ISCTE IUL
Instituto Universitário de Lisboa

Department of Information Science and Technology

Agile or Non-Agile, That Is the Question

Designing a Decision Support System for an Agile Approach in Software Development Projects

Luis Fernando Bordeaux Mattos

Dissertation submitted in partial fulfilment of the requirements for the degree of

Master in Information Systems Management

Supervisor:
Professor Abílio Oliveira, Assistant Professor,
ISCTE-IUL

Co-supervisor:
Professor Ricardo Ribeiro, Assistant Professor,
ISCTE-IUL

September, 2019

Acknowledgments

I want to express my gratitude to my supervisors, Professors Abílio Oliveira e Ricardo Ribeiro, for the support, incentive, and engagement through the learning process of this work. Furthermore, I would like to thank my classmates for the partnership and collaboration throughout the whole course. I wish you all the best in your careers. I am sure our paths will cross again someday.

Also, I would like to thank all research contributors, friends and colleagues of profession (of multiple nationalities), who have willingly collaborated in the various stages of the study. This work would not have been possible without your precious time and enthusiasm demonstrated in the activities.

Finally, I want to thank my family, who have supported me throughout the entire process, keeping me in harmony, even under all the sacrifices we had to make so that this goal could be achieved. I will be grateful forever for your love.

Resumo

Escolher uma abordagem adequada para gerenciar um projeto de desenvolvimento de software é essencial para maximizar as chances de se obter sucesso. Um dos grandes dilemas da atualidade diz respeito à opção por uma metodologia de desenvolvimento Agile ou tradicional. Existem diversas características dos projetos de software e dos ambientes organizacionais onde eles são executados que devemos considerar ao escolher uma opção apropriada. Embora as metodologias de desenvolvimento Agile venham se expandindo e consolidando mundialmente desde o início dos anos 2000 como soluções eficazes para se construir software, elas não se aplicam a todos os cenários de desenvolvimento. Quando usar Agile e qual das suas metodologias é a mais adequada são as grandes questões que pretendemos responder nesta investigação.

Através de uma abrangente revisão de literatura e de um estudo exploratório com profissionais com experiência nas metodologias Agile, exploramos e identificamos os principais fatores que favorecem o uso de uma abordagem Agile. Também estudamos as características de projetos de desenvolvimento de software e de ambientes organizacionais que levam as equipas de desenvolvimento a optar por uma das suas metodologias mais comuns: Scrum, Extreme Programming (XP), Kanban ou Lean Software Development (LSD). Com base nos resultados obtidos, concebemos um modelo conceitual para apoiar a tomada de decisão e desenvolvemos um protótipo de um sistema que implementa tal modelo conceitual. Nosso principal objetivo é esclarecer o que é importante considerar na escolha de uma metodologia Agile e ajudar o tomador de decisão a selecionar uma opção adequada. Os resultados desta investigação enriquecem a literatura voltada para os métodos de seleção de processos de desenvolvimento de software, e contribuem para a difusão do Agile entre as equipas de desenvolvimento e as organizações com nenhum ou baixo grau de maturidade em Agile, mas que estejam interessadas em conhecer mais ou adotar esta abordagem de desenvolvimento.

Palavras-Chave: Agile, sistemas de informação, desenvolvimento de software ágil, metodologias ágeis, adequabilidade do Agile.

Abstract

To know how to choose an approach to manage a software development project is essential to maximize the chances to achieve success. One of the great dilemmas we face nowadays concerns the option for an Agile or a traditional development methodology. There are several characteristics of software projects and the business environments in which they are performed that we must consider while choosing a suitable option. Although Agile development methodologies have been increasingly expanding and consolidating worldwide as effective ways of building software since the early 2000s, they are not a one-size-fits-all approach. When to use Agile and which methodology is most suitable are the great questions we aim to answer in this research.

Through a comprehensive revision of the literature and an exploratory study with Agile practitioners, we explored and identified the main factors that favour the use of an Agile approach. We also unveiled the characteristics of software development projects and organisational environments that lead development teams to opt for one of the common Agile frameworks: Scrum, Extreme Programming (XP), Kanban, or Lean Software Development (LSD). Based on the results obtained, we conceived a conceptual model to support decision making and developed a prototype of an information system that implements this conceptual model. Our major goal in this study is to clarify what is important to consider in the choice of an Agile methodology and help the decision-maker selecting an appropriate development approach. The results of this research contribute to the literature related to processes of selection of software development methodologies, as well as to the diffusion of Agile within development teams and organisations with none or low degree of maturity in Agile, but interested in knowing more or adopting this development approach.

Keywords: Agile, information systems, Agile software development, Agile methodologies, Agile suitability.

