements the user stories chosen in the sprint planning. A sprint usually lasts for two weeks but it can be shorter or longer depending on the needs of the product owner and the goal that is being tackled. It is highly recommended as a good practice that sprint length remains consistent throughout the whole development so the team can learn how to better manage the workload in that time frame and even help plan the next sprints better.

Daily scrum. Daily scrum or just daily is a very short activity where the development team has a meeting and discusses what they did the previous day, what are their goals for the day and any obstacles they might have encountered that might be mitigated by someone else on the team. This activity should last around 15 minutes. On special request the product owner might participate as a viewer in one of this meetings but this is not the norm although sometimes it might help to quickly solve any doubts the development team might have about the product or user story.

Sprint review. After each sprint is concluded, the team meets for an informal demo of the sprint, also to inspect for possible flaws. After this meeting the team informs the stakeholders of the completed goals and does another, now formal, demonstration of the increment in order to obtain feedback and approval by the product owner on whether or not the increment is released.

Sprint retrospective. A team meeting where each member discusses what went right and wrong not only in regards to the project and sprints but also about people, relationships, tools or even ceremonies. The aim of this meeting is for the team to reflect on the sprint and discuss means to improve for the next one.

2.2.4 Kanban

The Kanban framework [45] is an approach that focuses on the creation of products with accentuated importance on the continual delivery so as to not overburden the development team. Kanban works by using the "just-in-time" manufacturing process principles

to match the amount of work in progress to the team's capacity, this provides more flexible planning options, faster output and more transparency throughout the development cycle. Kanban was widely adopted by software developers because once the team understands its basic principles they can start applying them with little to no overhead.

Kanban boards. Kanban boards are the centerpiece of the Kanban approach. All of the work that has been done, needs to be done or is in the development stage is represented on the board. The board is a tool that allows for workflow optimization amongst the team and quick visual perception of the blockers and dependencies so that these can be promptly solved. While physical boards are popular and are adequate to visually represent the team's work, virtual boards are crucial for their traceability and accessibility from multiple locations. A basic Kanban board has three groups, each a workflow stage, "To Do", "In Progress" and "Done", the number of workflow stages is flexible and can be altered to better fit a team's needs and wants, fitting their unique process and team resources and constraints.

Kanban cards. In kanban every work item is represented through a card that goes on the board. These cards have important information about that particular work item such as: who is responsible for it, how long should it take, a description of the work to be done, and others. Virtual cards though have the benefit of allowing more information and as such screenshots and other technical details are also assigned to the card.

2.3 Sustainability Engineering

Sustainability is about finding a way to meet the needs of the present without compromising the ability of future generations to meet theirs [16]. With this definition we can understand that sustainability is crucial for humankind's survival and longevity on this planet, and as such we must do our best efforts to achieve it and leave a better world for future generations. As with most complex problems the solution is also complex, and with sustainability the problem worsens with the fact that it is an issue on a global scale and we all need to work together to overcome its obstacles and barriers, not only through individual effort but also through nationwide effort. There are three main sustainability dimensions [3], them being:

- **Environmental sustainability** in a broad way is related to the amounts of pollution that can be created, renewable resources that can be harvested and nonrenewable resources that can be depleted in an indefinite way without compromising the ecosystems and their natural balance.
- **Social sustainability** in a general sense is the capability of a social system to function at a set level of social well-being indefinitely.

- **Economic sustainability** is the ability of an economy to maintain a determined amount of economic production indefinitely.

2.3.1 Environmental challenges

Environmental sustainability is one of the most important if not the most important sustainability dimension, without which the remaining can not flourish because they will always be dependent on the environment. That said there are three major problems that need to be tackled and solved or at least assessed in the way they influence our world. Those problems are: global warming, water scarcity and waste management.

Energy Consumption and Global Warming

Although the steady rise of energy consumption is an issue, it is not very concerning in itself; the problem lies with the production of said energy. If all energy was produced through renewable energies we would not need to worry about this, but since only a quarter of it is achieved through green and sustainable methods [48] this leads to a big challenge: global warming. Global warming is the increase of the average temperature of the planet. This comes with some catastrophic effects such as the rise of sea level due to the melting of polar ice caps, the increase of desert area and the acidification of the oceans [19]. The biggest cause for the increase of global warming is the hole in the ozone layer, expansion of which is influenced by the release of greenhouse gases into the atmosphere creating a greenhouse effect. This all comes back to energy production since this industry is the biggest contributor of greenhouse gases emissions.

Water Scarcity

With only 3% of the world's water being fresh water and with only a third of that being available for our use this leads to a dire situation, where many people's access to water is compromised or non-existent. It is estimated that roughly 1.1 billion people in the world have no access to water and an astounding 2.7 billion struggle to get water at least one month of the year [42]. Furthermore, water sources that are necessary to maintain ecosystems and habitats and are also used for human consumption have become stressed with the increasing demand. This coupled with the fact that rivers, lakes and other fresh water bodies are drying up or becoming too polluted to use and that climate change is causing shortages in some areas and floods in others, due to the alteration of natural patterns of weather and water. Water scarcity was identified as the largest global risk in terms of potential impact over the next decade by the world economic forum [41].

2.3.2 Sustainability in Software Engineering

A system's sustainability is directly related with how well it is able to adapt to the ever changing environment it is inserted in, while continuing to function correctly. In general,

sustainability is linked to environmental issues, but as we saw this is not true, since we also have to keep in mind the social and economic aspects as well in order to achieve true sustainability. When talking about software or technological sustainability there are two similar sound, yet very different, terms that are important to know: "sustainability by IT" and "sustainability in IT" [8]. Sustainability by IT is when a system is specifically designed to educate about sustainability or aid in achieving some sustainability goals. On the other hand, sustainability in IT is when the term sustainability is used to describe or evaluate the components of IT itself, them being software or hardware. More often the definitions presented for sustainable software are a mix of the two terms and as such we can define it "as software, whose direct and indirect negative impacts on economy, society, human beings and environment that result from the development, deployment and usage of it are minimal and/or which as a positive effect on sustainable development" [23].

A software can only be truly green and sustainable if it comes from a green and sustainable process, therefore we can say it is critical that everyone involved in the development process has to be aware of the impacts, both positive and negative, their software can have on sustainability. Thus, we can define green and sustainable engineering as "the art of developing green and sustainable software with a green and sustainable software engineering process, i.e., the art of defining and developing software products in a way, so that the negative and positive impacts on sustainable development that result and/or are expected to result from it, over its whole life cycle, are continuously assessed, documented and used for a further optimization of the software product" [23]. As said previously, there is more than one dimension of sustainability, and when talking about software sustainability we extend the set of already mentioned dimensions by two with [3]:

- **Technical sustainability** focuses on the longevity of a system and their capability to adapt to the evolving environment conditions and requirements.

- **Individual sustainability** covers topics such as individual freedom, the ability for individuals to exercise their rights and rights to fulfillment and free development.

The more complex the system is, the more probable it is to affect one, if not all, of the five dimensions, especially when considering that many of the dimensions are connected through common goals and requirements. Therefore, all trade-off possibilities must be discussed and analyzed before any changes to the system can be made, but with the topic of software sustainability being broad and understudied this process can become quite hard. Nevertheless in [4] a list of principles for sustainability in regards to software engineering were presented:

- **Sustainability is systemic,** meaning a system can never be analyzed without regards for the environment it's inserted in.

- **Sustainability is multidimensional,** there are at least four sustainability dimensions: economic, social, environmental and technical. Individual sustainability is also considered by some authors.
- **Sustainability is interdisciplinary,** sustainability concepts in software design cannot come from software engineering alone and need to adopt concepts from other disciplines.
- **Sustainability transcended the software's purpose,** because software's impacts extend beyond the scope of the problem it is trying to solve affecting the social, economical and environmental ecosystems it is inserted in.
- **Sustainability is multilevel,** since during development we have to consider two different perspectives at the same time, i.e., the system being developed alongside its sustainability goals and requirements and the wider system it will be inserted in.
- **Sustainability is multi-opportunity,** it requires us to look at all possible trade-offs before altering the system in order to get the most out of the changes.
- **Sustainability involves multiple timescales,** it is necessary to think short as well as long term in order to predict and tackle sustainability goals effectively.
- **Sustainability is not a zero-sum,** meaning that changes made to support a long term sustainability goal do not necessarily impose sacrificing something in the present time.

All this said, there is a clear lack of theoretical understanding and practical methods to apply sustainability to software and its development process [12]. The need for action is urgent and taking self-responsibility in developing critical thinking about this topic is necessary; but this is not enough: there needs to be a bigger presence of sustainability discussion and goals in education to cultivate a mindset of change in future software designers. On a corporate level, there needs to be more workshops to educate the developers, leaders and clients alike about sustainability, its issues and its importance.

2.4 Summary

Requirements engineering is one of the most important stages of software development. It is composed of five major activities: elicitation, analysis and negotiation, documentation, validation and management. For our work we deepened our understanding of the elicitation, prioritization and trade-off analysis. We discussed some techniques associated with these processes techniques and discussed the advantages and disadvantages of each one. Agile is an up and coming methodology for software development and project management. It is highlight by its focus on development through continuous iterations,

team collaboration and early delivery. This leads to a process easily adaptable to changes. User stories are a key part of agile development. They consist of a brief and informal explanation of a system feature written from the perspective of its end user. Due to the importance of user stories we also studied the rules and recommendations for writing one. Two of the more used variants of agile are Scrum and Kanban. As such we presented the basics of their ideologies, their events and the ways they approach the product backlog and the user stories. Focusing on sustainability we explained the main environmental challenges (energy consumption, global warming, water scarcity and waste management) and highlighted some of the threats they present. Concerning software sustainability, we presented the differences between "sustainability in IT"and "sustainability by IT". We also pointed to the fact that sustainability is only about the environmental dimension, but also about the economic, social, technical and individual ones. The previous five dimensions are interrelated. To finish off we listed the various principles of software sustainability.

3

State of The Art

As a means to correctly evaluate the state of the art for the topic being researched a systematic mapping study was done. A detailed presentation of all the phases that entail this process will be addressed in this chapter along with the results and conclusions that resulted from it.

3.1 Systematic mapping study

As said previously a systematic mapping study was conducted in order to fully assess the state of the art on sustainability in requirements engineering, more specifically on sustainability applied to agile development methods, and generate a summarized report of the results found. Usually the process starts out with the researcher picking out one or more research questions and the collecting information relevant to each of those questions through query strings used in digital libraries, like DBLP. Once all the relevant data is extracted, it is analyzed against the research question it is pertinent too. The individual phases that are part of this systematic mapping study process will be explained in detail ahead in this section.

3.1.1 Research methodology

A systematic mapping study consists of three phases: a planning phase, a conduction phase and a discussion phase. The planning phase has the objective of delineating and selecting the research questions and defining the study selection strategy and the data to be extracted from each of the selected studies. The conduction phase is for presenting the execution of the search for each research question along with their respective results. Lastly, the discussion phase aims to present a detailed analysis while using the results obtained from the previous phase. The methodology used for the search was to gather all of the documents that were relevant to green or sustainable software development, requirements engineering or agile, or some of its popular approaches, i.e. Scrum and Kanban.

The search string, "(sustain | green) (agile | scrum | kanban | (requirement engineering))", was ran through DBLP only since it encompasses a great majority of the relevant libraries for Computer Science and Requirements Engineering, such as IEEEXplore, Science Direct, and SpringerLink, and due to the overwhelming number of results on other platforms like ACM Digital Library and Google Scholar that would be impossible to read and analyze in order to figure out which ones were relevant to the study. There was also a high number of false positives.

3.1.1.1 Planning phase

The focus of this first phase in the mapping study is to formulate the research question and its respective search string, that will be used later in the digital library platform. Afterwards we have to define the inclusion and exclusion criteria for the results of the search and selection of studies. Once the search is concluded a quick assessment of the documents' quality is done in order to retain only the most relevant ones. Finally, we need to define which information is relevant to extract from each of the selected studies. Next, all of these phases will be presented in detail.

Research questions. The definition of our goal for the study is based on the PICOC structure[47] as presented in Table 3.1. The PICOC analysis method is used to describe the five elements of a searchable question.

Table 3.1: PICOC Analysis

Population	Papers addressing the integration of sustainability in agile methodologies or any implementations of sustainability frameworks for RE.
Intervention	By searching on electronic libraries using specific queries containing relevant keywords.
Comparison	Given a set of criteria and by understanding the main characteristics of the existing approaches for assessment, prioritization and trade-off analysis, compare them and choose the ones that are better or fit better in this context.
Outcome	Overview and analysis of the state of art.
Context	Preparation for the Master Thesis on Informatics Engineering.

This method led us to the following two research questions:

"What are the existing works of sustainability in agile methodologies?"

"How is sustainability addressed in requirements engineering in agile methodologies"?

The search queries developed for the research questions are presented in Table 3.2.

Table 3.2: Search query building for the research questions

Research Question	Sustainability	Requirements	Methodology
What are the existing approaches of **sustainability** in **agile methodologies**?	(sustain | green)	N/A	(agile | scrum | kanban | user stor)
How is **sustainability** addressed in **requirements engineering** in **agile methodologies**?	(sustain | green)	(requirement engineer)	(agile | scrum | kanban | user stor)

Relevant information about the notation used in the query construction is that the symbol "|"represents an "OR"and the different groups of terms are represented in different columns and connected with an "AND"operator; the highlighted words in the research questions represent the keywords used to formulate the search query.

Search Strategy. The search strategy used was supported by two different methods: automatic and manual. The automatic search consisted in using the created search query, from the Table 3.2, and running it through the digital library. After the automatic search was concluded, a manual search was performed by combining different keywords and running them through the same digital library and also Google Scholar to make sure we have not missed any relevant document. Also, snowballing was used to identify relevant studies based on the reference list of each selected primary study.

Study selection strategy. To make sure we only selected relevant articles for this study, we defined criteria for article inclusion and exclusion. This inclusion/exclusion process was done in two phases: the first phase consisted in analyzing the title and abstract of the documents, this allowed us to quickly remove irrelevant documents from our list of candidates; in the second phase we performed a full reading and analysis of the articles contents. The documents that were not excluded from the second phase are the final articles used for data extraction in this study. The inclusion criteria for the first and second phase are presented in Table 3.3 and Table 3.4 respectively.

Assessing the quality of the studies. The quality of the articles is a top priority, since higher quality articles lead to more accurate and reliable results. For each study we used the number of citations presented in the digital library as a quality indicator since a study with higher citation count is probably more trustworthy than a study with fewer citations, that said the year of publication plays a role in this value.

Data collection and extraction. In order to increase the readability of the relevant data and facilitate the extraction an excel worksheet was created. In said worksheet a sheet was

Table 3.3: Study inclusion and exclusion criteria for the first phase

ID	Study inclusion criteria
IC1.1	The article is available in English
IC1.2	The article is from a conference, workshop or journal
IC1.3	The article is from year 1987 to the point of conducting the search (2021). 1987 was chosen as the starting year because it was on this year that the topic "Sustainable development" was approached on United Nations
IC1.4	The article is available in full text
IC1.5	The title and/or abstract of the article is related to any of the research questions
	Study exclusion criteria
EC1.1	The article is a duplicate
EC1.2	The article is not in English
EC1.3	The article is informal (slides, extended abstracts, blogs)
EC1.4	The article is older than 1987
EC1.5	The title and/or abstract does not relate to any of the research questions

Table 3.4: Study inclusion and exclusion criteria for the second phase

ID	Study inclusion criteria
IC2.1	The article content discusses and/or answers any of the research questions in a direct way, i.e., if it's a summary, a comparison, a validation or a proposal of different approaches regarding the research question
	Study exclusion criteria
EC2.1	The content of the article does not answer any of the research questions
EC2.2	The article mainly discusses challenges and problems in this domain but does not provide any beneficial solution or suggestion to solve such problem

created for each of the final papers selected. The sheet includes some basic information about each paper like the title, the number of citations, the year of publication and a section for data pertinent to each of the research questions categories. The template for each sheet can be found in Table 3.5.

Protocol review. A systematic mapping study should have a thorough review process, usually conducted by external entities in order to assure impartiality. However, given the time constraints, this was not possible, and the evaluation of this study was done only by the advisers of this dissertation.

3.1.1.2 Conduction

The conduction phase of the study was performed by running the search strings from Table 3.2 on DBLP. After the searches, the papers were selected based on the title and abstract according to the inclusion and exclusion criteria on Table 3.3. Then, the papers

Table 3.5: Data Extraction Summary Table

Data Extraction	
1. Paper Basic Information	
1.1 Paper id:	DBLP:conf/greens/DickDKN13
1.2 Paper title:	Green software engineering with agile methods
1.3 Paper conference/journal/workshop:	2nd International Workshop on Green and Sustainable Software
1.4 Paper year:	2013
1.5 Paper number of citations:	14
1.6 Digital library	IEEE Xplore
2. Sustainability in Agile	
2.1 What is the approach regarding sustainability in agile proposed by the paper?	The paper proposes alterations/improvements to the current agile life cycle by introducing and modifying agile events that introduce sustainability criteria, monitoring and implementation into the general agile life cycle
2.2 What is the approach regarding sustainability in agile requirements engineering?	N/A

Table 3.6: Automatic search exclusion synthesis

	1st Sub Query	Main Query
1st Matches	0	45
EC1.1	0	0
EC1.2	0	-1
EC1.3	0	-1
EC1.4	0	0
EC1.5	0	-39
2nd Matches	0	4
EC2.1	0	0
EC2.2	0	-3
Final Total	0	1

that were selected from the first phase were fully read while applying the second iteration of inclusion and exclusion criteria, from Table 3.4. Finally, after the final papers were selected, the data and relevant information was extracted from the documents and added aforementioned excel worksheet. This stage was done on February 5th, 2021. The following Table 3.6, presents a summarized view of the number of papers encountered and excluded due to each criteria along with the final values for the conduction. From Table 3.6 we can see that from the total number of 45, the original number of documents found through the automatic search, we ended up with only 1 papers. To make sure no study was missed, as mentioned before, a manual search was also conducted. From this manual search no more studies were found. Therefore, we ended up with a total of one final article chosen.

3.1.2 Discussion of the results

After completing the search and extracting all the relevant data, we revisited our research questions and answered them and discussed the results from the study. The discussion will be based on the final papers selected. An important thing to notice is that no documents were found regarding the first research question, only for the main one, regardless the sub question will still be discussed in the following subsections. This section is subdivided into subsections, one for each of the research questions where the discussion and data analysis will be presented.

Research Question 1 - How is sustainability addressed in requirements engineering in agile methodologies?

As presented in Table 3.6 we can see that no documents were found regarding this sub question. Such a result means that little to no work has been done regarding these three topics together, sustainability, requirements engineering and agile. A justification for this could be the fact that software sustainability and agile methodologies are still in their infancy so their understanding by the scientific community is still diminished. That accompanied with the fact that the question relates those two topics with requirements engineering produces an even less understood and underdeveloped combination of topics.

Main Research Question - What are the existing approaches of sustainability in agile methodologies?