Table of Contents

Acknowledgments .... **i**

Resumo .... **ii**

Abstract .... **iii**

Table of Contents .... **iv**

List of Tables .... **vii**

List of Figures .... **viii**

Acronyms and Definitions .... **x**

Chapter 1 – Introduction **1**

1.1. Theme Delimitation 1

1.2. Motivation 2

1.3. Research Question and Objectives 3

1.4. Methodological Approach 4

1.5. Dissertation Structure 4

Chapter 2 – Literature Revision **7**

2.1. Information Technology in the Organisations 7

2.2. Information Systems and Software Engineering 9

2.3. Software Development Methodologies 11

2.3.1. The Waterfall methodology 12

2.3.2. The Spiral methodology 12

2.3.3. The V-Model methodology 13

2.3.4. The Agile approach 14

2.3.5. Traditional vs. Agile 15

2.4. Agile Software Development 15

2.4.1. The birth of Agile 15

2.4.2. The four values and 12 principles of the Agile Manifesto 16

2.4.3. Reasons for adopting Agile 18

2.4.4. Agile suitability filters 20

2.5. Agile Methodologies 23

2.5.1. Scrum 23

2.5.2. Extreme Programming (XP) 26

2.5.3. Lean Software Development (LSD) 29

2.5.4. Kanban 31

2.5.5. A summary of the four Agile methodologies 34

Chapter 3 – Research Methodology **37**

3.1. Part 1 – Literature Revision & Exploratory Study 37

3.2. Part 2 – Prototype Development 39

3.2.1. Initiation phase 39

3.2.2. Planning phase 40

3.2.3. Execution phase 40

Chapter 4 – Exploratory Study 43
4.1. Interview and Focus Groups 43
4.1.1. Interview with an Agile specialist 44
4.1.2. Focus groups 44
4.1.3. Data analysis 45
4.1.4. Brief conclusions from the Exploratory Study 53
4.2. Conceptual Model 54
4.2.1. Step 1 – Agile approach suitability assessment 55
4.2.2. Step 2 – Agile methodology suitability assessment 64
4.2.3. Conceptual model validation 70
4.3. Requirements Gathering 74
4.3.1. High-level system requirements 75
Chapter 5 – Prototype Development 77
5.1. Initiation 77
5.1.1. Users mapping with Personas 77
5.1.2. System requirements specification with User Stories 80
5.1.3. Prioritisation of User Stories with Product Backlog 84
5.2. Planning 85
5.2.1. Activities identification 85
5.2.2. Organisation of activities using a Kanban board 86
5.3. Execution 87
5.3.1. User-System interaction design 87
5.3.2. Screens design 89
5.3.3. Data modelling 94
5.3.4. Prototype development 96
5.3.5. Prototype validation 107
Chapter 6 – Conclusions and Future Research 111
6.1. Conclusions 111
6.2. Research Limitations 112
6.3. Proposals for Future Research 113
6.4. Final Remark 113
Bibliography 115
Annexes 121
Annex A – 13th Annual State of Agile Report - Reasons for Adopting Agile 121
Annex B – 13th Annual State of Agile Report - Respondent Demographics 122
Annex C – 13th Annual State of Agile Report - Benefits of Adopting Agile 123
Annex D – 13th Annual State of Agile Report - How Success is Measured in Agile Initiatives 124
Appendices 125
Appendix A – Focus Group Script 125
Appendix B – Agile Specialist Interview Script 127

Appendix C – Interview Transcript with Agile Specialist [Portuguese] 131
Appendix D – Interview Transcript with Agile Specialist [English] 152
Appendix E – Focus Group Transcript - 1st Group 171
Appendix F – Focus Group Transcript - 2nd Group [Portuguese] 186
Appendix G – Focus Group Transcript - 2nd Group [English].......................... 200
Appendix H – Focus Group Transcript - 3rd Group [Portuguese] 213
Appendix I – Focus Group Transcript - 3rd Group [English] 230
Appendix J – Leximancer Ranked Concept List - Aspects Related to the Choice of an Agile Approach 246
Appendix K – Leximancer Ranked Concept List - Aspects Related to Choosing an Agile Methodology 247
Appendix L – Leximancer Ranked Concept List - System Requirements 248
Appendix M – Agile Suitability Assessment Form (MS Excel) 249
Appendix N – Equations of the Agile Suitability Assessment 254
Appendix O – Conceptual Model Validation Results 256
Appendix P – Prototype Validation Surveys 259

List of Tables

Table 1 - Traditional vs. Agile 15
Table 2 - Top five reasons for using Agile methodologies 18
Table 3 - Top five benefits of adopting Agile methodologies 19
Table 4 - Top five success indicators of Agile initiatives 19
Table 5 - Extreme Programming case study findings 29
Table 6 - Comparative summary of the four Agile methodologies studied 35
Table 7 - Map of questions and distribution of weights by assessment category 63
Table 8 - Agile methodology assessment map of questions and weights 69
Table 9 - Consolidated data of the projects used to validate the conceptual model 70
Table 10 - Results of the Agile approach suitability assessments (non-Agile projects) 72
Table 11 - Consolidated results of the Agile methodology suitability assessments . 73
Table 12 - High-level system requirements 75
Table 13 - User Stories of the high-level system requirement R01 81
Table 14 - User Stories of the high-level system requirement R02 82
Table 15 - User Stories of the high-level system requirement R03 82
Table 16 - User Stories of the high-level system requirement R04 82
Table 17 - User Stories of the high-level system requirement R05 83
Table 18 - User Stories of the high-level system requirement R06 83
Table 19 - User Stories of the high-level system requirement R07 84
Table 20 - User Stories of the high-level system requirement R08 84
Table 21 - The Product Backlog 85
Table 22 - System tables 96
Table 23 - User Stories for the prototype 97
Table 24 - Answers of the rating scale questions 108
Table 25 - Answers of the open-ended questions 108
Table 26 - Organisational Environment category detailed percentages 254
Table 27 - People category detailed percentages 255
Table 28 - Project Environment category detailed percentages 255
Table 29 - Agile approach suitability results (56 projects) 256
Table 30 - Agile methodology suitability results (49 projects) 257

List of Figures

Figure 1 - Porter and Millar's model. Source: Strategic Management Insight (Jurevicius, 2013)...... 8

Figure 2 - The phases of the Waterfall methodology. Source: ResearchGate (Calikus, 2015) 12

Figure 3 - The phases of the Spiral methodology. Source: Easy Projects (Icasas, 2013)......13

Figure 4 - The phases of the V-Model methodology. Source: LinkedIn (Asim, 2016) 13

Figure 5 - The Agile approach. Source: AgilePM® V2 (Dąbrowski, 2017) 14

Figure 6 - The four values of the Agile Manifesto 16

Figure 7 - The Slider model. Source: Leading Answers (Griffiths, 2007)......20

Figure 8 - Boehm and Turner radar chart. Source: Leading Answers (Griffiths, 2007)......22

Figure 9 - The Scrum lifecycle. Source: Microsoft (Boer, 2017)......23

Figure 10 - Scrum Board example. Source: Agile Velocity (Hawks, 2010) 24

Figure 11 - Burndown Chart example. Source: Scaling Your Business (Gonçalves, 2019)......25

Figure 12 - The Extreme Programming lifecycle. Source: ANTOSSOFTWARETOPICS (S., 2016) 27

Figure 13 - Cumulative Flow Diagram example. Source: Zepel (Jaikrishnan, 2019) 30

Figure 14 - Kanban Board example. Source: Software Management Project (Brodzinski, 2015)......32

Figure 15 - Literature Revision & Exploratory Study 39

Figure 16 - Prototype Development - Initiation phase......40

Figure 17 - Prototype Development - Planning phase 40

Figure 18 - Prototype Development - Execution phase......41

Figure 19 - Map of the concepts related to the choice of an Agile approach......46

Figure 20 - Map of the concepts related to choosing an Agile methodology 49

Figure 21 - Map of concepts related to the essential requirements for the system proposed......51

Figure 22 - Agile suitability assessment flow (steps of the assessment)......54

Figure 23 - Relevant aspects to assess by category of evaluation 55

Figure 24 - Is an Agile approach appropriate? 64

Figure 25 - Relevant aspects of the Agile methodology suitability assessment......64

Figure 26 - Development approaches used 71

Figure 27 - Agile methodologies applied (when the approach is Agile) 71

Figure 28 - Persona card - John......78

Figure 29 - Persona card - Angela 79
Figure 30 - Persona card - Peter 79
Figure 31 - Persona card - Anthony 80
Figure 32 - Kanban board (Trello) 86
Figure 33 - Use Cases - Agile Suitability Assessment 88
Figure 34 - Screens mapping 89
Figure 35 - Main page 90
Figure 36 - Project Information (Entry Data) 90
Figure 37 - Ongoing Assessments 91
Figure 38 - Project Information (View / Update Data) 91
Figure 39 - Completed Assessments 91
Figure 40 - Agile/Methodology Suitability Assessment Questionnaire (1) 92
Figure 41 - Agile/Methodology Suitability Assessment Questionnaire (2) 92
Figure 42 - Agile Suitability Assessment Result 93
Figure 43 - Agile Suitability Consolidated Result 93
Figure 44 - Class diagram 94
Figure 45 - Relational model 95
Figure 46 - Main page 99
Figure 47 - Project data entry page 99
Figure 48 - Questionnaire page - Organisational Environment (question 1 of 4) ..100
Figure 49 - Questionnaire page - Organisational Environment (question 2 of 4) ..100
Figure 50 - Questionnaire page - Organisational Environment (question 3 of 4) ..101
Figure 51 - Questionnaire page - Organisational Environment (question 4 of 4) ..101
Figure 52 - Questionnaire page - People (question 1 of 2) 101
Figure 53 - Questionnaire page - People (question 2 of 2) 102
Figure 54 - Questionnaire page - Project Environment (question 1 of 3) 102
Figure 55 - Questionnaire page - Project Environment (question 2 of 3) 102
Figure 56 - Questionnaire page - Project Environment (question 3 of 3) 103
Figure 57 - Agile approach suitability assessment (result page) 103
Figure 58 - Agile methodology suitability assessment (question 1 of 5) 104
Figure 59 - Agile methodology suitability assessment (question 2 of 5) 104
Figure 60 - Agile methodology suitability assessment (question 3 of 5) 105
Figure 61 - Agile methodology suitability assessment (question 4 of 5) 105
Figure 62 - Agile methodology suitability assessment (question 5 of 5) 105
Figure 63 - Agile methodology suitability assessment (results page) 106
Figure 64 - Project list of ongoing assessments 106
Figure 65 - Project list of completed assessments 107

Acronyms and Definitions

AI – Artificial Intelligence

CIO – Chief Information Officer

CTO – Chief Technology Officer

DBMS – Data Base Management System

DM – Data Mining

DSS – Decision Support System

DW – Data Warehouse

EC – European Commission

ERP – Enterprise Resource Planning System

ESS - Executive Support System

GSD – Global Software Development

IEEE – The Institute of Electrical and Electronics Engineers

IS – Information Systems

IT – Information Technology

LSD – Lean Software Development

MIS – Management Information System

PMI – Project Management Institute

PMO – Project Management Office

SDLC – Software Development Life Cycle

SME – Subject Matters Expert

SPAs – Single Page Applications

TPS^1 – Toyota Production System

TPS^2 – Transaction Processing System

UML – Unified Modelling Language

XP – Extreme Programming

Chapter 1 – Introduction

1.1. Theme Delimitation

One of the greatest challenges in any project is to succeed in achieving its objectives within the classic parameters of scope, time, cost, and quality. In software development projects, this mission becomes even tougher since what is at stake is something that is about to be invented and for which there is no unique recipe to succeed.

The Project Management Institute (PMI)[1] states in the Life Cycle Selection chapter of the "Agile Practice Guide" that there are many project formats and a variety of ways to undertake them. Therefore, project teams need to study the project's characteristics and know the organisation's environment to define the approach is most likely to fit (PMI, 2017). It reinforces the idea that to increase the chances of the project succeeding; it is key to choose a suitable methodological approach with adequate tools and techniques. Many projects fail or reach only partial goals because they do not adopt an appropriate methodology (Griffiths, 2015).

In software development projects the last decade showed an increasing need to produce solutions in a simpler, faster, and cheaper way, as well as with greater flexibility regarding the changes in requirements due to the dynamics of the business environments. Sutherland (2015) explains that the linear approach with one-way analysis, design, construction, and test became outdated concerning the adaptability to an evolving scenario, which cleared the way to the emergence of Agile[2] methodologies, capable of delivering faster results, more aligned to the customer's needs.

Thus, in response to the needs of software developers and a dynamic market, Agile software development methodologies have conquered more and more space in the software development industry. However, despite this trend, Agile, like any other process, is not a one-size-fits-all solution. When to use Agile and which of its methodologies is more suitable for the problem

[1] The Project Management Institute (PMI) is a global non-profit professional organisation for project management that offers globally recognized project management standards and certification programs (Project Management Institute, 2019).

[2] The term "agile" has two related but different meanings: the usual understanding, which refers to "the ability of moving quickly and easily"; and the meaning related to the class of software development approaches (Kourounakis et al., 2015). To differentiate the second from the first one, "Agile" is written in this document with a capital letter.

to be solved is a complex task given the many combinations of factors that influence the choice of an appropriate option.

In addition to the technical matters involved in decision-making, there is a human tendency to base decisions on unconscious biases rather than facts and data. As explained by Kahneman, Slovic and Tversky (1982) in their book "Judgment under Uncertainty", such unconscious biases are like shortcuts created by the mind enabling it to make faster decisions (usually over known data and recurrent situations), which can be positive for everyday events, but may lead to misjudgements when further information must be analysed.

Therefore, this study focuses on clarifying what is important to consider in the choice of an Agile software development methodology, aiming to support the decision-maker in selecting an adequate approach.

1.2. Motivation

I have been working in the area of software development since 1994 and, more specifically, managing software and other technology-related projects since 2005. In 2007 I had my first contact with Agile methodologies, using the frameworks[3] Extreme Programming and Scrum in software development projects.

On several occasions, when using an Agile methodology would be appropriate to run a given project, I came across the situation of having to convince the sponsor or business representatives about the benefits of using such methodology. Agile's lack of knowledge by these stakeholders was one of the recurring challenges I have experienced in many years managing software projects.

Despite that, most of the time, I succeeded in raising awareness of the decision-maker, but often expending an effort that could have been used in benefit of the project. Thus, having an instrument that could support the justification for using an Agile methodology would give the decision-maker more confidence to decide and could speed up the process. A tool like that could enable a correct and faster choice.

The everyday life and the dynamics of different professional environments have not given me enough room to work towards solving this problem and

[3] In software engineering, a framework is a software (or system) development methodology used to structure, plan, and control the process of developing an information system (Sommerville, 2016).

dedicate time to build such a tool. Besides, this is a solution that would require non-trivial research and development effort. In spite of this, the desire of dwelling on a solution to this problem was always there, waiting for an opportunity. The opportunity has arisen through this master's dissertation.

Although the main motivation for this work lies in my professional experiences, this is an issue that can affect organisations whose activities, primary or secondary, include or are related to the development of applications and information systems, especially those that do not have a well-defined decision process with this purpose. Therefore, this work may also benefit these organisations.

1.3. Research Question and Objectives

In this context, the following research question becomes pertinent: How can an information system help in choosing an Agile methodology?