Even though the search query for this question resulted in quite a few documents, 45 to be exact, only 4[15, 30, 33, 32] of those made it into the second phase of document filtering. A reason for this was due to a high amount of false positives, like the authors name including "green", and many of the documents not being related to software development or design but to other areas like resource management and assembly lines. Furthermore from the 4 second phase documents only 1 made it into the final stage of analysis and data collection, since two of the documents excluded were about the success factors and risk factors for sustainability in agile development and didn't discuss any way to achieve or avoid them. The other document only talked about a model to evaluate "the agile maturity of GSD vendors, in the context of green and sustainable software development"[30] but no way to actually implement this model or how to use it, hence not providing a solution to carry out sustainability in agile software development methods. The fact that only one document was selected as a final data source leads us to conclude that there is much research to be done in relation to sustainability related to agile software development practices. Nonetheless the one document that was accepted[15] proposes a process with the intention of integrating it into the regular agile process. This process would be in opposition to one that would run parallel to it, even though this is an acceptable solution,

this way reducing the entropy caused by the new process in already existing events and activities that make up agile software development. So to answer our question there are no implemented nor studied frameworks or models to integrate sustainability in the agile process and only one proposed untested method of doing so was found.

3.1.3 Threats to validity

Every study comes with its own threats to validity and ours is no exception even though measures were taken in order to mitigate those threats. In this study a significant threat was the accuracy of the automatic search. To combat this we tried optimizing our query strings to be as accurate and complete as possible. Nonetheless studies can still be left out if their titles don't contain one of our search terms. So to minimize this effect we also performed a manual search with different combinations of the search keywords, making sure the least number of relevant papers would be left out. Since we also excluded a study that was not in English, we could have also missed some important data there. We also used the snowballing method on reference list of the final selected primary study but no useful documents were found. Lastly since the selection of the documents was made by one person alone there could have been selection bias during the process. To aid in this regard an inclusion and exclusion criteria list was created.

3.2 Related work

Due to the underwhelming number of final results found during our mapping study we decided to investigate studies in areas similar to ours. The analysed documents mainly focused on the requirements engineering process and based themselves around the already existing processes and models. This was done to reduce the disarray that comes from introducing a new process into the development cycle and minimize the gap in RE practices between academia and industry. The proposed approaches tend to use well known artefacts, like metamodels, checklists, goal models and feature models, and adapt them to fit the concepts and criteria of sustainability.

Pensenstadler [27] introduces a theoretical example where from a requirements engineering elicitation process, using Domain Models, System Vision and Goal Models, one can derive use cases and their scenarios, which in turn can be transformed into sustainability oriented user stories.

The approach described by Brito et al. [6] presents a concern-oriented metamodel adapted to include sustainability as its primary focus. The solution this paper presents has as a goal identifying concerns, sustainable or not, and handling the conflicts between them.

The goal of the work by Paech et al. [25] is to support the elicitation of sustainability requirements by providing "a checklist of general and IT-specific details for the sustainability dimensions and a checklist of general influences between the dimensions" along

with a process. The process is used to iteratively refine the requirements of a system with sustainability considerations by identifying the needs and effects from each dimension based on the two checklists. It is of note that the authors suggest that work still needs to be done to this approach before it can be used in a real test case.

Finally, the work by Saputri and Lee [34] was the most ambitious. It proposes an approach that covers sustainability not only during the elicitation process but all the way until functional design decomposition. This paper makes use of familiar requirements engineering methods, such as goal-question-metric, goal models, combined with a scenario-based approach, and feature models. The solution proposed by the article has already been put to practice with good results.

3.3 Summary

In this chapter, we performed a systematic mapping study to assess the state of the art on sustainability in requirements engineering, in particular on sustainability in the context of agile development. We discovered that sustainability in software development in an agile content is still a very immature and and ill researched area. As a consequence, there is still a lack of approaches with respect to software development in an agile content, despite the growing interest and need. More importantly, we know that our work is necessary and our focus is on contributing to the area of sustainable software development.

4

A Survey on Sustainability at IT Companies

In the previous chapter, we performed a systematic mapping study to identify works related to sustainability requirements specification methods and techniques in the agile context. However, the results indicated that very few approaches were used by agile developers. Given this, we decided to gather empirical evidence through a survey in the industry. Thus, in this chapter our goal is to describe our investigation on how sustainability requirements are being addressed in the agile software development industry in Portugal. We start by describing the design of this survey, followed by the description of the participants and the analysis of the results. We conclude with some major insights.

4.1 Survey Design

The main goal of this survey is to identify how agile practitioners (in Portugal) address sustainability when developing software. We carried out an online survey with companies that use agile methodologies for system development. We adopted Kitchenham and Pfleeger's guidelines for personal opinion surveys [20] and the activities to conduct online surveys in Software Engineering proposed by Punter [29]. Thus the activities managed here were: (i) define the study; (ii) design the survey; (iii) develop the questionnaire; (vi) execute the survey; (v) analyze the data and (vi) report the results.

Regarding the definition of the study, we want to know how do professionals that use agile development address sustainability requirements in their projects. Afterwards, we selected by invitation our participants, that come from different companies in Portugal. The set of participants invited play different roles in their companies (e.g., product owners, developers, testers, Scrum masters) and are experienced in different agile methods [44] (e.g. Scrum, Kanban), thus providing different viewpoints. The survey was sent to 52 professionals from several companies, from which 32 responded (a response rate of 61,5%). Participants were informed about the purpose of the research and data confidentiality.

The survey[1] was made available through Google Forms to the participants contacted via email. We divided the questionnaire into 2 main sections: (i) General information questions to collect some indicators about their knowledge and experience in agile development, and (ii) Sustainability in software development questions. Since all the professionals were from companies that adopted agile development (a selection criteria), there was no need to explicitly ask them about the kind of software development process they used.

The first 5 questions information about the participants. The second section consisted of 10 questions (being 8 multiple choice and 2 open-ended) to answer different aspects related to our research question: if they address sustainability during development and in which activities; in those activities how sustainability is addressed; how sustainability requirements are specified; their knowledge on sustainability in their development context; which tools and methods they used; their opinion on the impact of software on sustainability; the main challenges when addressing sustainability in their projects; and the main reasons to consider sustainability requirements in their projects. The use of spreadsheets helped with the quantitative analysis process, while an inductive approach was used for the qualitative analysis of the data.

The subsequent sections introduce the participants, and the layout of the survey together with the reasoning for each of the questions. In particular, we start with a biographical note of the participants (e.g. years of experience and development methods used in their work), followed by presenting the questions and discussing the specific results for each question.

4.2 Participants

This survey was answered by 32 professionals in the Portuguese IT industry. The participants work in different companies playing different roles in software development. Table 4.1 summarizes our participants information.

From the total number of respondents, 32 respondents, 34.4% are professional developers (11 out of 32), 12.5% are business analysts (4 out of 32), 12.5% are Scrum masters (4 out of 32), another 12.5% are project leaders/managers (4 out of 32), 9.4% are product managers (3 out of 32), and 6.3% are testers (2 out of 32). The remaining 4 roles correspond to tech leads, operations, UX/UI and product owners. Each of these roles has a 3.1% (or 1) presence in our surveyed population. The majority of the respondents (22 out of the 32 respondents, or 68.8%) have between 1 and 5 years of experience, meaning we have a fairly new and somewhat inexperienced group of professionals. Even though this could be seen as negative, we believe younger people may be more keen to learn about and address sustainability issues in software development.

[1] https://drive.google.com/file/d/1zc6hpvq6_Nn3D-wLtkL7Mfqc5FgIDfZj/view?usp=sharing

Table 4.1: Respondents individual information.

ID	Preferred Agile Methods	Role in the Company	Experience with Agile (Years)
P1	Lean Development	Business Analyst	1 - 5
P2	Development Method (DSDM), Dynamic Systems, Feature Driven Development (FDD)	Business Analyst	1 - 5
P3	Development Method (DSDM), Dynamic Systems, Feature Driven Development (FDD)	Business Analyst	1 - 5
P4	Scrum	Business Analyst	1 - 5
P5	Crystal, Scrum	Developer	1 - 5
P6	Scrum	Developer	<1
P7	Scrum	Developer	<1
P8	Scrum	Developer	<1
P9	Kanban, Scrum, Scaled Agile Framework (SAFe)	Developer	1 - 5
P10	Scrum	Developer	<1
P11	Scrum	Developer	1 - 5
P12	Scrum	Developer	1 - 5
P13	Scrum	Developer	1 - 5
P14	Scrum	Developer	1 - 5
P15	Extreme Programming (XP), Scrum	Developer	1 - 5
P16	Extreme Programming (XP), Kanban, Lean Development, Scaled Agile Framework (SAFe), Scrum	Operations	1 - 5
P17	Kanban, Scrum	Product Manager	1 - 5
P18	Kanban, Lean Development, Scaled Agile Framework (SAFe)	Product Manager	>10
P19	Kanban, Scrum, Scaled Agile Framework (SAFe)	Product Manager	6 - 10
P20	Scrum	Product Owner	1 - 5
P21	Kanban, Scrum	Project Lead / Project Manager	1 - 5
P22	Scrum	Project Lead / Project Manager	1 - 5
P23	Kanban, Scrum	Project Lead / Project Manager	1 - 5
P24	Extreme Programming (XP), Lean Development, Scrum Scaled Agile Framework (SAFe)	Project Lead / Project Manager	>10
P25	Scrum	Scrum Master	1 - 5
P26	Kanban, Scrum, Lean Development	Scrum Master	1 - 5
P27	Scrum	Scrum Master	1 - 5
P28	Extreme Programming (XP), Feature Driven Development (FDD), Kanban, Lean Development, Scrum	Scrum Master	6 - 10
P29	Dynamic Systems, Kanban, Scaled Agile Framework (SAFe), Scrum	Tech Lead	6 - 10
P30	Scrum	Tester	<1
P31	Scrum	Tester	1 - 5
P32	Kanban, Scrum	UX/UI	1 - 5

Other interesting statistics are that 93.75% of the respondents (30 out of 32) work with some variant of agile methodologies and that out of those 30 respondents, 28 work with Scrum (corresponding to 87.5% of the whole population). Finally, all the participants develop software using agile approaches.

4.3 Survey Questions and Results

4.3.1 Question 1. Do you address sustainability in software development?

This question aims to get an overview of how often sustainability is present in the projects developed by the participants. The options available for the participant to choose from are: never, rarely, sometimes, often, and always. Only one option could be selected. Figure 4.1 synthesizes the received answers. A significant part of the respondents 34.4% (11 out of 32) never address sustainability in their projects, meaning that 65.6% (21 out of 32) of them do address sustainability to different extents (even if only rarely). This is a positive aspect that leaves the impression that developers and companies are already moving towards the right direction. That said, it is not surprising that only 12.5% of the participants (4 out of 32) always address sustainability, since this is, after all, a new concept in software development.

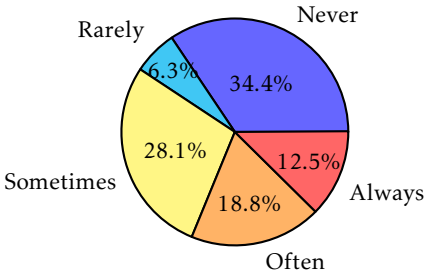

Figure 4.1: Addressing of sustainability in software development

4.3.2 Question 2. If you have answered yes, in which activities do you address sustainability?

This question is intended to narrow down and identify in which activity of the software's life cycle sustainability is addressed by our participants. In this question the respondent can choose from a number of different options namely requirements elicitation, software design, implementation, unit testing, system/integration testing, and maintenance. Here the participant could pick all of the options they wish. Figure 4.2 summarizes the results.

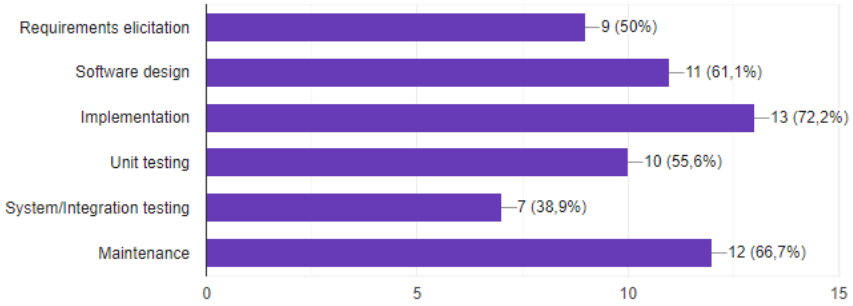

Figure 4.2: Activities where sustainability is addressed

Out of the 18 participants that reported in what activities they deal with sustainability, 86.6% of the answers do it in two or more activities. The top three most used activities are 'Implementation' with 72.2% (13 answers), 'Maintenance' with 66.7% (12 answers) and 'Software design' with 61.1% (11 answers) and out of the six activities used by the participants only 'System/Integration testing' reported a use of less than 50%. There is a discrepancy in the number of participants that answered this question (18) and the number of participants that said they address sustainability in software development in the previous question (21). This can be due to lack of a custom option where respondents can type what they want, or because 2 of these participants only have basic knowledge of sustainability and address sustainability in their projects "sometimes", and the other one only addresses it "rarely".

4.3.3 Question 3. If you selected any option above, how do you address sustainability in each of the selected activities?

This was an open ended question, aiming at further deepen the previous question to be able to understand how our participants addressed sustainability in each of the activities they selected previously. By analyzing the answers to this question we concluded that simple and easy to maintain solutions that are also capable of being adapted to the businesses needs are necessary to achieve software sustainability in our participants' perspective.

Another 'popular' factor relevant to software sustainability is the reduction of technical debt, where our participants use *"a set of practices that allow us to reduce as much as possible our technical debt"*. One of the participants also said that *"ethic behaviour of the team and the software"* is a key factor. It is also mentioned that *"continuous information sharing with all stakeholders"* are important to maintain everyone up to date and achieve a continuous flow of information and so the team is able to have time to properly discuss the trade-offs between the different system qualities. Thus, adaptability, technical debt, ethics in the team and software, inclusion of all to guarantee continuous flow for information, and trade-offs among different qualities, are some of the sustainability-related aspects raised by the participants.

4.3.4 Question 4. How do you specify sustainability requirements?

The goal of this question is to gather information on how our participants detail and specify the sustainability requirements necessary to address sustainability in their projects. Here the participant is presented with several options from which he can choose as many as needed. The options are: user stories, use case, backlog item, personas, acceptance criteria, part of the definition of done, informal text, UML models, none, and a space for free text where the participant could add a different option.

This question was analyzed in two ways: (i) regarding all of the 32 responses given and (ii) with the answers filtered to only include those who considered sustainability in their projects. The latter analysis left us with 21 answers (out of the 32). When analyzing

all of the responses we can see in Figure 4.3 that 75% of the respondents (24 participants) utilise user stories to specify sustainability requirements. This is to be expected since 30 participants work with a variant of agile and hence utilise 'User stories' regularly in their projects.

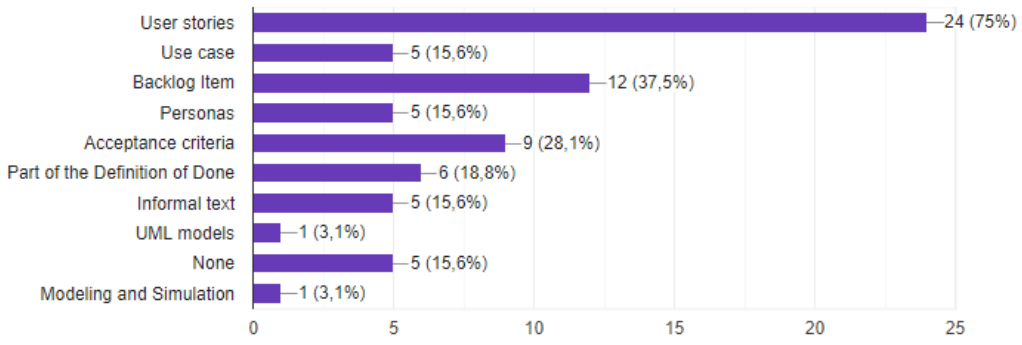

Figure 4.3: Sustainability requirements specification artifacts

On the other hand, when we filter out the participants that said they do not take sustainability into account, the story repeats itself (see Figure 4.4). Out of the 21 answers, 85.6% (19 participants) use "User Stories" to specify the sustainability requirements. This was once again expected since out of those 21 participants there are also 85.6% of participants that work with a variant of agile. This suggests that developers tend to specify sustainability using formats that they already known and use regularly.

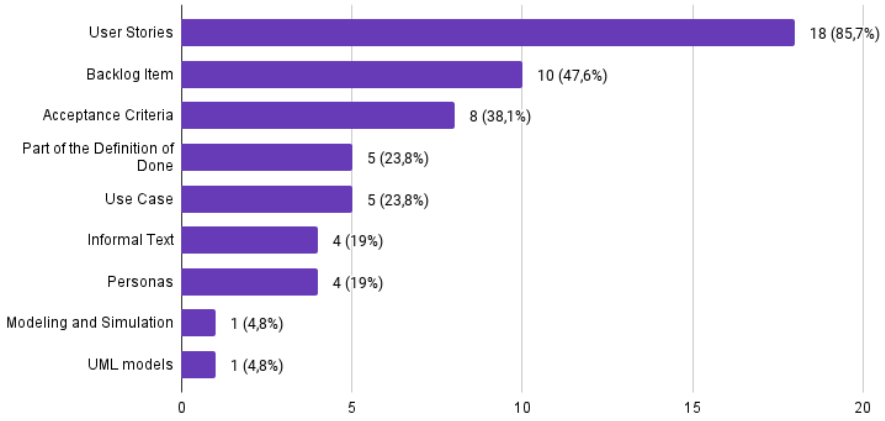

Figure 4.4: Sustainability requirements specification artifacts filtered

4.3.5 Question 5. Which of the following best describes your knowledge about sustainability in software development?

This question's objective is to ascertain the level of knowledge our participants know about sustainability in the software development realm. The available options to answer

this question are: no knowledge, basic, intermediate, advanced, and expert. Only one option could be selected. The analysis of this question is summarized in Figure 4.5.

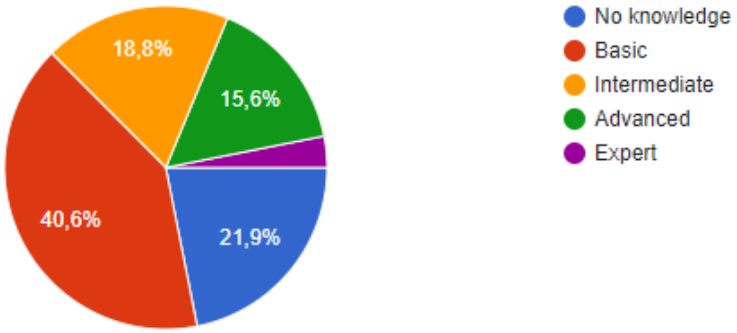

Figure 4.5: Knowledge about sustainability in software development

It was quite surprising to see that 78% (25 participants) of the respondents say they have at least basic knowledge about sustainability in software. Even though this is positive, 13 out of those 25 only have basic knowledge on sustainability. Furthermore, only 6 participants have knowledge at an advanced (15,6%, or 5 people) or expert level (3,1%, or 1 person). This number of highly knowledgeable participants is contrasted by the number of completely uninformed ones, 7 (21,9%). This leads to the conclusion that there is an overall lack of knowledge about sustainability in software development, indicating the need for significant investment to be made in training, both by academia and industry; practitioners have to be taught and coached on the concepts of sustainability, similarly to what is happening with agile/Scrum. This lack of knowledge can also be because universities are just now starting to include sustainability in their ICT curricula.For example, some universities (e.g., from UK [2], and USA[3]) already have sustainability in their courses, but we cannot say this is common in many institutions.

4.3.6 Question 6. Do you think you have tools or methods to be able to include sustainability in software development and maintenance?

This “yes"or “no"question aims to get the participants’ opinion on whether or not they have the appropriate tools to address sustainability in software development. We observed that 75% of the participants (24/32) report that they do not have tools or methods to deal with, or include, sustainability in software development. This is interesting as it supports our claim that sustainability in software development is still under researched and under developed.

[2]https://www.ucl.ac.uk/sustainable/education/embed-sustainability-curriculum
[3]https://sustainabilityinschools.edu.au/sustainability-curriculum

4.3.7 Question 7. If you answered yes to the previous question, what are the methods you use?

This question also aims to further investigate the preceding question and as such was designed with the purpose of discovering any tools or methods our participants use, or have used in the past, to address sustainability in their projects. This is important because it might uncover methods or tools that we might have missed during our research or any sort of custom methods they might have developed.

The responses given show that participants use several types of tools that are not necessarily designed for sustainability. For example, a participant that works with the OutSystems framework[4] said *"you could use Architecture Dashboard"* to help you view any potential "design smells" and opportunities to improve their software. Others use CAST[5] (software intelligence suite that focuses on benchmarking software quality and condition) tools, KIUWAN[6] (code quality analysis SaaS focused on security) and SonarQube[7] (quality of code inspection tool). And lastly, some use *"sustainability models and articles"* while others use *"modeling and automatic code generation"* to deal with sustainability. One of our participants also said that there are tools available but the clients are not usually willing to pay for them.

4.3.8 Question 8. How often do you think software has an impact on sustainability?

This question intends on gauging the participants perceptions on whether or not software has an impact on sustainability. This question has 5 options: never, rarely, sometimes, often and always. Only one option can be selected.

The answers to this question revealed that the number of participants that thinks that software always impacts sustainability is similar to the number of participants that thinks that software has no impact on sustainability — this number being 18.8% (6 participants). Other than that, it is good to see that 81% of the participants (29 participants), think that software has some sort of impact on sustainability even if not often (see Figure 4.6).

4.3.9 Question 9. In projects where sustainability was addressed, what were the main challenges found by the team?

With this question we wanted to identify the major problems developers face when addressing sustainability in their projects. This question has multiple choices and the participant can pick as many as needed. The options are: lack of support materials, lack of experts on the domain, lack of knowledge by the team, lack of methods, lack of tools, and none.

[4] https://www.outsystems.com/
[5] https://www.castsoftware.com/
[6] https://www.kiuwan.com/
[7] https://www.sonarqube.org/

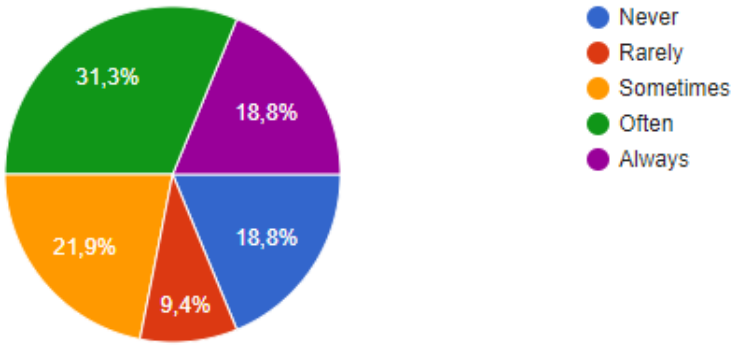

Figure 4.6: How often software impacts sustainability

There was also the option to choose a custom option where participants type what they want. The results from this question let us conclude that participants that try to implement sustainability in software encounter many types of challenges with high frequencies (see Figure 4.7). As such, we can safely say this area is very undeveloped and that there are still no easy ways to address sustainability in software development.

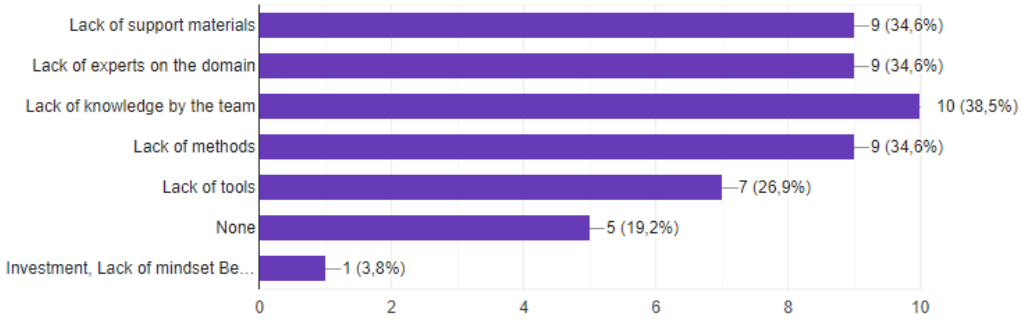

Figure 4.7: Challenges found with addressing sustainability

4.3.10 Question 10. What were the main reasons to consider sustainability requirements in your projects?

This question intends to discover what the mindset of our participants is on why they choose, or chose in the past, to include and implement sustainability requirements in their projects. The respondent can choose their answers from several options: client requirement, good reputation, legal obligations, social responsibility, organizational requirement, personal motivation, to improve the quality of the product and never was considered. Once again there was the option for the participant to add a custom option.

With this question we were able to gather the most important aspects, in our respondents opinion, to consider when addressing sustainability in a project (see Figure 4.8). The most important factor was *"To improve the quality of the product"*; this factor was chosen by 53.3% of our respondents, hence picked 17 times. In second place comes *"Good reputation"* and *"Personal motivation"* tied with 34.4%, picked 11 times each.

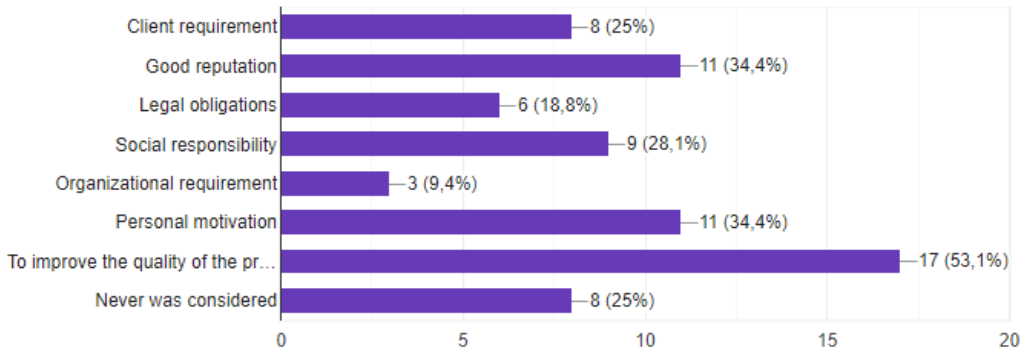

Figure 4.8: Reasons to consider sustainability in projects

4.4 Threats to validity

Internal validity. A threat to our questionnaire is that we might have asked the wrong questions, or at the very least constructed them poorly and making them ambiguous. To mitigate this, however, we structured the order of the questions and and made an effort to employ clear wording, by reviewing the formulation of the questions in several meetings among the authors.

Construct validity. Another threat is that we did not provide a definition for sustainability. Thus, each participant answered the questions with their own definition of sustainability in mind and as such could have failed to convey the information we needed. To mitigate this the questions were elaborated to be as objective as possible and reviewed by the senior authors.

External validity. Also related to the participants is the fact that we did not constrain the role of the respondents; this means that we might have data that are not representative of the overall IT industry. This was, however, defined in this way by design, to allow our participants to feel more relaxed and avoid bureaucratic permissions from the managerial level that could delay the study.

Conclusion validity. The data analysis process itself may also be a threat, due to the fact that wrong data analysis can yield wrong results and conclusions. So, to avoid this issue, we analyzed the data for each question individually through excel filters.

4.5 Other Related Surveys

In [5] the authors discuss the explanatory power of various sustainability indices applied in policy practice. Their results show that such indices fail to accomplish essential scientific requirements making them rather not useful if not misleading regarding policy advice.

In [31] it was identified that there was a lack of specific practices to be followed by global software development (GSD) vendors regarding the development of green and sustainable software. A study was carried on where agile practices were identified, aiming

to help GSD vendors to improve their agile maturity towards green and sustainable software development.

In [18], a study shows that practitioners consider software sustainability important, but more at a technical level, where environmental considerations are missing. The study revealed that the meaning of sustainability needs to be refined for specific project and application context.

Our study reinforces the above studies by also identifying awareness of sustainability by practitioners and complements the findings by pointing out the need to provide more sustainability requirements techniques and training on the sustainability area.

4.6 Major Insights

This section summarises some of the most relevant take away messages from the analysis of our survey. The study was limited to Portuguese professionals, where the results reveal the current situation of sustainability in agile development in Portugal, but it can serve as an indicator to be compared with companies worldwide.

Awareness about sustainability. Although limited, there is a growing interest in considering sustainability during software development (Q1). Indeed, even if the majority of the participants realise that software has some impact on sustainability, a relevant amount of practitioners do not share this opinion. Therefore, more awareness about the importance of sustainability in IT should be raised in industry (Q8).

Advantages of addressing sustainability. Participants gave us many reasons to consider sustainability in their projects, where the improvement of product quality is the main one, but good reputation, social responsibility and personal motivation were also highly mentioned. This is good news, since it is an indication that sustainability is a concern worth investing in (Q10).

Techniques for sustainability requirements. When sustainability is addressed, implementation and maintenance are the main activities where sustainability is considered (Q2). This suggests more effort is needed to promote sustainability as part of the requirements activities. On the other hand, it was not surprising that user stories are the most popular artifact (Q4). Participants seem to adapt user stories to help specify sustainability requirements; this would help the integration of sustainability in requirements activities.

Challenges to address sustainability in IT. Participants pointed out to several challenges when addressing sustainability in their projects, such as lack of tools, learning materials, methods, domain experts, knowledge (Q9). Indeed, a non-surprising finding reported by our participants was the lack of methods and tools to help eliciting and specifying sustainability (Q6). When used, some practitioners rely on quality of code and code generation tools to address sustainability, but not much was used for requirements (Q7). Finally, although many respondents reported that they have basic knowledge on sustainability, more training and knowledge dissemination on sustainability is needed (Q5). This is in accordance with [12] where a clear lack of theoretical understanding and

practical methods to apply sustainability to software and its development process is discussed. Indeed, the need for action is urgent and taking self-responsibility in developing critical thinking about this topic is necessary. But also there needs to be a bigger presence of sustainability discussion and goals in education to cultivate a mindset of change in future software designers. On a corporate level, there needs to be more workshops to educate developers, and clients alike about sustainability, its issues and its importance.

4.7 Summary

This chapter discussed the results of a survey we performed with a subset of the Portuguese industry aiming at identifying their current needs about sustainability as well as their practices when addressing sustainability in their agile projects. We were particularly interested in understanding how IT professionals elicit and specify sustainability requirements. To achieve this goal, we created a 10-questions survey aiming at gathering participants' information, their sustainability development issues (such as the impact of sustainability on their working environment), and the methods and tools they use to handle sustainability. The results helped us to design our approach described in the next chapter.

5

The Approach

Our solution consists of a set of pre-written general-purpose sustainability user stories supported by a framework based on piStar. This way we can offer a widely used agile development artefact, according to our survey, in the form of user stories and a goal-oriented diagram to help the user visualize and select the necessary sustainability qualities for the task at hand. Considering our proposed solutions aims to expand an already existing solution we only tackle four of the five sustainability dimensions that were already covered by the base solution. These dimensions are the Social Dimension, the Economic Dimension, the Technical Dimension and the Environmental Dimension, leaving out the Individual Dimension. In this chapter, we introduce the reader to the conceptualization of the developed solution, process and tool support, by explaining its context and integration with the piStar framework and its features.

5.1 Conceptual Model for the Approach

In this section we described the conceptual model of our approach. This model is the result of specifying the concepts related to sustainability, sustainability dimensions, requirements and user stories, and relating those with some of the elements of the iStar Language 2.0 metamodel [14]. The iStar model elements are adopted to express sustainability requirements and its conflicts. In the following sections we describe the conceptual model in detail, starting with the iStar model elements, followed by the sustainability concepts (dimensions and requirements) and ending with sustainability user stories concepts.

5.1.1 iStar Element, Goal and Quality

In this section, we describe the iStar model element component from the conceptual model. The *iStar Element* corresponds to the *Intentional Element* present in the metamodel for iStar 2.0 Language [14]. In our solution we only make use of two of the four intentional elements available in the iStar 2.0 Language and as such did not want to misrepresent the original definition for *Intentional Element*. The two *Intentional Elements* used are the

Goal element and the *Quality* element. These elements are used in the iStar language to represent "something that is desired or wanted"[14] by an actor and as such different types are needed to represent different types of requirements. In the iStar 2.0 language, **goals** are objectives for the system needed/wanted by an actor. These objectives are assessable in terms of their completion and have a clear set of achievements criteria. In our solution, these goals represent features that are needed to achieve a certain quality. For instance, 'Legislation' helps achieve 'Social Safety' and 'Justice' and it can be evaluated on its completion by, for example, checking if the necessary legal policies have been put in place. On the other hand, **qualities** represent an attribute for which an actor desires some level of achievement' [14]. Unlike goals, qualities can either have a specified level of achievement or not, having its achievement level "kept vague". In our solution, they represent the sustainability requirements like security and confidentiality. Both of these intentional elements are linked to each other with contribution links that have four types of contributions (break, hurt, help, make) that directly correlate to the influence the elements have on each other. These contribution types mean the following:

- **Make.** The source element has a positive influence on the target element and is enough for its accomplishment.
- **Help.** The source element has a positive influence on the target element but it's not enough for its accomplishment.
- **Hurt.** The source element has a negative influence on the target element but doesn't make the target unaccomplishable.
- **Break.** The source element has a negative influence on the target element and it's enough to make the target unaccomplishable.

5.1.2 Sustainability, Sustainability Dimensions and Sustainability Requirements

Here, we describe the Sustainability component from the conceptual model along with its relationships and dependents. In our model sustainability represents the overall definition and concept of sustainability, which, as provided in 2.3, is a way to meet the needs of the present without compromising the ability of future generations to meet theirs [16]. This sustainability concept is composed of five sustainability dimensions (environmental, social, economic, technical, individual). These dimensions are related to the sustainability concept indirectly through their generalization element, *Sustainability Dimension*. This generalization element is in turn the composing element of the sustainability concept. Also based on the sustainability dimensions, there are the sustainability requirements. These were designed to specify the various dimensions, by first breaking them down into smaller parts and then creating a requirement for each of those parts. The requirements can belong to one or several of the five sustainability dimensions and for this reason, they

are connected with the sustainability dimensions with a cardinality of 1 or more (1..*) both ways, meaning each requirement belongs to at least one sustainability dimension and each sustainability dimension is specified by at least one sustainability requirement.

5.1.3 Sustainability User Stories, User Stories and Relevant Stakeholder

Now we explain the concept of sustainability user story and its relations with the other components. *Sustainability User Story* represents our general-purpose sustainability user stories. In the conceptual model, it appears as a specialization of the known agile artefact. It represents the sustainability requirements created for software development. Because of this, the only direct relationships the sustainability user stories have in the conceptual model are the sustainability requirements and the user story.

To select the template in which to write the sustainability user stories, we analyzed the experimental results of an MSc thesis [11] whose focus was on researching the quality of user stories written in several templates. With that analysis, we gathered a better understanding of the pros and cons of the more commonly used templates for writing user stories and as such concluded that the most appropriate template is the Connextra (CON), which coincidentally is the most adopted format in SE [11]. Since, in our case, we are only interested in knowing how easy the user stories are to read and comprehend, we made our decision for the format used based on the data about that, not taking into account the data concerning the creation of user stories unless the other, more pertinent, data points were too similar and a deciding factor was needed. After deciding which data mattered most for us, we analyzed the research values present on [11] and chose CON because it was the one with the best comprehension time and a higher amount and percentage of relevant eye fixations while at the same time having the lowest time spent with irrelevant eye fixations [11]. The comprehension accuracy was nearly identical across all of the templates [11], so it was not relevant for decision-making.

To make these user stories truly general-purpose, a sort of surrogate stakeholder [26] had to be created to accommodate the uncertainty of the final user role of the user story. To try and solve this issue, we created the 'Relevant Stakeholder'. This will act as a temporary user role for the sustainability user stories until the user decides to change it to make it compatible with its needs. This relevant stakeholder component is present in the conceptual model linked to the sustainability user story with composition link, meaning it is part of the sustainability user story and it cannot exist without it, and multiplicity of 1 to 1, meaning each user story has one relevant stakeholder and each relevant stakeholder belongs to one sustainability user story.

5.1.4 Finalized Model

The conceptual model for our sustainability user stories was constructed as a UML class diagram, presented in 5.1.

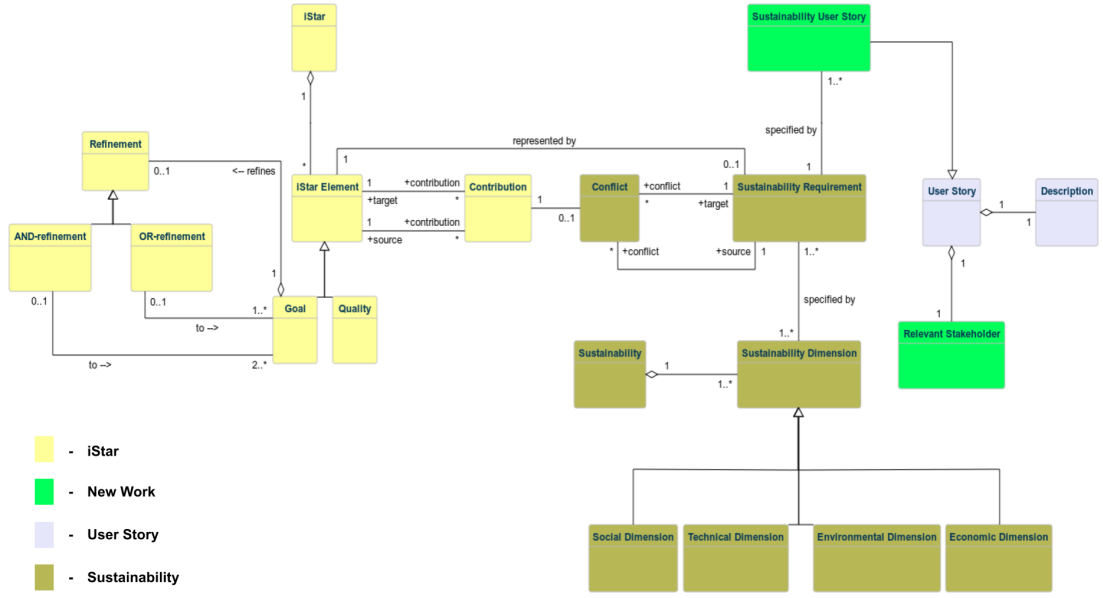

Figure 5.1: Conceptual Model

As shown in the figure, the different colours in this conceptual model represents different themes: olive green corresponds to sustainability 5.1.2, light yellow represents the iStar Language 2.0 metamodel 5.1.1 , blue represents user stories 5.1.3 and light green highlights the concepts of our developed work 5.1.3.