Therefore, our research function focuses on determining how to develop a system to justify when the option for an Agile methodology is preferable, compared to others, to support software development.

We intend to answer this question through the study of the characteristics of software development projects and organisations, identifying those that favour the use of Agile methodologies. Based on the results of this study, we will also propose a conceptual model that supports the choice decision and, accordingly, develop a prototype that implements this decision model.

In terms of general objectives, we aim to:

1) Determine the suitability criteria of a software development project for an Agile approach or, in other words, identify what is determinant for the adoption of an Agile methodology in a software development project;
2) Identify the main aspects to consider in the determination of the best Agile methodology to use in a given software development context (limited to the Agile methodologies Scrum, Extreme Programming, Kanban, and Lean Software Development);
3) Elaborate a decision model based on the suitability criteria (determined in objective 1) and on the main aspects (identified in objective 2);
4) Validate the proposed decision model (based on a collection of historical data of software development projects);

5) Model and design the information system that will implement the proposed decision model (using Agile practices and techniques);
6) Develop (and validate) a prototype for the proposed system.

1.4. Methodological Approach

The research methodology applied in this dissertation falls into two distinct but complementary approaches. A conceptual-analytical approach combined with an artefacts-building approach (Järvinen, 2000).

An extensive literature revision[4], complemented by an exploratory study with the application of focus groups[5] and an interview with one Agile specialist, provided diverse data sources that were analysed and originated the conceptual model proposed. These activities are part of the conceptual-analytical approach.

The previous stage was the basis for the next steps, including the design, modelling, and development of the information system prototype. This work fits in the category of an artefacts-building approach since the prototype implements the conceptual model, supporting the assumption that an information system based on the conceptual model proposed can be built.

Both approaches complement each other because the development of the system prototype is oriented towards the implementation of the proposed conceptual model and is intended to produce the results of this model.

1.5. Dissertation Structure

This document is organised in six chapters.

Chapter 1 (Introduction) contextualizes the work through the delimitation of the theme under study and the motivation for the work. The research question and the objectives of the work are also described.

Chapter 2 (Literature Revision) presents the main concepts related to the theme researched and review previous studies.

[4] A literature revision surveys books, scholarly articles, and other relevant publications to a particular subject of research, allowing a critical analysis of these works on the subject of investigation. It provides an overview of the sources explored while researching a particular topic, allowing the readers to see how the actual research fits in a wider area or field of study (University of Southern California, 2019).

[5] Focus groups are group interviews that relies on the interaction within a group of people over subjects introduced by the researcher. The idea is that the group interaction can produce insights on the subjects discussed that would be hardly available without the interaction among the participants (Morgan, 1997).

Chapter 3 (Research Methodology) introduces the research work, explaining the methodology used, the phases of the work, and the articulation between these phases.

Chapter 4 (Exploratory Study) involves the data collection and analysis of the research process. It also gathers and defines the high-level requirements for the proposed information system.

Chapter 5 (Prototype Development) describes the work necessary for the implementation of the system prototype.

Chapter 6 (Conclusions and Future Research) presents an overall assessment of the research, and the objectives reached. It also describes the limitations of the study and identifies possible avenues for expanding and continuing it in the future.

Chapter 2 – Literature Revision

2.1. Information Technology in the Organisations

The competitiveness faced by organisations nowadays requires them to have more agile and flexible management processes, where the utilization of Information Technology (IT) is critical not only for the success of their businesses but for their survival in a constantly evolving market.

Luftman, Lewis, and Oldach (1993) stated that IT, in a broader way, encompasses information systems, hardware and software, automation, telecommunications, and multimedia resources used to provide data, information, and knowledge. In this context, it can be said that IT is not only used by the business environment but is integrated to it, representing an essential element for the functioning of the three organisational levels (operational, tactical and strategic).

Initially, the information technology did not have such a significant role in the organisations, with the use of IT resources limited to the necessity of performing automated tasks (Coutinho, 2017). However, with technological advances, improvements in the area of software, hardware, network, and telecommunications broadened the benefits of IT. As a consequence, the application of technology in the organisational environment changed, becoming essential for the management of information at all levels. More recently, the constant reduction of costs with IT infrastructure and the increased offering of Cloud computing services have amplified even more the general access to the benefits of information technology (Coutinho, 2017).

To justify the importance of IT for the organisations, Michael Porter and Victor Millar (2004) used the concept of "value chain" and "value system". The value chain is the set of activities performed by the company, divided into two categories: activities that add value to the business, known as primary activities, and support or secondary activities. The primary activities are subdivided into inbound logistics activities (receipt and stock of raw materials), operations, outbound logistics (distribution), marketing, sales, and after-sales. The value system, also known as the supply chain, is composed of the union of the value chains from several companies and suppliers, forming a complete chain that goes from the raw material to the final consumer. According to the authors, the Information Technology assumes strategic importance for the organisation from the moment it enables changes in the way it performs each of the activities of the value chain, aiming to promote greater efficiency and performance.

Figure 1 presents Porter and Millar's value chain model, where the organisation's primary activities (in blue) are supported by the secondary activities (in orange).

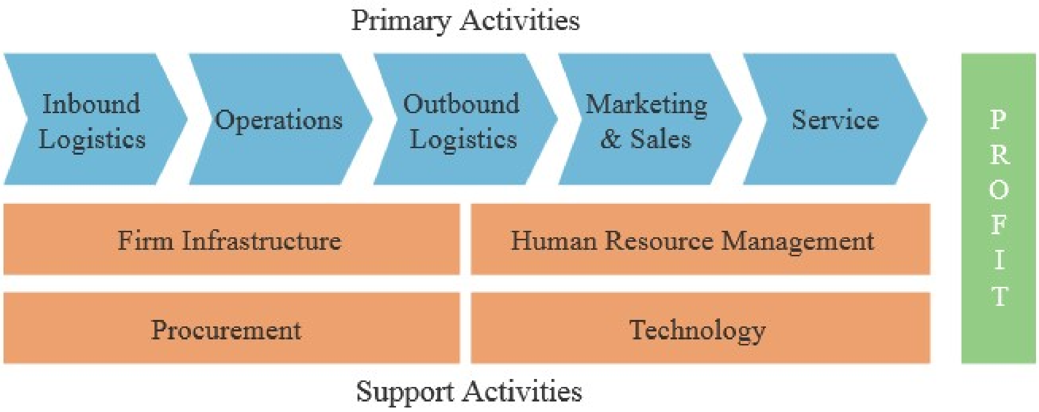


Figure 1 - Porter and Millar's model. Source: Strategic Management Insight (Jurevicius, 2013)

Being a vital component for the organisation's operations, the company needs to identify the information and technologies that support and influence the value chain. In this context, having a plan for information technology management is, therefore, mandatory for all types of companies that deal with IT in a well-defined manner.