5.2 Tool Support

The conceptual model of our approach provides a representation of sustainability user stories along with their components and relations to sustainability requirements. This conceptual model was the first step into creating tool support that could provide developers with means to address sustainability in agile development. As described in the model, our framework makes use of the iStar language to represent our sustainability requirements. More specifically, it is based on a Sustainability Requirement Catalogue [2] built using the piStar framework. This catalogue offers the user the ability to choose out of the several main sustainability qualities available from the four different sustainability dimensions, also represented in the model. Our tool support consisted in expanding the already available sustainability requirements catalogue by adding new features that were necessary for the user to have better control over the sustainability features/qualities needed and to allow for the inclusion of sustainability user stories in the catalogue. These new features will be discussed in the subsections below.

5.2.1 Original sustainability catalogue and framework

To develop our tool support, we used as a base an already existing Sustainability Requirements Catalogue [2] built on the piStar framework. This way we could be sure we were using a good foundation for our tool and extending an already proven solution. The sustainability catalogue used is described in detail in a Master's Thesis [2] in computer science and was developed in piStar [28]. piStar is an online iStar modelling tool, that supports the iStar 2.0 language model and is lightweight and easy to use, running entirely on the browser without any prior installation. The catalogue developed in this MSc is organized in three major blocks of elements, this blocks are represented with different colours in figure 5.2. The first block of elements, shown in figure 5.2 inside the rectangle with the black outline, is the dimensions of sustainability that are addressed by the catalogue. The second block has the main sustainability qualities, shown in figure 5.2 inside the rectangle with the dark blue outline. These main qualities have contributions that link to the sustainability dimensions represented on the first block of the catalogue and/or to other main qualities. Lastly, the third block of elements is concerned with the refined elements, shown in figure 5.2 inside the rectangle with the red outline, these can either be qualities or goals and once again have contribution links, this time linking to the main sustainability qualities and/or amongst themselves.

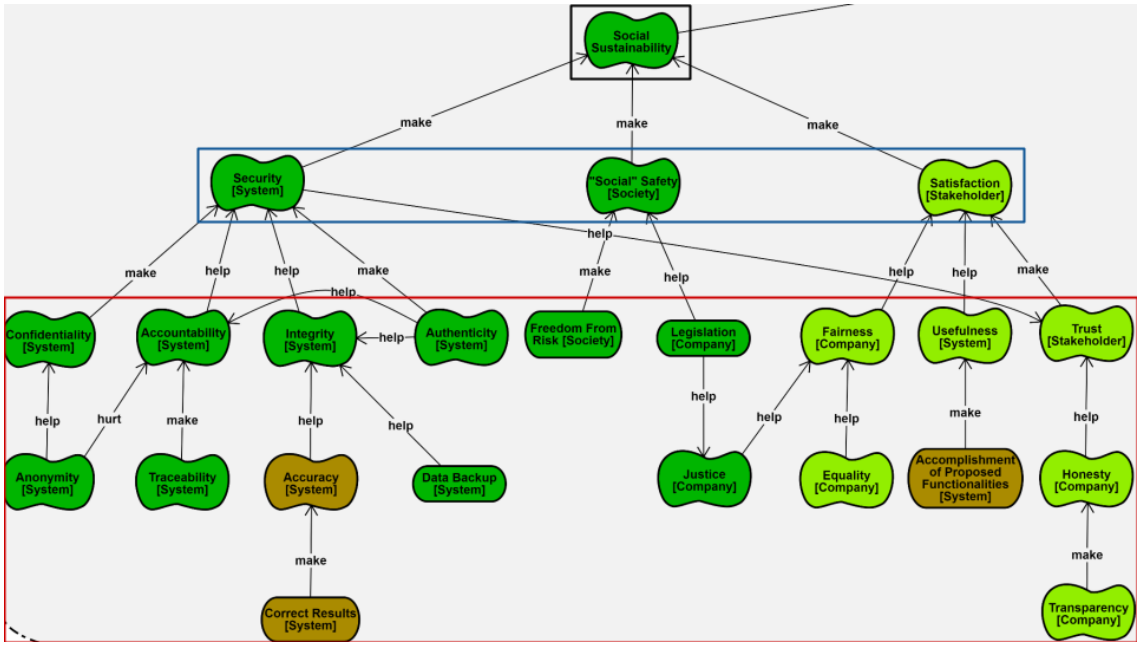

Figure 5.2: Catalogue block structure

All of the refined qualities have effects on other refined qualities and/or main qualities. To better represent this and make it easy for the user to grasp their effects, we introduced, in the sustainability qualities selection screen, a notation (similar to the one used by iStar language) using the plus sign (+) and the minus sign (-) followed by the name of the sustainability quality, an example can be seen in figure 5.3. In this example the current

sustainability quality positively affects the anonymity quality.

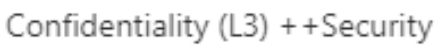

Figure 5.3: Notation example

The notation has four different levels of impact, and to better understand the notation, an example has been provided below. In that example, we will refer to the 'Confidentiality' quality as the influencer quality and the 'Security' quality as the influenced quality. Using our example in figure 5.3, we would acknowledge that the 'Confidentiality' quality has a 'make' impact on the 'Security' quality. The four different levels are the following:

- '+ +' represents the equivalent of a 'make' impact, meaning that the influencer quality is critical to the accomplishment of the influenced quality.
- '+' represents the equivalent of a 'help' impact, meaning that the influencer quality is helpful in the accomplishment of the influenced quality but not necessary.
- '-' represents the equivalent of a 'hurt' impact, meaning that the influencer quality is harmful to the accomplishment of the influenced quality but not an impediment.
- '- -' represents the equivalent of a 'break' impact, meaning that the influencer quality is an impediment to the accomplishment of the influenced quality and in direct conflict with the accomplishment of the influencer quality.

5.2.2 Tool Features

This section describes the main functionalities provided by the tool, that includes the catalogue loading, sustainability quality selection, user story editing, among others.

Full Catalogue Loading → As suggested, this feature consists of loading the full sustainability catalogue along with the general-purpose sustainability user stories for all of the catalogue elements. The user stories will be loaded with a placeholder user role that can be altered afterwards with the use of another feature.

Sustainability Quality Selection → For the user to generate the sustainability user stories, s/he first is required to select the qualities needed for the project. For that purpose, the framework allows the user to do a granular selection of the sustainability qualities through the use of a menu comprised of four tabs, one for each sustainability dimension. Those tabs in their turn contain the main qualities for the dimension it regards to and the respective refined qualities. The granularity comes from the fact that the user can select any of the sustainability qualities individually instead of by main quality like in the original sustainability catalogue. Along with this selection process, there is a warning system that notifies the user whenever two conflicting qualities are selected.

This notification comes in the form of a pop-up notification to immediately warn the user and also creates a conflict window where all of the current conflicts are stored.

The way this selection screen works is by only presenting the user with the main sustainability qualities at first and gradually giving the user more options as elements are selected. This selection feature happens through the use of a pop up window that is divided into 4 tabs, each corresponding to a different sustainability dimension. Each of these tabs has within it the main sustainability qualities that belong to the tab's dimension. Alongside these main qualities there is a check box. By clicking the check box the user is selecting this quality and at the same time making visible the next level of refined qualities that are related to the main quality.

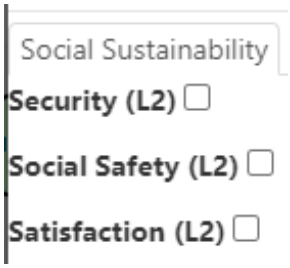

Figure 5.4: Sustainability Quality selection window in the Social Dimension tab with no selected qualities

For example, Security is a main quality in the social sustainability tab and it belongs to level 2 in the catalogue hierarchy. By clicking the security check box we made visible the next level of qualities that can be selected, in this case, 'Confidentiality', 'Accountability', 'Integrity' and 'Authenticity'. These qualities that are now visible can be selected at will by the user and they may enable the selection of further refined qualities if such exist related to the selected qualities.

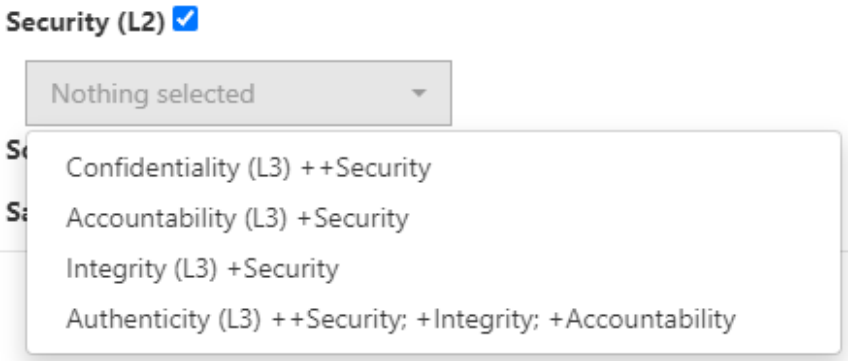

Figure 5.5: Sustainability Quality selection window in the Social Dimension tab with a checkbox ticked and next level visible

User Role Replacement → To ease and speed up the user's customization and adaptation of the sustainability user stories to the task at hand, the tool provides a feature that

replaces user roles in the selected user stories. This is done semi-automatically, meaning it needs only to be given the user's input on the desired user roles for each of the main qualities selected, i.e. security. Subsequently, the user roles provided by the user are propagated to all of the sustainability user stories that refine each of those main qualities.

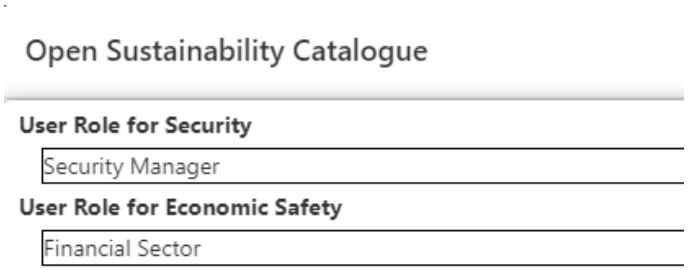

Figure 5.6: Example of the user role replacement window with user roles

User Story Manual Editing → To account for further needs and extend the adaptability of the framework. The framework allows the user to manually edit individual user stories and fine-tune them to his/her liking. This editing allows the user to edit all of its contents and not just the user role, unlike the previous feature. With this freedom, the user stories can be completely rewritten and because of that, we are not accountable for the quality of the resulting user stories. This said the user can always revert to the original version of a user story by reloading the catalogue.

Saving Progress → A simple yet important feature is the save feature. This is an extension to the basic save feature of the iStar framework that allows the user to save a picture of the current state of the catalogue diagram and a text file with the current state of the model. This enables the user to load the catalogue at a later date without having to re-do all of the work. Our tool extends this feature by also creating a text file with the current state of the sustainability user stories derived, and the conflicts between them, from the currently selected sustainability qualities.

5.3 The Process

The process we created in order to derive sustainability user stories with our tool support is represented in diagram 5.7. We will describe the steps that are a part of the process using the diagram as a reference.

First User Interaction The process begins once the user loads the tool in his/her browser. After that, the user has five options to choose. These choices are represented in the first part of the diagram, as seen on the close up on figure 5.8, and are the following:

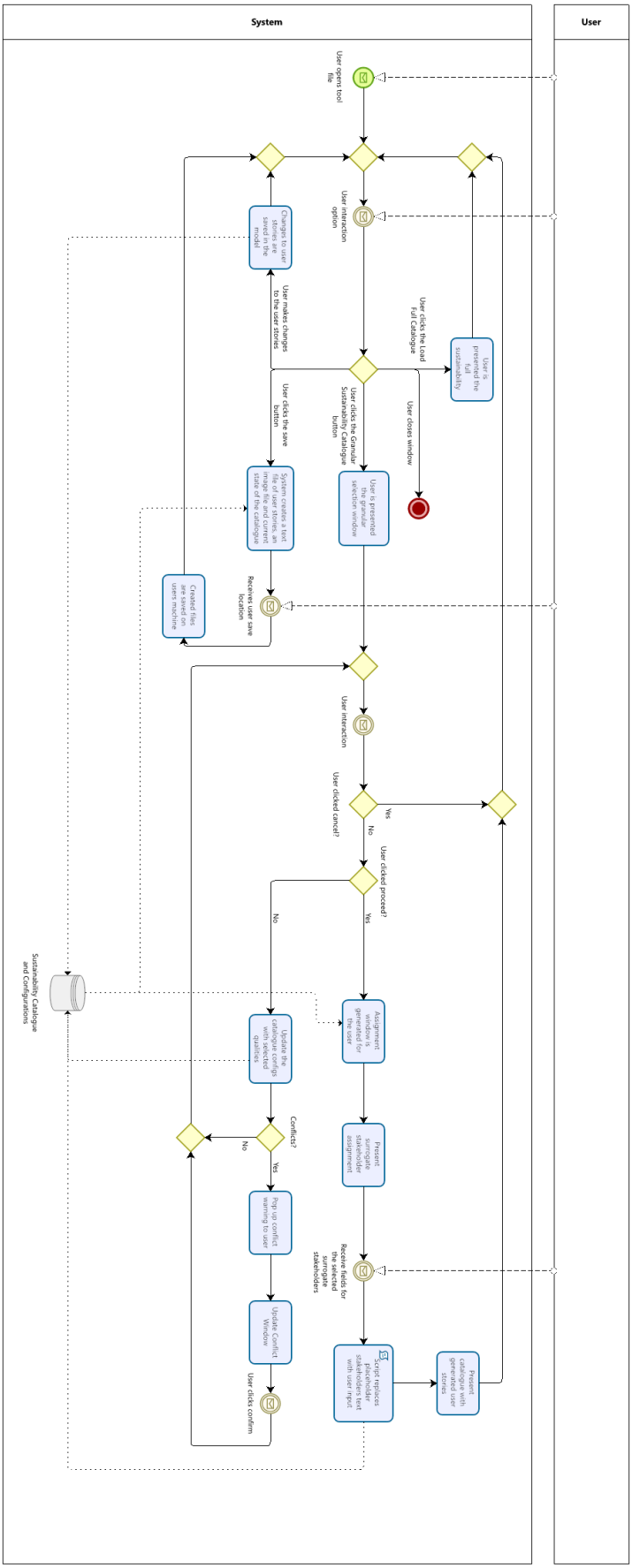

Figure 5.7: Diagram of Sustainability User Story derivation process

- the user clicks on the granular sustainability catalogue button commencing the sustainability qualities selection;
- the user clicks on the load full catalogue displaying the complete catalogue on the screen;
- the user clicks on the save button downloading the three progress files as a result;
- the user makes changes to individual items on the catalogue;
- lastly, the user can simply close the window ending the process.

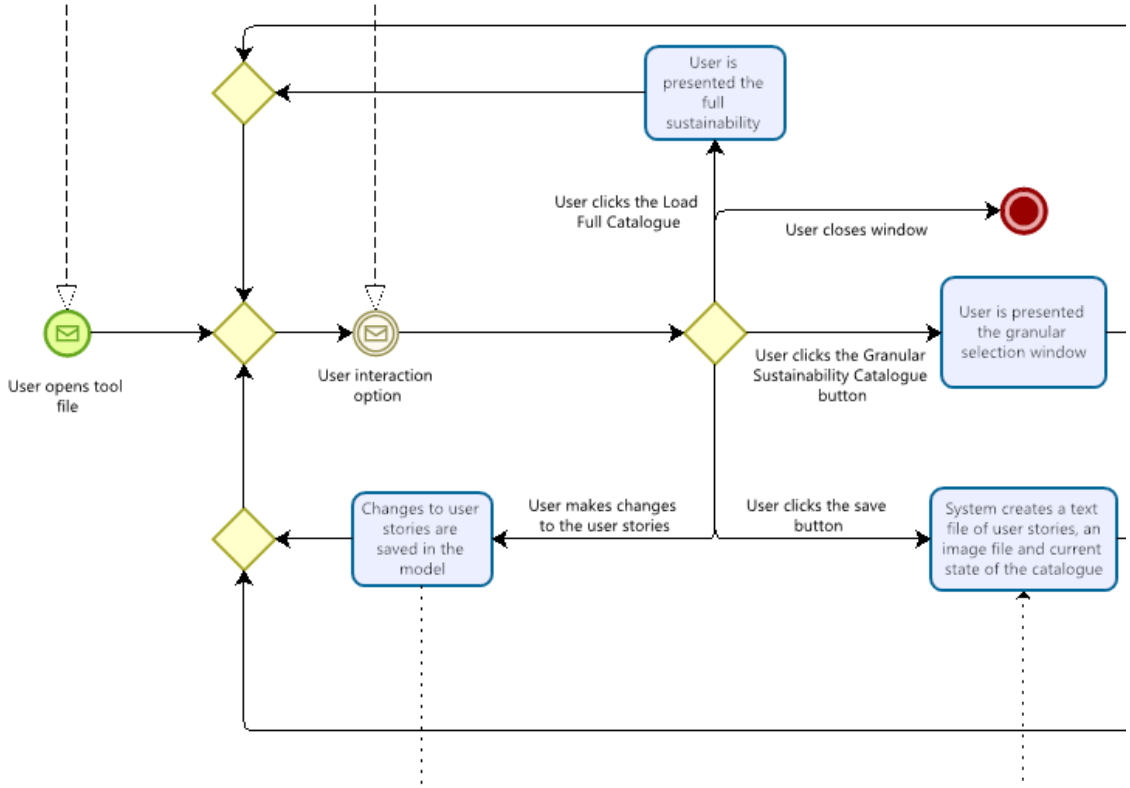

Figure 5.8: Close up image of the choices available to the user

The process loops on this first step until the user chooses to start the sustainability qualities selection or closes the window. The process also comes back to this step once the sustainability qualities selection is finished and sustainability user stories are derived.

Main Process. This process is where the sustainability user stories are ultimately derived from and is composed of two steps, the selection of user stories and the assignment of user roles, both of which are described below. This steps are represented in the diagram as can be seen the a close up in figure 5.9.

Sustainability Quality Selection Step. This step is where the user selects the sustainability qualities from which s/he wants sustainability user stories. As such this is a crucial

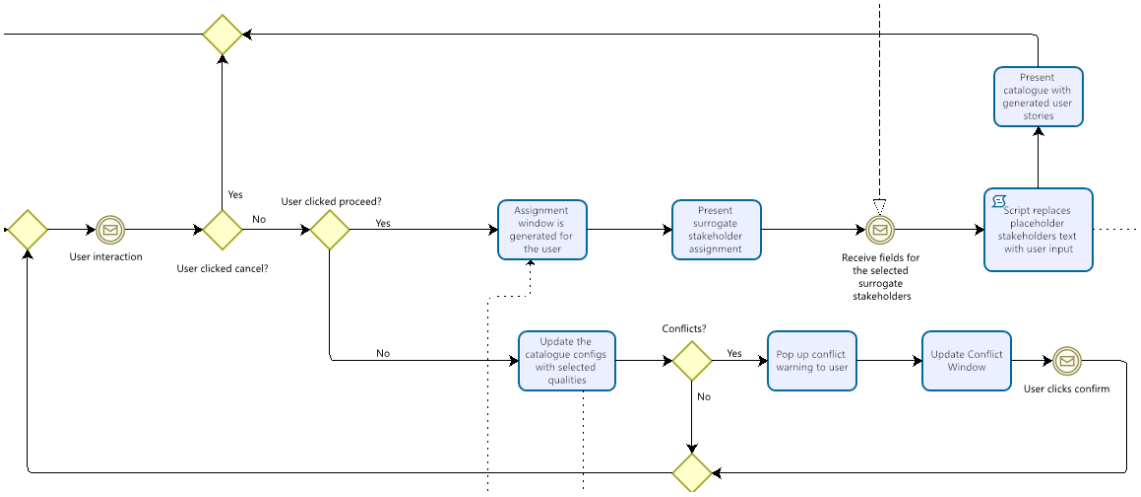

Figure 5.9: Close up image of the main process

step and is mandatory for the user to complete. This step starts once the user clicks on the "Granular Sustainability Catalogue"button. Afterwards, the user is presented with a selection screen, as described in paragraph 5.2.2, here the user selects the sustainability qualities s/he wants.

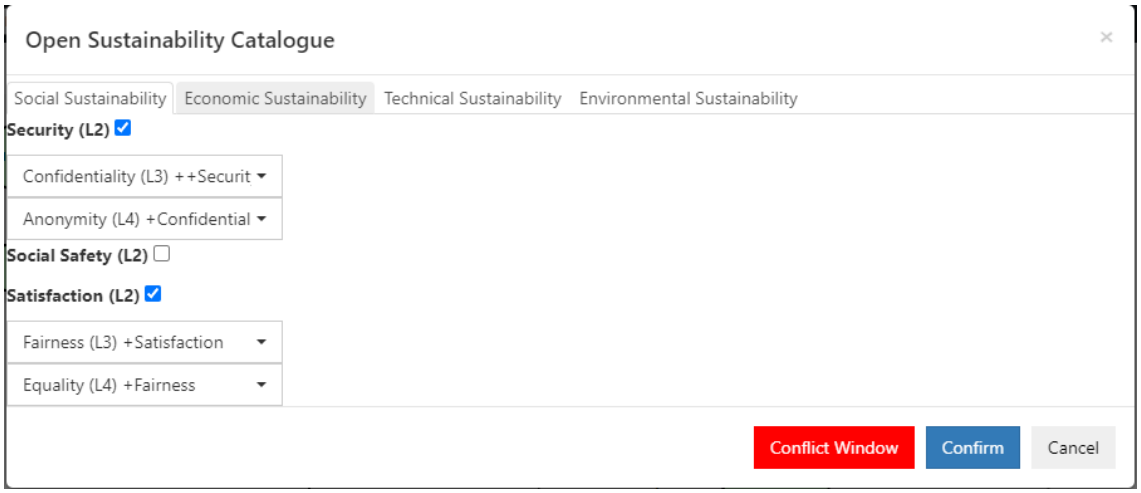

Figure 5.10: Example of Sustainability Quality selection window in the Social Dimension tab with randomly selected qualities

Each time the user selects a sustainability quality the systems checks whether a conflict has been created between the recently selected quality and a previously selected one. If a conflict occurs the system shows the user a warning via a pop-up window that the user can only close by clicking the 'Ok' button. **??**

Furthermore, this conflict is then added to a list of existing conflicts that the user can consult at any time by clicking the 'Conflict Window' button on the sustainability quality selection screen. **??**

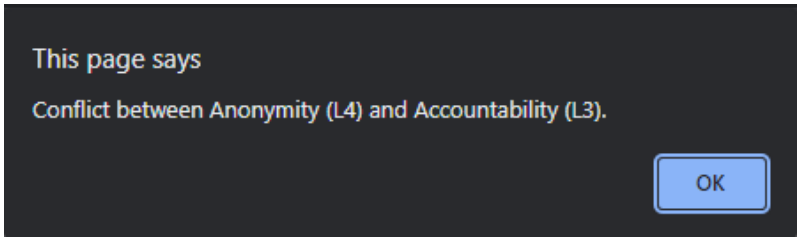

Figure 5.11: Example of a conflict pop-up

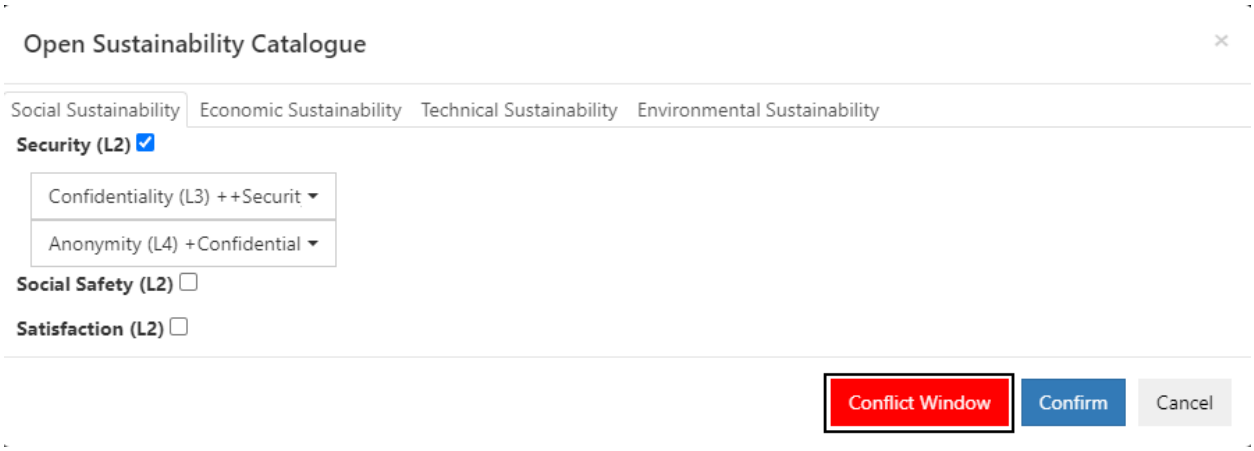

Figure 5.12: 'Conflict Window' button placement

Figure 5.13: Example of the sustainability window with a conflict

If no conflict is detected upon the selection of a sustainability quality the user can just keep on selecting other qualities until s/he is satisfied or a conflict happens. If the user chooses to cancel this selection and close the selection window the currently selected qualities are not lost and the progress will remain once the user opts to re-open the selection window, unless the tool is restarted or the page refreshed.

User Role Replacement Step. Once the user is done with the selection of sustainability qualities and clicks the 'Proceed' button the user is presented with what we call an 'Assignment Window', as shown in figure 5.6, where the system displays a window with input text boxes to replace the user roles of the sustainability user stories selected.

Instead of having a text box for each user story, we group the user stories by main quality, like 'Security' and 'Social Safety', and as such replace the user roles in all of the user stories belonging to the same main quality with only one text boxes. This makes for a quick way to get some level of customization to the user's sustainability user stories. Just like the progress with the sustainability quality selection window if the user types something in one of the boxes and cancels the operation or has already completed this process once the text boxes will retain what was typed by the user, until s/he refreshes the page or restarts the tool.

Individual User Story Editing Sub-Process This sub-process is started once the user clicks the 'Save Catalogue' button and is only really available in case the user has loaded any of the catalogue elements, through selection or by loading the full catalogue, since its aim is to individually edit information pertaining to any of the catalogue elements. This process allows the user to edit all of the contents of the user story and not just the user role, as was available during the main process, allowing for total freedom and customization. That said, we can not take responsibility nor vouch for the integrity and correctness of the sustainability user stories resulting from this sub-process as their intent and structure can be lost due to erroneous changes.

Save Sub-Process Once the user clicks the save button this sub-process is started. It is simple and consists in a single extra step, shown in figure , in which the user selects where to save the files that will be downloaded. Once that information is input into the system the files are downloaded onto the users machine.

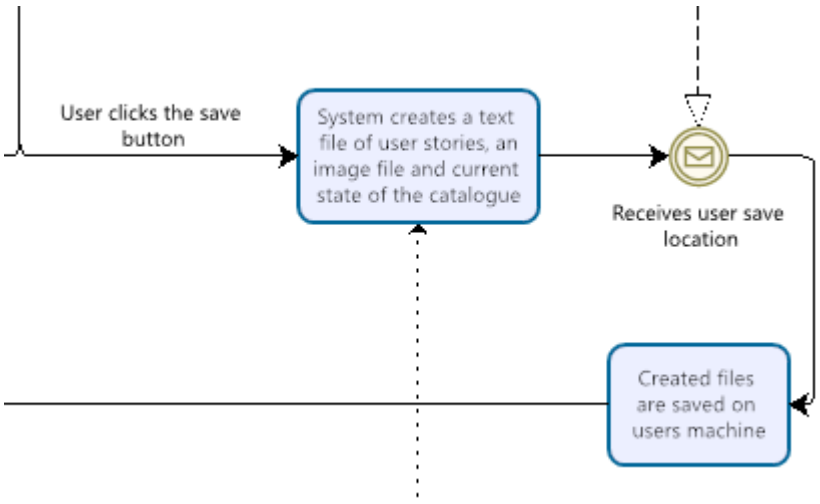

Figure 5.14: Close up image of the main process

5.4 Summary

This chapter presented our framework, where we started by discussing the conceptual model which defines its overall structure. This is followed by the description of the tool and the process for using it.

6

Applying the Process To a Case Study

This case study is a validation process that aims to assert whether or not our tool works in a real-life environment. To do that we will be applying it to a real-life (sustainability) project called U-Bike. We will go through the developed process for our tool and generate sustainability user stories for U-Bike.

6.1 Study of a case: The U-Bike Project

To validate the tool support and get a sense of its real usefulness, we will apply it to a real-world example. This example is based on the U-Bike project https://www.ipbeja.pt/sas/gaad/U-Bike/Paginas/default.aspx [7], an initiative promoted by the Portuguese Institute of Mobility and Transportation (IMT) in collaboration with universities spread across the country. This project promotes healthy lifestyles and ecological mobility in universities with the use of bicycles at no cost to the users.

These users can be any member of a university, student, professor or any other faculty member. These bicycles are leased to the users by their university for any time necessary, ranging from a full school year to only a couple of weeks in the case of an Erasmus student or any other special circumstance, and the only thing the user pays is a caution that is refunded upon returning the bike. This caution can be paid in full or in instalments if agreed by both parties to not discourage the use of the program.

The university that provides the bicycle also makes sure the user has insurance for any accidents that might happen with the bike during its lease time. The bikes provided by the institutions can be traditional bikes or electric bikes, it depends on what is available at that university and what the user prefers. All of these bikes come with the necessary equipment to properly maintain them and have to abide by a set minimum of features and specifications that have been predetermined by the IMT.

This case study aims to present an overview of what is needed for this system to function in a sustainable manner. To achieve this, we will select sustainability qualities using the catalogue and then derive the sustainability user stories for those qualities. The sustainability qualities will be related to a surrogate stakeholder, in this case, the

surrogate stakeholders are: system, user, bike, university and environment.

6.2 Case Study Applied

This section will be divided into two parts: the selection of the sustainability qualities and the derivation of the sustainability user stories. For the selection process, we'll make use of the loading of the full sustainability catalogue, so we can get an overview of all the available sustainability qualities, and the 'Sustainability Quality Selection Step' mentioned in 5.3. The selected qualities will be split into their respective main categories, for example 'Security', and will be presented alongside their rationale for selection. This rationale is based on what, for us, are the needs the U-Bike project would need for its software to be sustainable. The user story derivation focuses on the substitution of the user roles of the selected sustainability user stories, using the process described in 5.3, and the individual editing of user stories if necessary, using the process described in 5.3.

6.2.1 Selection of Sustainability Qualities

Security. So that the U-Bike system can be secure, it must be able to securely save the users' data (personal information, bank account details) and only be able to be accessed by authorized people through secure communication channels (refers to *confidentiality* and *authenticity*). Another crucial factor is that the system cannot charge the wrong user, or value, mistake the owner of a bike or assign it to the wrong user (refers to *integrity*). Even though the system must maintain the users' data confidential it must also make the user accountable by taking proper action when payments are missed, whether it be by contacting/alerting the responsible entities or by automatically issuing the return or repossession of the bicycle (refers to *accountability*). The summary of all selected 'Security' sustainability qualities are represented in figure 6.1.

Figure 6.1: Screenshot of selected Security Qualities

Social Safety. The system must always safeguard the user who rents a bike (refers to *Freedom from Risk*) by making sure s/he has an active insurance that can be activated in

case of an accident with the bike and ensuring each owner is provided with a maintenance kit for the bike making to assure it's kept in working condition because if the bike is well maintained it is likely to be safer. Still in terms of "Social Safety", there is *legislation*, as in the system must be able to provide the users at all times with non-ambiguous documentation of the rules and terms of use for U-Bike. This safeguards both the user and the university's rights and duties. The summary of all selected 'Social Safety' sustainability qualities are represented in figure 6.2.

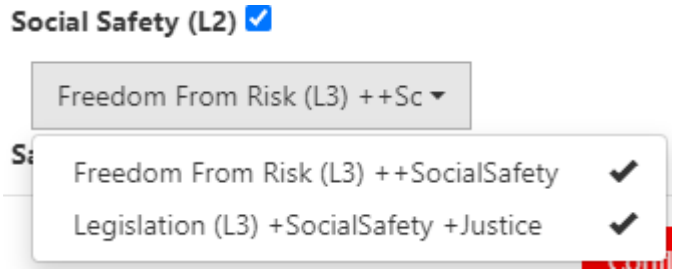

Figure 6.2: Screenshot of selected Social Safety Qualities

Satisfaction. The user must be contempt with the available features and the functioning of the system to keep coming back. The system should have metrics available to measure the satisfaction of the users and/or should periodically inquire users on their level of satisfaction. The users' satisfaction can also be increased by implementing new features further down the road or by improving existing ones (refers to *usefulness*). One important feature for the system to be sustainable and also keep users coming back is *trust*. This is achieved if the system is clear on the way it handles the users' data and keeps a clear and honest communication with its users (refers to *honesty* and *transparency*). The summary of all selected level 3 (L3) and level 4 (L4) 'Satisfaction' sustainability qualities are represented in figure 6.3 and figure 6.4 respectively.

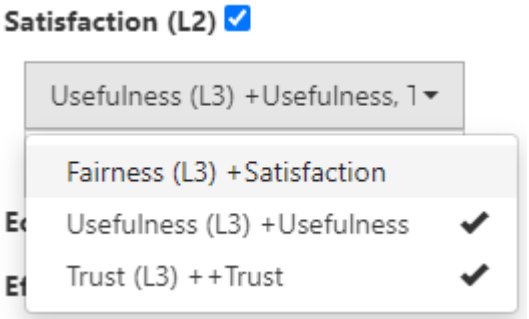

Figure 6.3: Screenshot of selected L3 Satisfaction Qualities

Efficiency. Since the U-Bike project is not specific to one single university and can be adopted by pretty much any higher education institution it is important for it to have

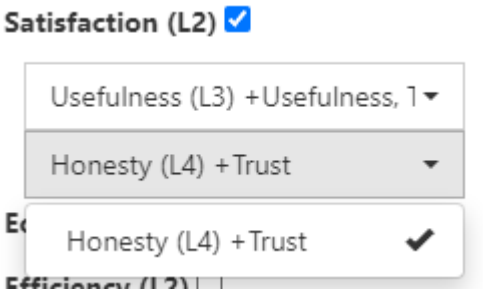

Figure 6.4: Screenshot of selected L4 Satisfaction Qualities

shared resources that can be utilized by each of the different institutions, this being databases and maybe even servers (refers to *resource utilization*). Since more than one institution will implement U-Bike together with the fact that users can request bikes to an institution other than their own, in the case they live nearby one during school break for example, it is good to have *interoperability* between the several systems. That way they can communicate with each other and validate the identity of a user and allow a bike to be leased. The summary of all selected 'Efficiency' sustainability qualities are represented in figure 6.5.

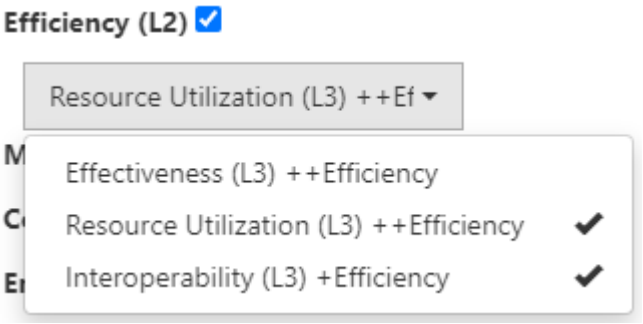

Figure 6.5: Screenshot of selected Efficiency Qualities

Maintainability. For the system to be successful in the long run it needs to be easily maintainable, for example, if an error occurs it should not be too complicated nor take very long to fix it (refers to *maintainability*). The system also has to be quick and easy to change according to new requirements or hardware specifications that may arise during its life cycle (refers to *modifiability* and *adaptability*). The summary of all selected level 3 (L3) and level 4 (L4) 'Maintainability' sustainability qualities are represented in figure 6.6 and figure 6.7 respectively.

Environmental Safety. Since U-Bike is a sustainability-oriented project it is expected that the system worries about *Environmental Safety*. As such the system needs to make sure it uses renewable energy sources as much as possible, uses as few physical mediums as possible (paper and hard drive space), this can be achieved by saving just the necessary

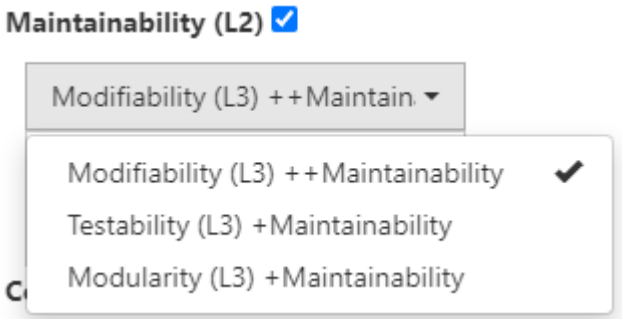

Figure 6.6: Screenshot of selected L3 Maintainability Qualities

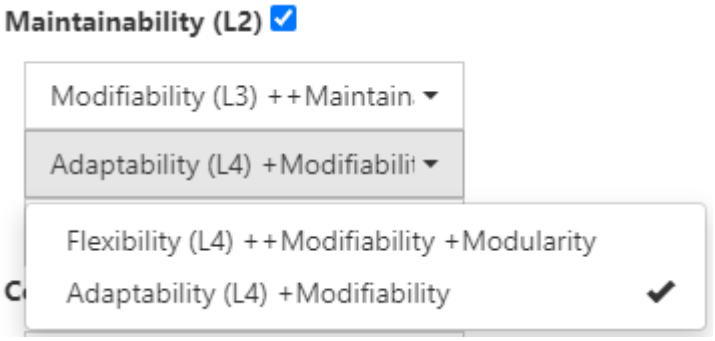

Figure 6.7: Screenshot of selected L4 Maintainability Qualities

items and having compression algorithms for 'cold' storage items and sending official documents, such as invoices, through email rather than traditional mail. The summary of all selected 'Environmental Safety' sustainability qualities are represented in figure 6.8.