According to Rodrigues (2017), one of the key aspects covered by IT planning involves the management of the organisation's technological infrastructure, communications, and information systems portfolio. Regarding the systems portfolio, they are composed by specific business tools, as well as solutions like Enterprise Resource Planning (ERP), Decision Support Systems (DSS), Database Management Systems (DBMS), Data Warehouse (DW), Artificial Intelligence (AI), Data Mining (DM), Telecommunication Systems and specific packages that need to be acquired or developed by the organisation. In this context, the companies began to organise themselves to respond to this need either by having their own systems development and maintenance structure or contracting outsourced services from specialized suppliers.

Therefore, to properly define their IT strategy, the organisations must seek alternatives that permit them to have an efficient IT structure, allowing them to focus on their essential businesses. This structure should be flexible enough to facilitate the gradual implementation of changes, enabling a continuous improvement process. Finding such alternatives is not a trivial process, and the companies often need to resort to subject matters experts (SMEs), which has created an opportunity for a wide variety of information technology consulting firms to grow and expand.

In addition to Porter and Millar's approach, it is also relevant to note that, in the last 20 years, we observed the raising of new markets and the disappearance of traditional ones. We also witnessed the virtual extinction of geographical and cultural boundaries and the collapse of market blockades. To respond and adapt to these changes accordingly, Information Technology had to become a discipline capable of providing solutions that enable the organisations to execute and disseminate new business strategies (Rodrigues, 2017). In a world so dependent on technology, information, and the innovations driven by them, the ability to foresee problems and anticipate distinct scenarios, analyse data more accurately, and make decisions over reliable information are essential capabilities for doing business as well.

John McCarthy, a Stanford University professor and one of the fathers of Artificial Intelligence, predicted in 1961 that computing would someday be organised as public utilities (Coutinho, 2017). McCarthy was right since IT is nowadays as important as the great variety of commodities that exist to meet the basic needs of people in everyday life.

2.2. Information Systems and Software Engineering

Mattos (2005) defines Information System (IS) as a system specialized in processing data or information, comprising a set of communication, control, memory and processor modules interconnected through a network where the logical relations between the modules are defined by the programs executed by the information system.

To understand the comprehensiveness of the information systems in the organisations it is important to know how they are generally classified. In this context, Sommerville (2016) states that information systems can be structured in organisational levels, functional areas, type of support, and role. In the book "Software Engineering", the author presents this classification in more detail.

- Systems classification by organisational levels:
    - Departmental Systems are systems developed for the departmental areas of an organisation;
    - Business Systems are a set of departmental systems, combined with other applications;
    - Interorganisational Systems are information systems that interconnect several companies or branches.
- Systems classification by functional areas:
    - Information systems at the departmental level support the functional areas of the company. The main functional information systems are Accounting, Financial, Industrial, Marketing, and Human Resources Information Systems.
- Systems classification by type of support:
    - Transaction Processing System (TPS) supports functional activities and administrators;
    - Management Information System (MIS) supports functional activities and management;
    - Decision Support System (DSS) supports decision making by managers and analysts;
    - Executive Support System (ESS) supports the chief executive in decision making.
- Systems classification by role in the organisation:
    - ERP - Enterprise Resource Planning Systems have operational focus and integration of business processes. They are usually divided into modules, and the main idea is the integration of business information. ERPs can be considered a breakthrough in transaction processing systems and management information systems;
    - BI - Business Intelligence Support Systems focus on decision support at the managerial and strategic levels. ERP can be considered a BI entry. Some manufacturers provide the two options together. BI is as if it were the advance of decision support systems (DSS) and executive support systems (ESS).

According to Fonseca (1997), to be effective, regardless of system classification, the information systems must meet expectations such as: meeting the real needs of users; sustain costs compatible with their reality;

be adaptable to new technologies; and be aligned with the company's business strategies. By meeting these requirements, the organisation is prepared to, at the right time, use its systems with higher chances of success. However, to satisfy these requirements, the organisation must be supported by well-defined software engineering processes.

Schmidt (2013) defines software engineering as a discipline that combines technical and project management practices to design and build software. It focuses on the product environment while considers its entire development cycle. The software product is the goal of the software engineering effort and the numerous ways of implementing this effort originates the broad range of software development lifecycles (or methodologies) that exist. The core practices associated with software engineering can be seen in the different ways each of these methodologies performs the activities needed to build the software.

2.3. Software Development Methodologies

Murch (2012) defines Software Development Life Cycle (SDLC) as a repeatable process for building information systems containing a set of guidelines and standards. Also known as Software Development Methodology, every IT organisation should have one so that development teams may follow its rules and guidelines, and apply its principles in development activities.

A software development methodology delivers value to an organisation by addressing business needs within the software development environment. The utilization of a suited methodology allows project teams to increase systems development efficiency, reduce risks, and eliminate waste. In addition, it leverages design and reuse, increasing the overall quality of the resulting software (Murch, 2012).

For many years, since the emergence of the first computers in the 1940s until the beginning of the 21st century, what prevailed in relation to software development methodologies was the use of practices strongly derived from engineering processes, which resulted in the so-called traditional software development methodologies such as the Waterfall, the Spiral, the V-Model methodologies. However, in recent years, an increasing necessity to build systems faster, cheaper and with higher flexibility to face the dynamics of the markets has given space to the emergence of Agile software development methodologies, best suited to meet such need.

2.3.1. The Waterfall methodology

Balaji and Murugaiyan (2012) define Waterfall as a software development methodology in which the system is built following sequential steps that flow downwards without overlapping, like a waterfall (cf. Figure 2). The steps of the development occur in the same order, and each step has a defined time constraint for the tasks to be accomplished.

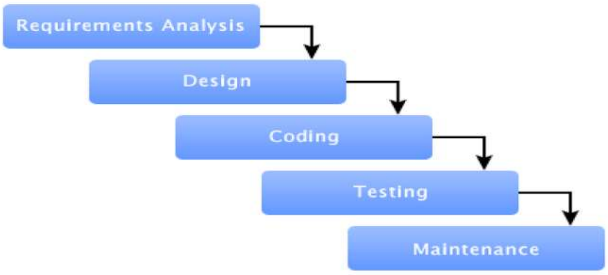


Figure 2 - The phases of the Waterfall methodology. Source: ResearchGate (Calikus, 2015)

According to the researchers, in the Waterfall methodology, the activities and deliverables of each phase are fully completed before moving ahead. The product's requirements must be clearly defined so the design may initiate. Once the design finishes, code implementation can start. Testing is done after the code has been totally developed. The last stage promotes the support and maintenance required to keep the system updated. In the Waterfall methodology, the documentation is produced at the end of each phase.

The Waterfall methodology produces major benefits for large and complex software development initiatives, with strict phases and fixed deadlines. On the other hand, one of the main problems with the methodology is that the defects are usually found later in the development cycle because the quality team is not involved in the project from the beginning (Royce, 1970).