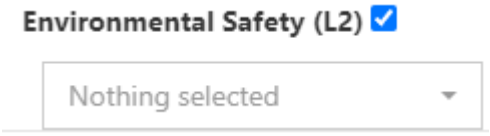

Figure 6.8: Screenshot of selected Environmental Safety Qualities

6.2.2 Derivation of Sustainability User Stories

After selecting the sustainability qualities, selecting them in the 'Quality Selection menu' and choosing our User Roles we now have our derived General Purpose Sustainability User Stories. For the user stories to be traceable to their respective requirements, they will be split by categories in the same way the requirements were previously.

The resulting user stories presented below have been assigned a user role based on the main quality they belong to. The user roles used are shown in figure 6.9:

Security. In the security category, belonging to the Social tab, we selected *Confidentiality* and *Accountability*. These sustainability qualities are both from level 3 on the catalogue

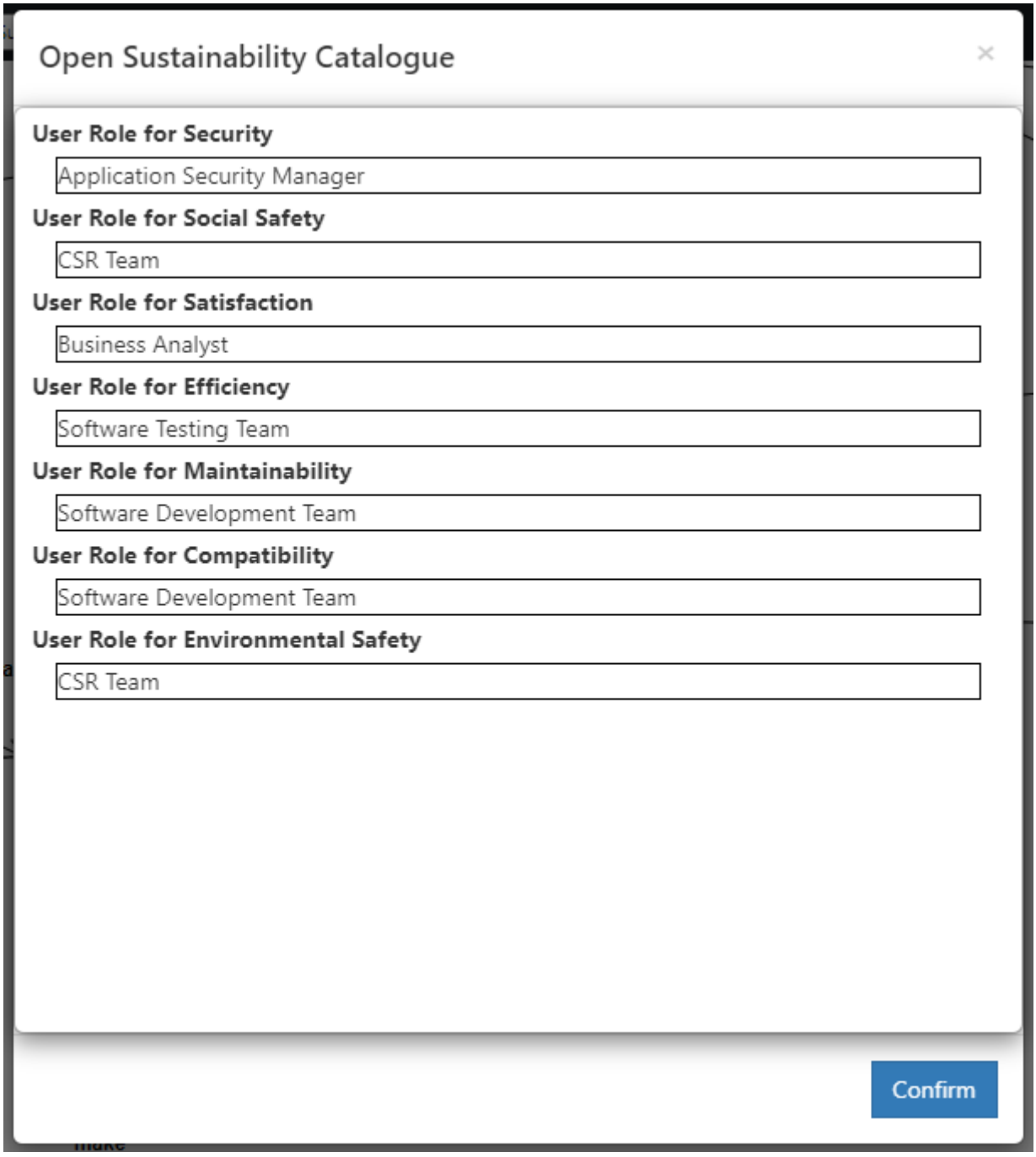

Figure 6.9: User roles used for user story derivation

and were assigned the user role of *Application Security Manager*. The resulting user stories are:

- **Confidentiality:** "As a Application Security Manager, I want the system to keep sensitive data protected from unauthorized sources so that it can achieve Confidentiality."
- **Accountability:** "As a Application Security Manager, I want the system to be able to uniquely identify a user's actions and requests to her/him only so that it can achieve Accountability."

As the user stories are general-purpose there were a couple of grammatical errors due to the user role change and manual editing was performed. The changes are highlighted using bold. The results were:

- **Confidentiality:** "As **an** Application Security Manager, I want the system to keep sensitive data protected from unauthorized sources so that it can achieve Confidentiality."
- **Accountability:** "As **an** Application Security Manager, I want the system to be able to uniquely identify a user's actions and requests to her/him only so that it can achieve Accountability."

Social Safety. Still in the Social tab we have selected *Freedom From Risk* and *Legislation* from the *Social Safety* category. Unlike the previous category, this one came error-free and as such, no changes were made. The resulting sustainability user stories were the following:

- **Freedom From Risk:** "As a CSR Team, I want the system to be concerned about my well-being, by mitigating or eliminating potential risks to me so that I can achieve Freedom from Risk."
- **Legislation:** "As a CSR Team, I want the company to have system specific policies in place, like terms and conditions and cookie policies, and disclosed to the users in order to achieve legality."

Satisfaction. In the *Satisfaction* category the paradigm changes a bit, i.e., we have more sustainability qualities and user stories than just the ones we selected. We selected three qualities (*Usefulness*, *Trust* and *Honesty*). But we had five elements, the two extra elements are a sustainability quality, *Transparency*, and a task, *Accomplishment of Proposed Functionalities*. These qualities got selected by the system for one reason, a 'make' link. *Accomplishment of Proposed Functionalities* and *Transparency* have 'make' links to *Usefulness* and *Honesty* respectively, this makes them crucial pieces in achieving the sustainability qualities they are connected to. The reason these elements, unlike others with

'make' links, got added without the user's input is that they are at the end of their chain, lowest level on their respective branches, and they have only one connection to another element and with them being crucial in the completion of user-selected qualities it only makes sense to include them.

With this said the resulting user stories are:

- **Usefulness:** "As a Business Analyst, I want the system to satisfy the user's goals so that it can achieve Usefulness."
- **Accomplishment of Proposed Functionalities:** "As a Business Analyst, I want the system to correctly perform its intended functions so that it can achieve the Accomplishment of Proposed Functionalities."
- **Trust:** "As a Business Analyst, I want the system to have clear and truthful disclosure of information so that it can achieve Trust."
- **Honesty:** "As a Business Analyst, I want the system to not be part of nor be able to be used for shady or nefarious purposes and be transparent so that it can achieve Honesty."
- **Transparency:** "As a Business Analyst, I want the system to publicly disclose its information, such as treatment of data and data collection processes, so that it can achieve Transparency."

Efficiency. The *Efficiency* category, with the sustainability qualities we selected, has an odd disposition due to the fact that the elements of this catalogue are pre-positioned and even though we selected efficiency as part of the environmental sustainability it still appears closer to economic sustainability. Just like what happens in *Satisfaction*, there are extra elements selected in the catalogue once the process is finished. In this case the extra elements are three tasks, namely: *Standard Implementation*, *Partnership* and *System Integration*. *Standard Implementation* is also related to *Maintainability* but we will not repeat it in that category and will present it here. The user stories are the following:

- **Efficiency:** "As a Software Testing Team, I want the system to utilize only the necessary/needed resources to complete its operations/tasks so that it can achieve Efficiency."
- **Resource Utilization:** "As a Software Testing Team, I want the system to use only the necessary amounts and types of resources that allow it to meet the functioning requirements so that it can achieve Resource Utilization."
- **Interoperability:** "As a Software Development team, I want the system to be able to communicate with other systems, devices or applications in a coordinated way without special effort from the backend so that it can achieve Interoperability."

- **Standard Implementation:** "As a Software Development Team, I want the system to be developed according to a predetermined set of standards and rules so that it can achieve Standard Implementation."
- **Partnership:** "As a Software Development Team, I want the system to have or be open to collaborations with other systems that further our interests or goals so that we can achieve Partnership."
- **System Integration:** "As a Software Development Team, I want all of the sub-systems to be integrated and work together in a seamless way, communication and data exchange, so that it can achieve System Integration."

Maintainability. Just like in the previous categories *Maintainability* there are two extra elements, both of them being tasks: *System Integration*, described previously, and *Open Source System*. The resulting user stories are:

- **Modifiability:** "As a Software Development Team, I want the system to be cost and time efficient to modify so that it can achieve Modifiability."
- **Adaptability:** "As a Software Development Team, I want the system to be easily adaptable to operational and usage environments, like new software or hardware, so that it can achieve Adaptability."
- **Open Source System:** "As a Software Development Team, I want the software to be publicly available under a copyright license that grants users the rights to study, change and distribute the software so that it can achieve Open-Source."

Environmental Safety. Unlike the other categories, this is the only one where the derived user stories were not directly chosen by the user. This happens because there is only one user story and it belongs to a task that is directly related to the *Environmental Safety* quality through a 'make' link and because of that it is not optional for the user to chose if s/he chooses *Environmental Safety*. The resulting user story is as follows:

- **Freedom From Hazardous Risks:** "As a CSR Team, I want the system to eliminate or mitigate a potential risk to the environment so that it can achieve Freedom from Hazardous Risks."

6.3 Summary

This chapter described the application of our approach to a real case study, the U-Bike project, where we followed the process described in the previous chapter. Next we present a qualitative evaluation of the approach through an online survey.

7

Evaluation

This survey aims at assessing how specialists in software development evaluate the tool support for our developed solution. This evaluation was done based on videos and explanations provided by us and focused on the users' perceptions of the use of the tool since the actual tool wasn't provided to the participants. The current chapter discusses results gathered from companies located in the Lisbon area. This survey was sent to the same participants that answered the first survey about the state of sustainability in IT (52) and some additional ones (4) that come from academia that are knowledgeable on requirements and sustainability. This survey had only 15 participants that answered The survey [1] was also made available through Google Forms and was sent to the participants via email. The following sections introduce the participants, the layout of the survey and the reasoning for each of the questions. More specifically, we start with a biographical summary of the participants (e.g. years of experience and development methods used in their work), followed by a presentation of the questions and a discussion of the results for each question.

7.1 Participants

This survey was answered by 14 professionals in the Portuguese IT industry and 1 researcher. The participants work in different companies and work different roles in software development inside those companies. Table 7.1 summarizes the participants information. From the total number of 15 respondents, 33.3% are project leaders/managers (5 out of 15), 13.3% are requirements engineers (2 out of 15), 13.3% are professional developers (2 out of 15), and 13.3% are software architects (2 out of 15). The remaining 4 roles present in our population are UX/UI, business analyst, product manager and researcher. Each of these roles has a presence of 6.7% (or 1) in our surveyed population.

The majority of our surveyed group, 53.3% (or 8 out of 15), have between 1 and 5 years of experience with agile software development, meaning we again have a fairly new and somewhat inexperienced group of professionals. On a personal note, this is not

[1] https://forms.gle/DAbdWZgZABTH28Zx9

Table 7.1: Respondents individual information.

	Preferred Agile Methods	Role in Org	Experience with Agile (Years)
P1	Kanban, Lean Development, Scaled Agile Framework (SAFe)	Architect	>10
P2	Kanban, Scrum	Project Lead / Project Manager	>10
P3	Scrum	Developer	1 - 5
P4	Scrum	Project Lead / Project Manager	1 - 5
P5	Kanban, Scrum	Project Lead / Project Manager	1 - 5
P6	Scrum	UX/UI	1 - 5
P7	Scrum	Product Manager	1 - 5
P8	Scrum	Researcher	1 - 5
P9	Kanban, Scrum	Requirements Engineer	6 - 10
P10	Kanban, Scrum	Project Lead / Project Manager	1 - 5
P11	Kanban, Scrum	Business Analyst	6 - 10
P12	Extreme Programming (XP), Scaled Agile Framework (SAFe)	Requirements Engineer	>10
P13	Extreme Programming (XP), Scaled Agile Framework (SAFe)	Project Lead / Project Manager	>10
P14	Extreme Programming (XP), Scaled Agile Framework (SAFe)	Architect	1 - 5
P15	Kanban, Scrum	Developer	<1

necessarily a negative point, as younger people can be more open-minded to new things and are usually more in touch with environmental issues than any other generation before. On the other hand, 40% of our participants (6 in 15) have more than 5 years of experience, with two thirds of them having more than 10 years and a third having between 5 and 10 years of experience. Another interesting statistic is concerned with the adoption of agile methodologies for software development, where 100% of our participants make use of some type of agile approach and 86.7% (or 13 out of 15) work with Scrum, making it the most common approach amongst our participants.

7.2 Survey Questions and Results

The survey is structured with a total of 12 questions, covering the different aspects of the tool support we developed for our solution, like usability and usefulness. The questions are a mix of multiple choice and open ended questions, with 10 multiple choice questions (namely questions 1 through 10) and 2 open ended questions (11 and 12). Along with the questions the survey provided two short videos and image support explaining the sustainability catalogue used as a base for our solution, the new features developed as tool support and the process to use the tool. We will split the questions by their overall purposes, such as ease of use or usefulness, to keep the document organized and repetition to a minimum.

7.2.1 Ease of Use

The first group of questions target the ease of use of our tool support and aim to get an overview of how easy to use our tool was perceived to be. The questions are split

between the two features we meant to obtain feedback on (selection feature and user role replacement feature) and the overall rating of the process to derive a user story. All of the three questions had 5 values to choose from that range from 1 to 5, with 5 being the highest, meaning the feature was perceived to be very easy to use, and 1 being the lowest, meaning the feature was perceived to be very difficult to use.

Question 1. How do you rate the ease of use of the selection feature (steps 1 to 4)? Figure 7.1 synthesizes the received answers. The answers received were all between the values 3 and 5, this is good as it shows no one perceived the feature to be difficult to use. The value with the most incidence was 4 with 46.7% (7 out of 15). The values 3 and 5 had the same incidence with 26.7% (4 out of 15). This is good since the UI was something that was not our focus for this dissertation yet we managed to achieve something that appears to be easy to use.

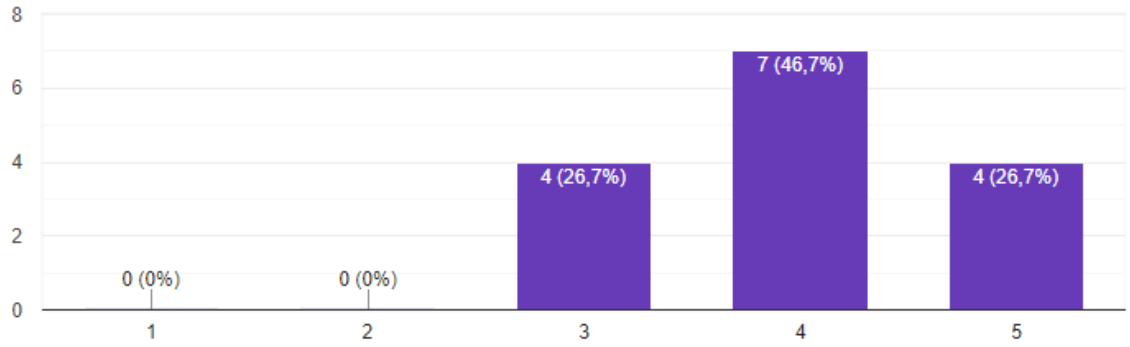

Figure 7.1: Summary of the Answers for Question 1

Question 2. How do you rate the ease of use of the user role replacement feature (step 5)? Figure 7.2 summarizes the received answers. The answered values for this question are, like that previous, also between 3 and 5. Unlike the previous question though the most answered value was 5 with 40% (6 out of 15), followed by 4 with 33.3% (5 out of 15) and lastly 3 with 26.7% (4 out of 15). Again this is pretty good feedback on something that was not our main focus yet we still achieved a good result.

Question 3. How easy would you rate the process to derive the sustainability user stories? We also aimed to evaluate the overall ease of use of the whole process of deriving sustainability user stories. The answers received are summarized in Figure 7.3. In the figure, we can see that the answered values all range from 3 to 5 and all have an equal pick rate of 33.3% (5 out of 15).

7.2.2 Usefulness

This group of questions target the usefulness of the features of our tool support and the sustainability user stories. There are three questions, two for each of the features we want

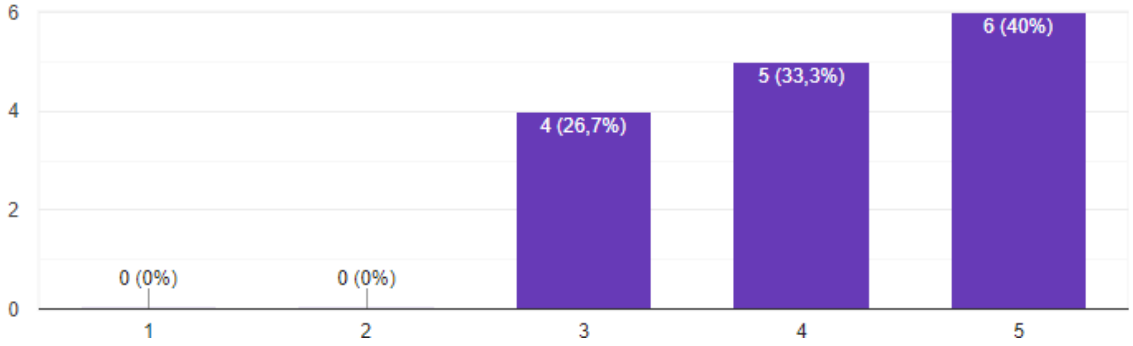

Figure 7.2: Summary of the Answers for Question 2

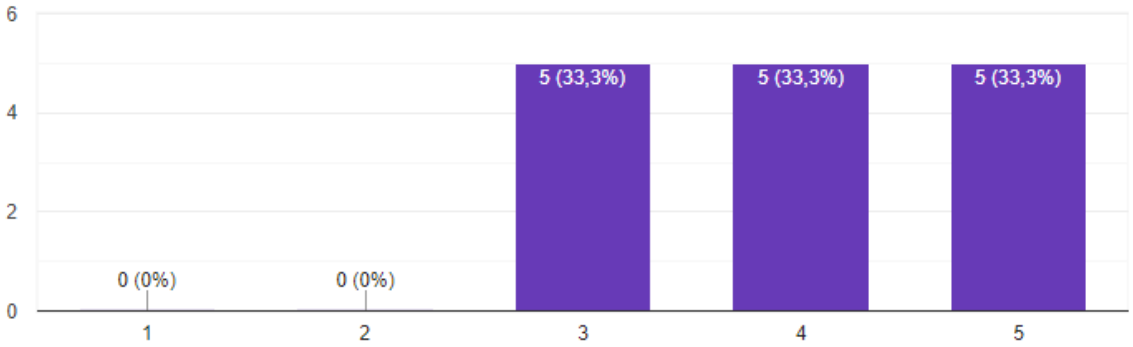

Figure 7.3: Summary of the Answers for Question 3

feedback on (selection feature and user role replacement feature) and one for the user stories. These questions are all multiple choice where the user can choose a numeric value from 1 to 5, with 5 being the highest, meaning the subject in question is very useful, and 1 being the lowest, meaning the subject in question is very useless.

Question 4. How do you rate the usefulness of the selection feature (steps 1 to 4)? This question has a sparser answered value range than the previous ones, ranging from 2 to 5. Even though most feedback is positive with 33.3% (5 out of 15) answering a value of 5 and 20% (3 out of 15) answering with a value of 4. There was 6.7% (or 1) that answered 2 and 40% (6 out of 15) that answered neutrally with a value of 3. Figure 7.4 synthesizes the answers for this question.

Question 5. How do you rate the usefulness of the user role replacement feature (step 5)? This question had the same results as the previous one. We found this odd and individually checked each user's answers to see if it might be a case of people choosing randomly and as such answering the same value again but found that 40% of our participants (6 out of 15) voted differently. The results of this question are shown in figure 7.5.

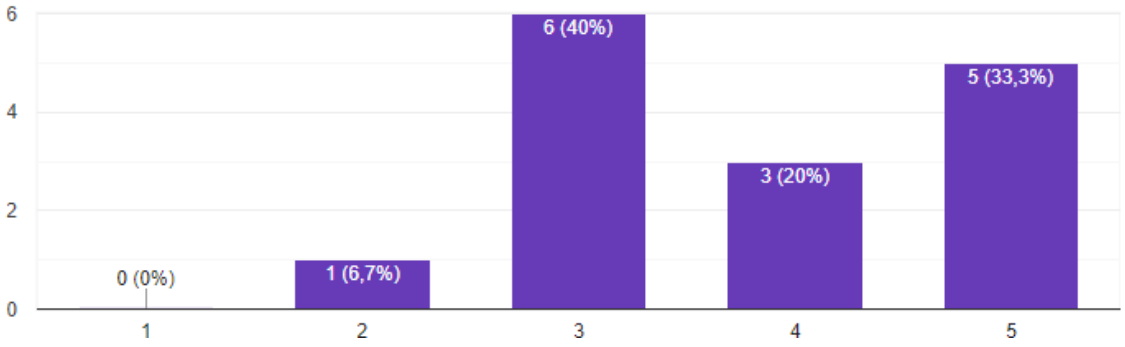

Figure 7.4: Summary of the Answers for Question 4

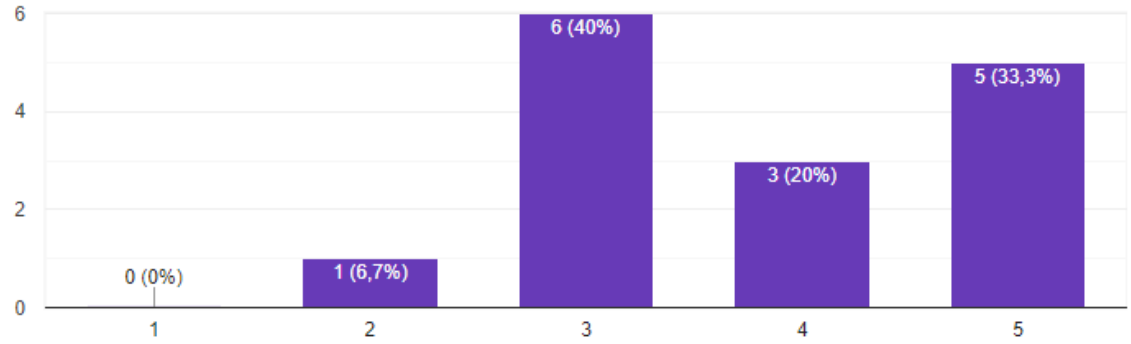

Figure 7.5: Summary of the Answers for Question 5

Question 6. How do you rate the usefulness of the sustainability user stories? This question had one participant selecting the lowest possible value of 1. On the other hand, it had fewer participants selecting the neutral value 3, only 26.7% (4 out of 15), and had more selecting the value 4. This value has the same selection rate as the value 5, which is 33.3% (5 out of 15). So even with a participant answering with the lowest value possible this question still managed to average a slightly better value than the previous. Figure 7.6 synthesizes the received answers.

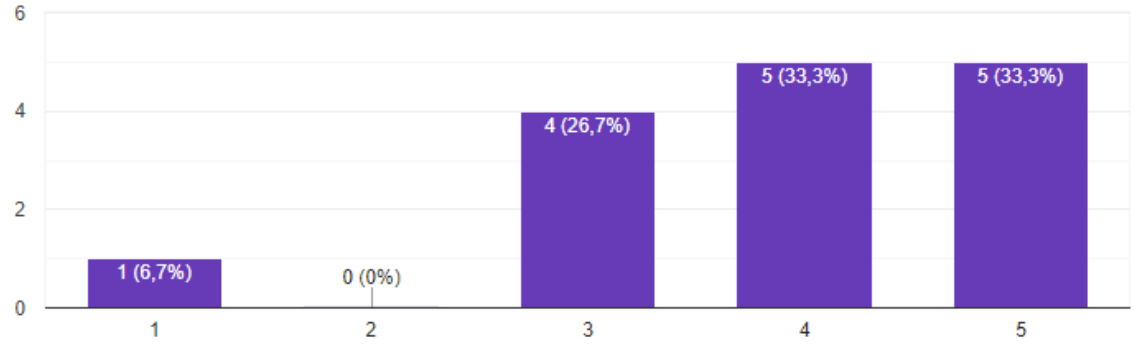

Figure 7.6: Summary of the Answers for Question 6

7.2.3 User Feedback/Opinion

Question 7. How do you rate this tool as a whole? The aim of this question was to assess the users' likeness of the overall tool support we developed. This includes the usefulness and ease of use of all the features of the tool and not just the two we inquired about in the questions above. The answers to this question are summarized in Figure 7.7. This question also had one participant answer with the lowest possible value of 1 but to oppose that we had 53.3% of participants (8 out of 15) selected the value 4. The remaining participants split their votes into the value 3 and value 5, with both options having a selection percentage of 20% (3 out of 15).

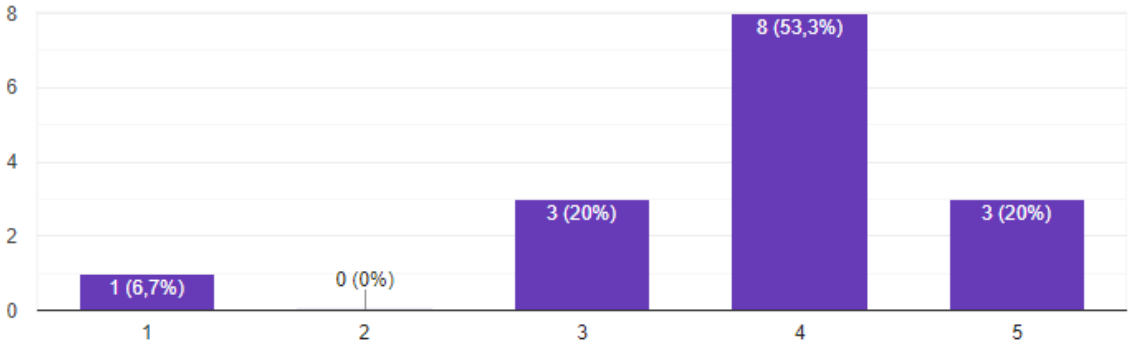

Figure 7.7: Summary of the Answers for Question 7

Question 8. Would you consider using the sustainability user stories as a basis for your future projects? This question aims to assess whether or not our participants thought our derived user stories would be good enough to base their user stories on and allows for more concrete feedback than the previous questions where the participants would use numerical values while complementing their results. This question is also a multiple choice question but instead of a value range the user can choose one of three options: Yes, Perhaps or No. The response was mostly positive, 66.7% of participants (10 out of 15) said they would consider using the derived sustainability user stories as a basis for their projects. Other 26.7% of participants (4 out of 15) said they would perhaps consider using them and only 6.7% (1 out of 15) said they would not consider using our user stories. Figure 7.8 summarizes the participants answer for this question.

Question 9. If anything, what would you change? This question was designed to gather information on the bigger flaws of our sustainability user stories are. The participants had 4 options to choose from plus an empty text field where they could type their own option. This question is not mandatory and as such the participants could choose not to answer. The 4 options are: "The goal of the User Story", "The reason of the User Story", "The user role of the User Story", "All of the above". The figure 7.9 summarizes the 9 answers obtained on this question. Out of these 9 answers, 44.4% (4 out of 9) said they

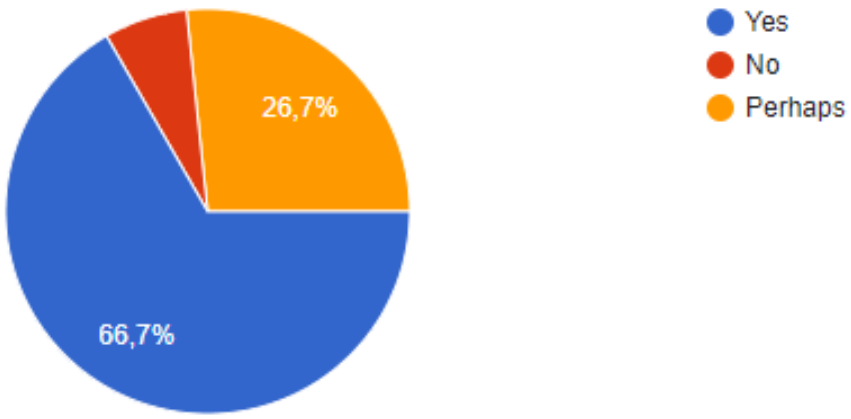

Figure 7.8: Summary of the Answers for Question 8

would change the reason of the user story, 22.2% (2 out of 9) said they would change the user role of the user story and 22.2% (2 out of 9) said they would change all of the above. Lastly, one participant chimed in with her/his answer and said they would change the goal and the reason of the user story.

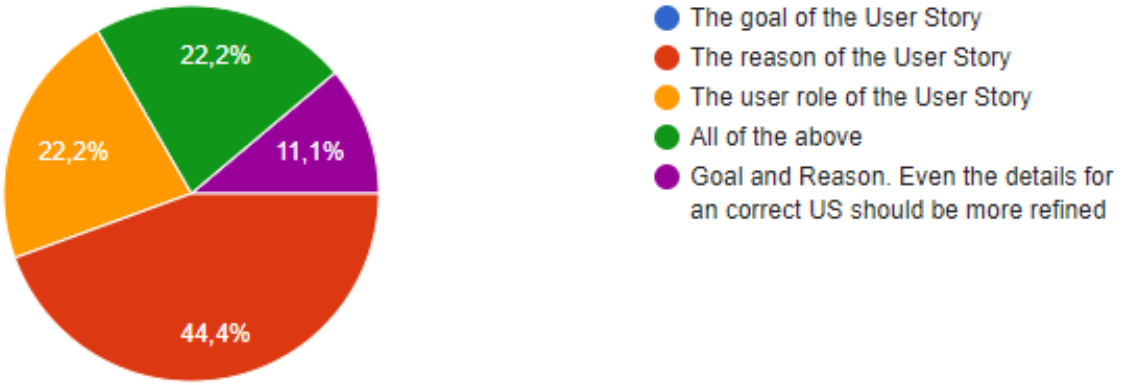

Figure 7.9: Summary of the Answers for Question 9

Question 10. Would you consider using this sustainability catalogue and configuration module? This question aimed to gauge whether our participants would consider using our tool to derive the sustainability user stories. This question has three possible answers to choose from: "Yes", "Perhaps", "No". The response was mostly positive with 53.3% of the participants (8 out of 15) saying 'Yes' and 40% (6 out of 15) saying 'Perhaps'. Only 1 participant said 'No'. The results for this question are summarized in figure 7.10.

Question 11. Which aspects of the sustainability catalogue and configuration module do you consider most relevant or positive? This question is open ended and mandatory with the aim to gather the users' thoughts on what the best features and aspects of the tool were and/or sustainability user stories. Our participants had an array of different answers,

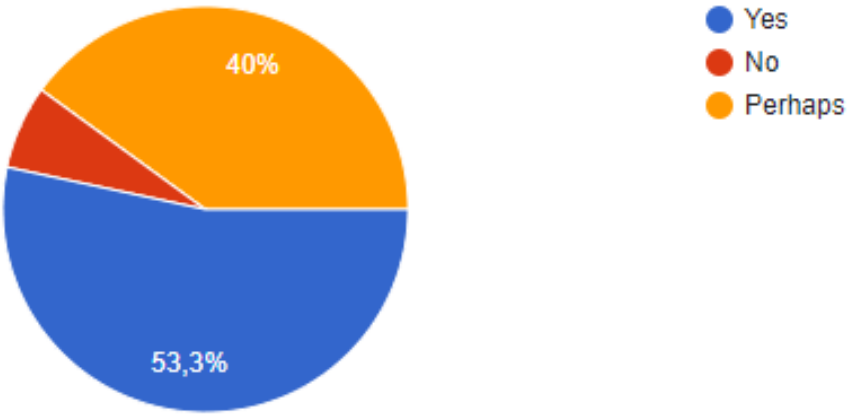

Figure 7.10: Summary of the Answers for Question 10

some said the best part was the "possibility of generating user stories" three others said it was the "transparency and organization" the tool provides. One participant also said it was the "ease of use and the usefulness of the selection feature" while others said the best aspect was the sustainability part of the tool and catalogue. With our participants' feedback, we understood that sustainability was well received and that a tool with a catalogue 'format', or another highly organizable one, is desirable and a colour coding for easy identification of features and main qualities.

Question 12. Which aspects of the sustainability catalogue and configuration module would you change or modify? And why? This last question was also open ended and mandatory, with the goal of knowing what we did not do so well and what we can improve on. By allowing the participants to freely type their opinions we knew understood that some UI changes are needed, like the "position of the buttons", more "flexibility" needs to be introduced and one even said some type of algorithm to recommend elements from the catalogue to the user "based on its usage". But the most reported thing our participants want is the continued improvement and support of the tool and the user stories by for example evolving the "core functionality to refine more relevant details in User Stories" so that they can become more "complete and, therefore useful".

7.3 Discussion

This section discusses the results of a survey performed with a subset of the Portuguese industry and researchers with the goal of gathering feedback on our developed solution, both the sustainability user stories and the tool support. We were particularly interested in knowing if they considered our user stories good enough to be used in their projects, even if only partially.

This survey was answered by 15 people, 14 IT professionals and 1 researcher, from several companies working in several application domains. The analysis revealed that the

majority of the participants provided positive feedback regarding our work. Despite that, some participants had some thoughts and harsh critics on how it at all could be improved to be made useful in their professional life, even awarding some of the questions with the lowest possible score.

On the other hand, 66.7% said that they would consider using our user stories as a basis for their future projects and 53.3% said that they would consider using the catalogue and configuration module.

Finally, all of the questions had more positive feedback than negative with no question response average dropping below 3.8, out of 5. Identifying the shortcomings of our solution is important to know so as to improve it as much as possible in the long run and achieve its full usefulness. The survey was able to achieve this, despite the low amount of participants, and we discovered that one of the most important things for the participants is that the solution (tool and user stories) is continuously improved.

7.4 Threats to Validity

There are always threats to the validity of any survey. In this section, we'll discuss some threats and respective mitigations.

Internal validity. A threat to the survey is always that we could be asking the wrong questions or at least asking them in a wrong way, by using ambiguous language or confusing writing. To mitigate this we grouped the questions by topic and made certain we used simple and clear phrasing by reviewing the questions multiple times among the authors.

External validity. One threat is related to the participants, whereas we didn't obtain any information that might identify the company of each respondent in order to not stress the participants about being possibly identifiable and as such feeling more at ease, to be honest with the answers they provide. Although in doing this we can't be sure if we're having a full representation of the IT industry or just one company's perspective on the matter.

Conclusion validity. The data analysis itself may be a possible threat because wrong data analysis can provide wrong or misleading results and as such sway the authors to wrong conclusions. Additionally, this survey had a reduced number of participants, 15, and as such the conclusions might be compromised. To mitigate this possibility, we analyzed the data for each question thoroughly and individually so as to not make any mistakes or confusion with other questions' data.

7.5 Summary

This chapter presented a qualitative evaluation of our approach where a survey was answered by 15 participants. The survey was composed by 12 questions that covered different aspects such as ease of use and usefulness of the approach. Also, we asked for

the participants to give some feedback about the approach and the tool. The results were positive, reinforcing the importance of pursuing this research topic.

8

Conclusions

In this chapter, we'll present our final conclusions along with a summary of the work done for this MSc dissertation and some ideas and suggestions for future work and improvements that can be done to our work. What were our main obstacles and what we learnt along the way, like the constant increase in society's need for sustainability. Our development of the sustainability user stories and improvement of the sustainability catalogue aims to help with such a need. Even though we created sustainability user stories their continuous improvement is paramount so they can be as useful and complete as possible.

8.1 Summary of Our Work

Our work for this Master's thesis can be divided into 4 different stages: idealization, state of the art, development, and evaluation. Each of these stages had its own challenges we needed to overcome to achieve our goals. In this section, we'll discuss these 4 stages of our workflow and, afterwards, draw our conclusions.

The first stage of our thesis had the objective of defining our goals and getting a concrete idea of what our solution to sustainability in agile software development would be. This said we ended up with the objective of not only creating general-purpose sustainability user stories but also integrating them into a process in which they can be made usable by interested users. This way we could contribute to the achievement of sustainability, more specifically, sustainability in software. Furthermore, our solution needed to be generic enough that it could be useful to as many people and project domains as possible and achieve its full usefulness. However, at this point we didn't know the best format in which to write the user stories, nor if any sustainability user stories had already been published. Adding to that, we also didn't know if there were already any methods/processes to implement sustainability in agile software development.