2.3.2. The Spiral methodology

According to Boehm (1988), the Spiral methodology emerged from experiences with Waterfall. The concept of this development process is based on a progression of development activities performed repeatedly (cf. Figure 3). In the Spiral methodology, the system is built iteratively, and each cycle is responsible for producing a part of the system. We can say that the Spiral methodology corresponds to the Waterfall methodology executed the number of times necessary for the complete development of the software.

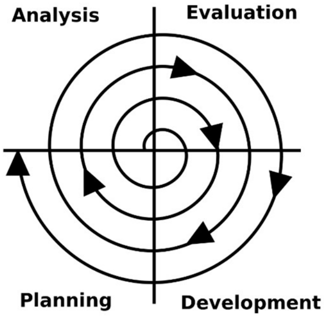


Figure 3 - The phases of the Spiral methodology. Source: Easy Projects (Icasas, 2013)

2.3.3. The V-Model methodology

The V-Model (from Validation & Verification) is a software development methodology also derived from the Waterfall methodology. However, instead of following a linear path, the phases are bent upwards after coding stage forming a "V" shape (cf. Figure 4).

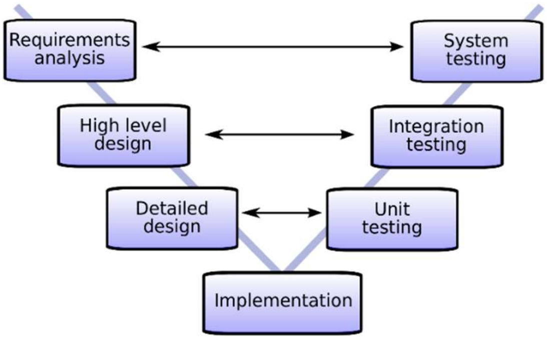


Figure 4 - The phases of the V-Model methodology. Source: LinkedIn (Asim, 2016)

The V-Model methodology relies on the verification and validation of the deliverables[6] from the current phase before moving forward. Each development step is linked to a testing step: system's test scenarios are defined based on requirements and functional specifications; integration tests are based on high-level design; unit tests are based on detailed design. After the specifications are produced (the first leg of V), coding starts. Once coding finishes, the successive testing steps are performed (the second leg of V).

[6] In project management, a deliverable is a product (or service) produced as a result of the project effort and is intended to be useful to the client of the project. It can be a software, a document, a training program or other asset required by the project (Burley, 2017).

Balaji and Murugaiyan (2012) affirm that, in the V-Model, requirements changes are possible in any phase, which is one of the greatest advantages of the model. Nevertheless, researchers state that its biggest disadvantage is that V-Model methodology is not flexible, leaving no room for customizations in the process. Besides that, they affirm that the model is not a good choice for short-term projects, and it is best suited for large organisations that work with big development teams.

2.3.4. The Agile approach

Martin (2003) presents Agile as an approach for software development based on an iterative and incremental concept[7] that instead of in-depth planning at the beginning of the project, embraces changing requirements over time and encourages constant feedback from the client. Agile cross-functional teams work on iterations to build the software over a fixed period of time, and this work is organised into a product backlog[8], which is prioritised based on business values. The goal of each development cycle is to produce valuable working software.

Using Agile, leadership encourages teamwork, accountability, and face-to-face communication. Business stakeholders and developers must work together to align the product with customer needs and company goals. More than the application of these principles in combination, Agile involves adopting a new mindset based on Agile values and principles. Figure 5 shows that these values and principles are the central elements of Agile practices that, combined, form the different types of Agile methodologies.

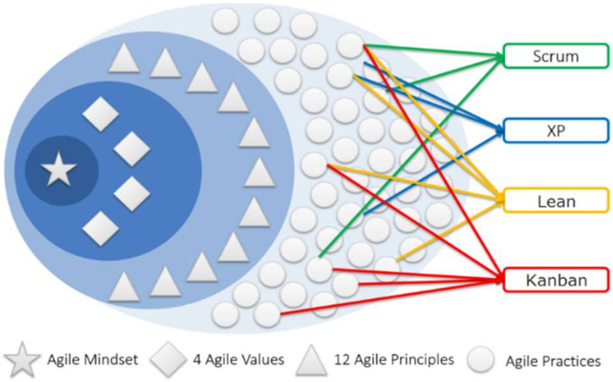


Figure 5 - The Agile approach. Source: AgilePM® V2 (Dąbrowski, 2017)

[7] A development concept based on a cyclic process of designing, coding, testing, and refining the software, adding more functionality each cycle, until the final version is ready (Larman & Basili, 2003).

[8] A product backlog is a list of new features, changes to existing features, bug fixes, infrastructure changes or other related products an Agile team delivers as part of an Agile development effort (PMI, 2017).

2.3.5. Traditional vs. Agile

Table 1 summarizes the main differences between traditional software development methods and Agile methods (SCRUMstudy, 2014).

Table 1 - Traditional vs. Agile

	Traditional	Agile
The emphasis is on	Processes	People
Development approach	Linear	Iterative & Incremental
Management approach	Centralized	Decentralized
Upfront planning	High	Low
Organisation	Managed	Self-organised
Documentation level	Comprehensive	Minimal (just enough)
Requirements prioritisation	Fixed in the plan	Based on business value
Change management	Formal / Low flexibility	Informal / High flexibility
Leadership style	Command & Control	Collaborative / Servant
Performance measurement	Plan conformity	Business value
Quality assurance	Process centric	Customer centric
Customer engagement	Depending on the process lifecycle	High throughout the development effort

2.4. Agile Software Development

2.4.1. The birth of Agile

The concepts of Agile software development were officially conceived in February 2001, when 17 software developers gathered in a hotel in the United States state of Utah to discuss and propose new software development methods as alternatives to the traditional approaches in force, considered for them as documentation driven and heavy weighted processes (Martin, 2003). The output of this meeting is known as the "Manifesto for Agile Software Development" or, the Agile Manifesto.

The signatories included in the Agile manifesto what they believe to be “better ways of developing software by doing it and helping others do it” (Beck et al., 2001). The pillars of the manifesto are supported by four essential values and 12 principles in strong contrast to the traditional software development practices and standards. Comprehending these values and principles is the key to understand the Agile mindset.

2.4.2. The four values and 12 principles of the Agile Manifesto

The Manifesto for Agile Software Development comprises four foundational values that lead the Agile approach to software development. Griffiths (2015) affirms that each Agile methodology applies the four values in different ways, but all of them rely on these values to guide the development. A good way to think about the manifesto is that it defines preferences, not alternatives, encouraging the focus on certain areas, but not eliminating others (Kourounakis et al., 2015). Figure 6 depicts the four values of the Agile Manifesto.