Given the challenge presented in stage one of 'not knowing' our second stage consisted in deepening our knowledge of the existing methods of applying sustainability in software development and sustainability user stories. To do this we did a state of the art analysis through the use of a systematic mapping review, this way we could be sure we would collect as much pertinent information as possible and not miss any relevant materials. Our main search questions were about any existing approaches to sustainability in agile methodologies and how sustainability is addressed in requirements engineering in agile methodologies. With this study, we understood that there were nearly no scientific documents on the topics we were researching and as such, we needed to conduct further research manually to make certain we didn't miss any relevant information. That research was based on papers that investigated sustainability in software even if it wasn't in the specific agile domain. Now that we had gathered enough information, we needed to start implementing a functional solution.

That was the focus of our third stage, development. Our development stage was itself divided into two main goals: writing the sustainability user stories and developing a process to use the user stories. Both of those goals were discussed during several meetings until a consensus was reached on how to achieve both. For the writing of the sustainability user stories, it was a matter of choosing the appropriate format in which to write them and then a continuous process of building and rebuilding those user stories until they were accepted by all members. For the process to use the user stories we ended up using a pre-existing sustainability catalogue and modified it in order to include both our user stories and improve existing features. The features we improved on were: the selection of sustainability qualities, by refining the selection menu and increasing the granularity at which the user can select qualities; the save feature, by creating a new file, with the derived user stories, that's downloaded along with the already existing ones; and we had to make the catalogue compatible with user stories by modifying the structure of the elements. Furthermore, we implemented a feature to replace semi-automatically the user roles for the derived user stories, so we could facilitate the user's process of getting the user stories project ready.

The fourth and final stage was the evaluation, where we assessed the validity and real-world usefulness of our solution. This was achieved by inquiring IT professionals and academics through a series of questions about our solution, tool and user stories alike, and applying the approach to a real world scenario, the U-Bike project. Since the survey participants had no context or knowledge of the solution we had to find a concise and practical way of doing it. Here we arrived at a solution, of including two small voiced over videos in the survey, these videos covered the background of the tool and the sustainability catalogue used as a base and then our developed work. With this, we managed to obtain 15 answers, although a very small population we managed to gather insights and constructive criticism all the same. The feedback on the solution was in

majority positive with 93.3% of participants saying they would use, or at least consider, using our user stories as a basis for their projects and none of the feedback questions had an average rating of less than 3.8. The case study we performed was based on the U-Bike project, a project that promotes sustainable mobility in higher level education institutions through the use of bicycles, both traditional and electric. Using this project as a base, we performed the processes described in 5.3 and used the catalogue to select the wanted sustainability qualities for a system that would be made for U-Bike. After the sustainability qualities were selected we used the tool support to derive the sustainability user stories presented.

8.2 Contributions

This MSc dissertation contributes not only with an approach to derive general-purpose sustainability user stories from a sustainability catalogue, and tool support, but also with a survey on how agile practitioners, in Portugal, address sustainability in their development of software, since this was an area we noticed there was little research and data on. This survey resulted in a paper that will be published in the proceedings of the 16th International Conference on Research Challenges in Information Science.

Furthermore, we provide an application of the approach to a case study in this case the U-Bike project. Lastly, we also conduct a preliminary evaluation of the approach by surveying IT professionals in Portugal on their thoughts on our approach, gauging their interest and feedback on its usefulness and ease of use.

8.3 Limitations

The approach presented however has some limitations, one of these limitations is the fact that the individual dimension is missing from the catalogue and set of sustainability user stories. Then we have to note that the evaluation was limited due to a low number of participants, however, some did provide some negative feedback and that must be addressed and any legitimate issues solved. Lastly, the tool support itself has some limitations on its usability and design since at this point it is merely a prototype.

8.4 Future work

Even though we finished our work for this MSc dissertation, there are some improvements that can be done to further better the tool and user stories. For instance, the individual dimension has not been implemented in the sustainability catalogue. Sustainability requirements and user stories need to be researched and introduced in order to enrich the catalogue and give it more usefulness. Furthermore, the tool could include more features that were not implemented due to time and/or complexity constraints, such as a search feature for sustainability qualities present in the catalogue, custom sets of predetermined

sustainability qualities for fast starts or an algorithm that can learn the users' behaviours and suggest sustainability qualities based on the type of project and personal use. On the user stories side, definitions of epics, themes and initiatives need to be researched for sustainability.

Bibliography

[1] T. Abdou, P. Kamthan, and N. Shahmir. "User Stories for Agile Business: INVEST, Carefully!" In: 2014 (cit. on p. 2).

[2] D. Albuquerque. *A Sustainability Catalogue for Software Modelling*. Master's Thesis. Campus Caparica, Portugal, 2020 (cit. on pp. 42, 43).

[3] C. Becker et al. "Requirements: The Key to Sustainability". In: *IEEE Software* 33.1 (2016), pp. 56–65. DOI: 10.1109/MS.2015.158 (cit. on pp. 14, 16).

[4] C. Becker et al. "Sustainability Design and Software: The Karlskrona Manifesto". In: *Proceedings of the 37th International Conference on Software Engineering - Volume 2*. ICSE '15. Florence, Italy: IEEE Press, 2015, pp. 467–476 (cit. on pp. 1, 16).

[5] C. Böhringer and P. Jochen. "Measuring the immeasurable — A survey of sustainability indices". In: *Ecological Economics, Elsevier* 63 (2007) (cit. on p. 36).

[6] I. S. Brito et al. "A concern-oriented sustainability approach". In: *2018 12th International Conference on Research Challenges in Information Science (RCIS)*. 2018, pp. 1–12. DOI: 10.1109/RCIS.2018.8406676 (cit. on p. 25).

[7] I. S. Brito, A. Moreira, and J. Araújo. "Handling Nonfunctional Requirements for Smart Cities". In: *Proceedings of the XXIII Iberoamerican Conference on Software Engineering, CIbSE 2020, Curitiba, Paraná, Brazil, November 9-13, 2020*. Ed. by C. P. Ayala et al. Curran Associates, 2020, pp. 334–341 (cit. on p. 53).

[8] C. Calero and M. Piattini. "Puzzling out Software Sustainability". In: *Sustainable Computing: Informatics and Systems* 16 (2017), pp. 117–124. ISSN: 2210-5379. DOI: https://doi.org/10.1016/j.suscom.2017.10.011. URL: https://www.sciencedirect.com/science/article/pii/S2210537916301676 (cit. on p. 16).

[9] M. Carraz et al. "Developer Economics: State of the Developer Nation 16th Edition report". In: (2019). URL: https://www.slashdata.co/free-resources/developer-economics-state-of-the-developer-nation-16th-edition (cit. on pp. 2, 10).

[10] D. Carrizo, O. Dieste, and N. Juristo. "Systematizing requirements elicitation technique selection". In: *Information and Software Technology* 56.6 (2014), pp. 644–669. DOI: https://doi.org/10.1016/j.infsof.2014.01.009 (cit. on pp. 6, 7).

[11] R. P. Casmarrinha. *Avaliação da Qualidade de User Stories*. Master's Thesis. Campus Caparica, Portugal, 2020 (cit. on p. 41).

[12] R. Chitchyan et al. "Sustainability Design in Requirements Engineering: State of Practice". In: *IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)* (2016), pp. 533–542 (cit. on pp. 17, 37).

[13] N. Condori-Fernandez and P. Lago. "Characterizing the contribution of quality requirements to software sustainability". In: *Journal of Systems and Software* 137 (2018), pp. 289–305. ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2017.12.005. URL: https://www.sciencedirect.com/science/article/pii/S0164121217302984 (cit. on p. 2).

[14] F. Dalpiaz, X. Franch, and J. Horkoff. "iStar 2.0 Language Guide". In: *CoRR* abs/1605.07767 (2016). arXiv: 1605.07767. URL: http://arxiv.org/abs/1605.07767 (cit. on pp. 39, 40).

[15] M. Dick et al. "Green software engineering with agile methods". In: *2nd International Workshop on Green and Sustainable Software, GREENS 2013, San Francisco, CA, USA, May 20, 2013*. Ed. by P. Lago et al. IEEE Computer Society, 2013, pp. 78–85. DOI: 10.1109/GREENS.2013.6606425. URL: https://doi.org/10.1109/GREENS.2013.6606425 (cit. on p. 24).

[16] W. C. on Environment and Development. *Our Common Future*. Oxford University Press, 1987. URL: https://EconPapers.repec.org/RePEc:oxp:obooks:9780192820808 (cit. on pp. 1, 14, 40).

[17] D. Firesmith. "Prioritizing Requirements". In: *Journal of Object Technology* 3.8 (2004). URL: http://www.jot.fm/issues/issue_2004_09/column4/column4.pdf (cit. on p. 8).

[18] I. Groher and R. Weinreich. "An Interview Study on Sustainability Concerns in Software Development Projects". In: *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. 2017, pp. 350–358. DOI: 10.1109/SEAA.2017.70 (cit. on p. 37).

[19] *IPCC — Intergovernmental Panel on Climate Change*. URL: https://www.ipcc.ch/ (visited on 2020-12-17) (cit. on p. 15).

[20] B. A. Kitchenham and S. L. Pfleeger. "Personal Opinion Surveys". In: *Guide to Advanced Empirical Software Engineering*. Ed. by F. Shull, J. Singer, and D. I. K. Sjøberg. London: Springer London, 2008, pp. 63–92. ISBN: 978-1-84800-044-5. DOI: 10.1007/978-1-84800-044-5_3. URL: https://doi.org/10.1007/978-1-84800-044-5_3 (cit. on p. 27).

[21] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/master/template.pdf (cit. on p. ii).

[22] Q. Ma. "The effectiveness of requirements prioritization techniques for a medium to large number of requirements: a systematic literature review". In: (2009-01) (cit. on p. 8).

[23] S. Naumann et al. "The GREENSOFT Model: A reference model for green and sustainable software and its engineering". In: *Sustainable Computing: Informatics and Systems* 1.4 (2011), pp. 294–304. ISSN: 2210-5379. DOI: https://doi.org/10.1016/j.suscom.2011.06.004. URL: https://www.sciencedirect.com/science/article/pii/S2210537911000473 (cit. on p. 16).

[24] C. Pacheco, I. García, and M. Reyes. "Requirements elicitation techniques: a systematic literature review based on the maturity of the techniques". In: *IET Software* 12.4 (2018), pp. 365–378. DOI: https://doi.org/10.1049/iet-sen.2017.0144 (cit. on pp. 6, 7).

[25] B. Paech et al. "Towards a Systematic Process for the Elicitation of Sustainability Requirements". English. In: vol. 2541. UID/CEC/04516/2019; 8th International Workshop on Requirements Engineering for Sustainable Systems, RE4SuSy 2019; Conference date: 24-09-2019. CEUR Workshop Proceedings, 2019 (cit. on p. 25).

[26] B. Paech et al. "Towards a Systematic Process for the Elicitation of Sustainability Requirements". In: *Proceedings of the 8th International Workshop on Requirements Engineering for Sustainable Systems co-located with 27th International Conference on Requirements Engineering, Jeju, South Korea, September 24, 2019*. Ed. by B. Penzenstadler, C. C. Venters, and R. Chitchyan. Vol. 2541. CEUR Workshop Proceedings. CEUR-WS.org, 2019. URL: http://ceur-ws.org/Vol-2541/paper1.pdf (cit. on p. 41).

[27] B. Penzenstadler. "Supporting Sustainability Aspects in Software Engineering". In: 2012-01 (cit. on p. 25).

[28] J. Pimentel and J. Castro. "piStar Tool – A Pluggable Online Tool for Goal Modeling". In: *2018 IEEE 26th International Requirements Engineering Conference (RE)*. 2018, pp. 498–499. DOI: 10.1109/RE.2018.00071 (cit. on p. 43).

[29] T. Punter et al. "Conducting On-Line Surveys in Software Engineering". In: *Proceedings of the 2003 International Symposium on Empirical Software Engineering*. ISESE '03. USA: IEEE Computer Society, 2003, p. 80. ISBN: 0769520022 (cit. on p. 27).

[30] N. Rashid and S. U. Khan. "Agile practices for global software development vendors in the development of green and sustainable software". In: *J. Softw. Evol. Process.* 30.10 (2018). DOI: 10.1002/smr.1964. URL: https://doi.org/10.1002/smr.1964 (cit. on p. 24).

[31] N. Rashid and S. U. Khan. "Agile practices for global software development vendors in the development of green and sustainable software". In: *Journal of Software: Evolution and Process* 30.10 (2018). e1964 JSME-17-0140.R3, e1964. DOI: https://doi.org/10.1002/smr.1964. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/smr.1964. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/smr.1964 (cit. on p. 36).

[32] N. Rashid and S. U. Khan. "Developing Green and Sustainable Software using Agile Methods in Global Software Development: Risk Factors for Vendors". In: *ENASE 2016 - Proceedings of the 11th International Conference on Evaluation of Novel Approaches to Software Engineering, Rome, Italy 27-28 April, 2016*. Ed. by L. A. Maciaszek and J. Filipe. SciTePress, 2016, pp. 247–253. DOI: 10.5220/0005913802470253. URL: https://doi.org/10.5220/0005913802470253 (cit. on p. 24).

[33] N. Rashid and S. U. Khan. "Using agile methods for the development of green and sustainable software: Success factors for GSD vendors". In: *J. Softw. Evol. Process.* 30.8 (2018). DOI: 10.1002/smr.1927. URL: https://doi.org/10.1002/smr.1927 (cit. on p. 24).

[34] T. R. D. Saputri and S.-W. Lee. "Addressing sustainability in the requirements engineering process: From elicitation to functional decomposition". In: vol. 32. 8. e2254 smr.2254. 2020, e2254. DOI: https://doi.org/10.1002/smr.2254. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/smr.2254. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/smr.2254 (cit. on p. 26).

[35] *Scrum Guide*. URL: https://www.scrumguides.org/scrum-guide.html (visited on 2021-01-15) (cit. on p. 12).

[36] S. Sharma and S. K. Pandey. "Revisiting Requirements Elicitation Techniques". In: *International Journal of Computer Applications* 75 (2013), pp. 35–39. DOI: https://doi.org/10.5120/13166-0889 (cit. on pp. 5, 6).

[37] I. Sommerville and P. Sawyer. *Requirements Engineering: A Good Practice Guide*. 1st. USA: John Wiley & Sons, Inc., 1997. ISBN: 0471974447 (cit. on p. 5).

[38] *The NaPiRE Project*. URL: http://www.re-survey.org/#/explore (visited on 2021-02-08) (cit. on p. 6).

[39] The Standish Group. *Chaos Report*. 1995 (cit. on p. 5).

[40] *User Stories*. URL: https://www.atlassian.com/agile/project-management/user-stories (visited on 2021-01-15) (cit. on pp. 10, 11).

[41] *Water crises are a top global risk | World Economic Forum*. URL: https://www.weforum.org/agenda/2015/01/why-world-water-crises-are-a-top-global-risk/ (visited on 2020-12-17) (cit. on p. 15).

[42] *Water Scarcity | Threats | WWF*. URL: https://www.worldwildlife.org/threats/water-scarcity (visited on 2020-12-17) (cit. on p. 15).

[43] *What is agile?* URL: https://www.atlassian.com/agile (visited on 2021-01-15) (cit. on p. 10).

[44] *What is agile?* URL: https://www.atlassian.com/agile/project-management/epics-stories-themes (visited on 2021-01-15) (cit. on pp. 10, 27).

[45] *What is kanban?* URL: https://www.atlassian.com/agile/kanban (visited on 2021-01-15) (cit. on p. 13).

[46] *What is scrum?* URL: https://www.scrum.org/resources/what-is-scrum (visited on 2021-01-15) (cit. on p. 12).

[47] "Why Do We Need Systematic Reviews?" In: *Systematic Reviews in the Social Sciences*. John Wiley & Sons, Ltd, 2006. ISBN: 9780470754887. DOI: https://doi.org/10.1002/9780470754887.ch1 (cit. on p. 20).

[48] *World Energy Consumption Statistics | Enerdata*. URL: https://yearbook.enerdata.net/ (visited on 2020-12-17) (cit. on p. 15).

[49] M.-X. Zhu et al. "A non-functional requirements tradeoff model in Trustworthy Software". In: *Information Sciences* 191 (2012). Data Mining for Software Trustworthiness, pp. 61–75. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2011.07.046 (cit. on p. 5).

A

Sustainability Questionnaire

A.1 Background Information

1. **How many years of experience do you have with Agile Software Development?**
   - < 1
   - 1 - 5
   - 6 - 10
   - > 10
   - No experience
2. **Which Agile method(s) do you work with?** (Multiple selections allowed)
   - Crystal
   - Development Method (DSDM)
   - Dynamic Systems
   - Extreme Programming (XP)
   - Feature Driven Development (FDD)
   - Kanban
   - Lean Development
   - Scaled Agile Framework (SAFe)
   - Scrum
3. **Which of the following best describes your role in your company?**
   - Architect
   - Business Analyst
   - Developer
   - Product Manager

- Product Owner
- Project Lead / Project Manager
- Requirements Engineer
- Scrum Master
- Tester
- Another option... (Free text)

4. **How many people is your Agile project team composed of?**
   - < 5
   - 5 - 10
   - 11 - 20
   - > 20

A.2 Sustainability in Software

1. **Do you address sustainability in software development?**
   - Never
   - Rarely
   - Sometimes
   - Often
   - Always
2. If you have answered yes, in which activities do you address it? (Multiple selections allowed)
   - Requirements Elicitation
   - Software Design
   - Implementation
   - Unit testing
   - System/Integration testing
   - Maintenance
3. If you selected any option above, how do you address sustainability in each of the selected activities?
4. **How do you specify sustainability requirements?** (Multiple selections allowed)
   - User stories

- Use case
- Backlog Item
- Personas
- Acceptance criteria
- Part of the Definition of Done
- Informal text
- UML models
- None
- Another option... (Free text)

5. **Which of the following best describes your knowledge about sustainability in software development?**
   - No knowledge
   - Basic
   - Intermediate
   - Advanced
   - Expert
6. **Do you think you have tools or methods to be able to include sustainability in software development and maintenance?**
   - Yes
   - No
7. If you answered yes to the previous question what are the methods you use? (Free text)
8. **How often do you think software has an impact on sustainability?**
   - Never
   - Rarely
   - Sometimes
   - Often
   - Always
9. In projects where sustainability was addressed, what were the main challenges found by the team? (Multiple selections allowed)
   - Lack of support materials

- Lack of experts on the domain
- Lack of knowledge by the team
- Lack of methods
- Lack of tools
- None
- Another option... (Free text)

10. **What were the main reasons to consider sustainability requirements in your projects?** (Multiple selections allowed)

    - Client requirement
    - Good reputation
    - Legal obligations
    - Social responsibility
    - Organizational requirement
    - Personal motivation
    - To improve the quality of the product
    - Never was considered