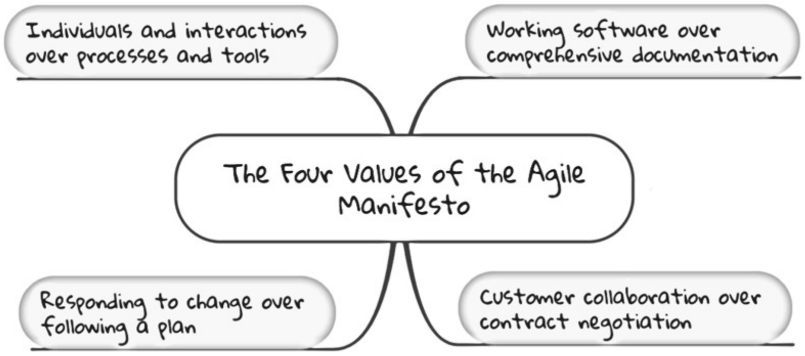

Individuals and interactions over processes and tools
Working software over comprehensive documentation
The Four Values of the Agile Manifesto
Responding to change over following a plan
Customer collaboration over contract negotiation

Figure 6 - The four values of the Agile Manifesto

Griffiths (2015) explains what is behind the four Agile values:

- Value 1 "Individuals and Interactions Over Processes and Tools": Software is built by people. To increase the chances of success, people need to work together and collaborate. It must include all stakeholders and not only the project team. Valuing people more than processes and tools does not mean that process and tools are not important. They are still critical to software development. Without the right tools and processes, the project team will face difficulties. The point is that they must meet the team's needs, not the other way around.
- Value 2 "Working Software Over Comprehensive Documentation": Agile values documentation, but values working software more. Documentation has its place and can be a great resource for users who wonder what the software does and how it works. Some projects often require extensive documentation before the development of the software; however,

spending excessive time on them can be needless. For the team to incorporate the Agile mindset, it is necessary to comprehend that the primary goal of software development is to deliver a solution that offers benefits to the customer rather than extensive documentation.

- Value 3 "Customer Collaboration Over Contract Negotiation": Successful Agile teams work closely with their clients and communicate with them frequently. It is not a customer's role to generate the ideas that guide the systems building; however, by getting frequent feedback, the project team will be able to understand what the customers need. Also, it is important to separate the relationship between the team and the customers from the relationship with the product. Contract negotiations are necessary, but they sometimes can stand a barrier between the team and the customer, which does not help the creation of an ideal solution.
- Value 4 "Responding to Change Over Following a Plan": After planning or even after the beginning of the development of the system, the customer may change his mind about what he/she wants. It may be due to several reasons. Because of that, some code may be dropped and time lost, but if the team works with short iterations, the lost time may be minimized. Changes are a reality (sometimes the only certainty) in a software development project, and there is no surprise in a plan that needs to be modified. Thus, the team must be flexible enough to embrace changes as the scenario changes, otherwise the plan will become irrelevant, and the resulting software may not be useful to the client.

Beck et al. (2001) wrote the 12 principles of the Agile Manifesto aiming to guide the Agile implementation clearly describing a culture in which change is welcomed, the customer is the focus of the work, and the effort to build the software is aligned with business needs:

1) "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software."
2) "Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage."
3) "Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale."
4) "Business people and developers must work together daily throughout the project."

5) "Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done."
6) "The most efficient and effective method of conveying information to and within a development team is face-to-face conversation."
7) "Working software is the primary measure of progress."
8) "Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely."
9) "Continuous attention to technical excellence and good design enhances agility."
10) "Simplicity – the art of maximizing the amount of work not done – is essential."
11) "The best architectures, requirements, and designs emerge from self-organising teams."
12) "At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly."

2.4.3. Reasons for adopting Agile

Since the formal creation of Agile concepts in 2001, the software industry has been gradually increasing the adoption of Agile practices. There are many reasons for that; however, more broadly, we can say that the need for a more flexible and efficient process to build software due to a dynamic business environment is the major reason for the growth of Agile utilization. This statement is supported by the results of "The 13th Annual State of Agile Report", a survey conducted by CollabNet VersionOne (2019). It points out the top five reasons for the adoption of Agile methodologies as:

Table 2 - Top five reasons for using Agile methodologies [9]

#	Reason	Percentage of respondents
1st	Accelerate software delivery	74% of respondents
2nd	Enhance the ability to manage changing priorities	62% of respondents
3rd	Increase productivity	51% of respondents
4th	Improve business / IT alignment	50% of respondents
5th	Enhance software quality	43% of respondents

[9] See Annex A for the complete list of reasons for using Agile published in the survey.

The survey involved a broad range of industries in the global software development community and analysed 1.319 responses between August and December 2018. As in previous years, CollabNet VersionOne collected responses from a diverse set of organisation sizes, geographic locations, roles, and industries[10].

The study also identified the main benefits of adopting Agile (cf. Table 3).

Table 3 - Top five benefits of adopting Agile methodologies [11]

#	**Benefit**	**Percentage of respondents**
1st	Ability to manage changing priorities	69% of respondents
2nd	Project visibility	65% of respondents
3rd	Business / IT alignment	64% of respondents
4th	Team morale	64% of respondents
5th	Delivery speed / time to market	63% of respondents

The survey provides a clear indication that Agile adoption has been steadily growing. According to 22% of respondents, all of their teams are using Agile methodologies. Just two years ago, this rate was less than 8%.

Also, at the project level, respondents reported that their organisations recognize Agile success. 95% said that at least some of their Agile projects were successful, and 48% reported that most or all of their Agile projects were successful. The top five indicators commonly used to measure success within Agile initiatives (cf. Table 4), according to the respondents, are:

Table 4 - Top five success indicators of Agile initiatives [12]

#	**Success indicator**	**Percentage of respondents**
1st	Customer / user satisfaction	52% of respondents
2nd	Business value	48% of respondents
3rd	On-time delivery	41% of respondents
4th	Quality	38% of respondents
5th	Productivity	33% of respondents

[10] See Annex B for details about the respondent demographics.

[11] See Annex C for the complete list of benefits of adopting Agile published in the survey.

[12] See Annex D for the complete list of success indicators of Agile initiatives.

2.4.4. Agile suitability filters

Agile suitability filters are models to help determine the likely fit or gap for using Agile approaches. These models assess factors associated with Agile suitability and then provide recommendations (PMI, 2017). Four assessment models for use as Agile suitability filters were identified in the scope of this research. It is important to note that these models are not meant to check the Agile suitability to an organisation; instead, they focus on indicating the appropriateness of adopting an Agile approach in the scope of software projects. Both approaches have quite different goals.

2.4.4.1. The Slider

The Slider is the oldest Agile suitability model created. Its intention is to show that an Agile or Traditional choice is not necessarily binary (Griffiths, 2007). The model was designed inspired by an old-fashioned car heater control (cf. Figure 7), allowing for: "No", a "Mixture", or a "Fully" Agile approach, indicated by the arrow. Characteristics that would pull the pointer one way or another are depicted in Figure 7.

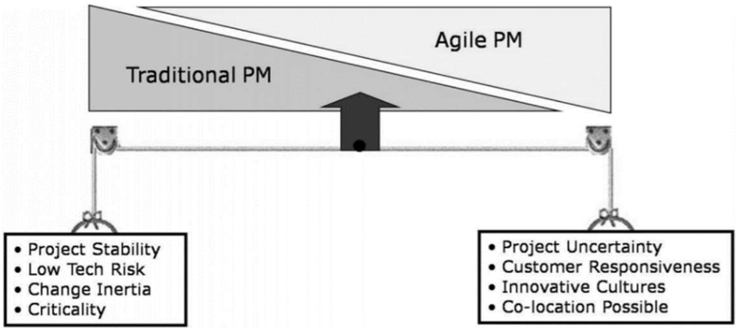


Figure 7 - The Slider model. Source: Leading Answers (Griffiths, 2007)

The occurrence of project uncertainty or instability; high customer responsiveness; the existence of an innovative culture; and the possibility of co-location[13], would pull the pointer towards the Agile side. Otherwise, Agile usage would be more difficult and might favour a more traditional approach. In the case of mixed results, a combination of Agile and traditional methods could suit best.

[13] Co-location is an Agile practice where the members of the project team work at the same physical location allowing face-to-face interactions and promoting higher collaboration (Griffiths, 2015).

2.4.4.2. The DSDM suitability filter

Derived from the Dynamic Systems Development Method (DSDM)[14], this model consists of a list of "Yes / No" questions whose idea is to check conformance to project characteristics that favour Agile (Griffiths, 2007):

- The acceptance of the Agile philosophy before starting work;
- The acceptance of empowered teams;
- The commitment to provide significant end-user involvement;
- Incremental delivery;
- Easy access by the team to customers and end-users;
- The stability of the team;
- The development of team skills;
- The size of the team;
- A supportive relationship (trust and collaboration);
- The development technology.

These characteristics are measured through questionnaires. The presence or absence of these attributes indicates the level of adequacy to using Agile. Negative answers do not mean that Agile is not adequate; rather, they highlight potential risk areas that require attention.

2.4.4.3. The Boehm & Turner model

Barry Boehm and Richard Turner (2003) in their book "Balancing Agility and Discipline: A Guide for the Perplexed" created a visual way of assessing the project characteristics they claim determine suitability to an Agile approach. The idea is that the project is assessed through five attributes:

1) Personnel – Agile projects run better with a low number of beginners and a higher number of experienced developers;
2) Dynamism – Measures the likelihood of changes in the project. If it is high, then the project is more suited to an Agile approach;
3) Culture – Verifies whether the organisation can accommodate concepts such as emergent requirements, empowered teams, and servant leadership (Agile prone);

[14] The Dynamic Systems Development Method (DSDM) is an Agile methodology focused on the project lifecycle (with an iterative approach to developing software) and delivering real value to the business (Griffiths, 2015).

4) Team Size – Agile methods are easier to execute and manage with small teams;
5) Criticality – Refers to the consequence of a system failure. Agile is more suitable for applications where failure results in a loss of convenience.

The scores are plotted on a radar diagram (cf. Figure 8):

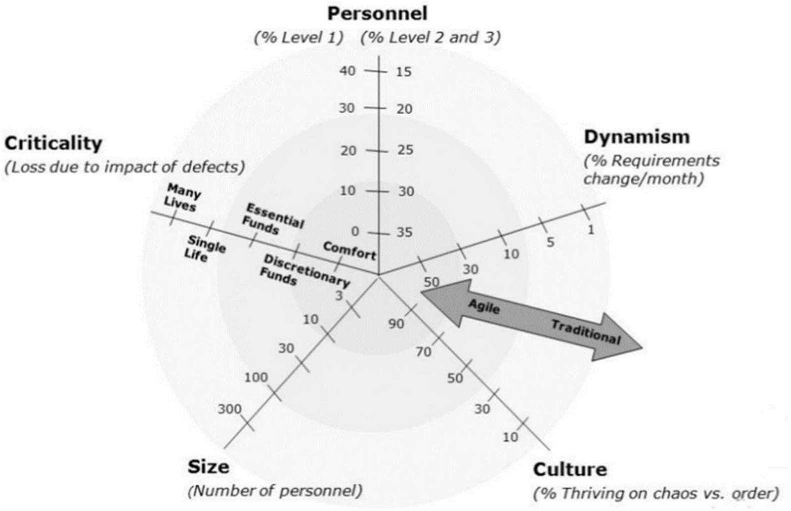


Figure 8 - Boehm and Turner radar chart. Source: Leading Answers (Griffiths, 2007)

Scores towards the centre indicate a good fit for Agile, while scores towards the outside indicate a better fit for a more traditional approach.

2.4.4.4. The PMI model

Having as base the previous suitability models and expanding the application to consider the middle ground of hybrid methodologies, the Project Management Institute (2017) proposed in the "Agile Practice Guide" a model to help organisations assess whether projects should be undertaken using predictive, hybrid or Agile approaches.

The model consists of the evaluation of organisational and project attributes under three categories:

- Culture – The objective is to check if a supportive environment with buy-in for an Agile approach is in place;
- Team – To verify if the team has a suitable size to be successful in using Agile and if its members have the necessary experience and access to the business representatives;
- Project – To evaluated rates of changes, the possibility of incremental delivery, and the project criticality.

2.5. Agile Methodologies

Griffiths (2015) states that the most common non-hybrid[15] Agile methodologies are: Scrum, Extreme Programming (XP), Lean Software Development (LSD), and Kanban[16]. All these methodologies share similar philosophies; however, from the point of implementation, each one has its principles, terminologies, and practices.

2.5.1. Scrum

2.5.1.1. Main concepts and practices

Sutherland (2015), Scrum's creator, presents his creation as a framework for implementing Agile methods. It is an iterative development model used to manage software and product development. Fixed-length iterations called Sprints, lasting two to four weeks long, allow the team to ship software on a regular cadence. At the end of each sprint, stakeholders and team members meet to plan the next steps.

Scrum follows a set of roles, responsibilities, and meetings that rarely change. During each sprint, the Scrum team uses visual resources, like task boards and burndown charts, to show progress and get incremental feedback. It is focused on productivity and relies on its principles, process, practices, and tools to ensure the product is built quickly and with minimum complexities. The iterative and incremental approach is the main principle of Scrum lifecycle, implemented through a specific set of steps (cf. Figure 9).

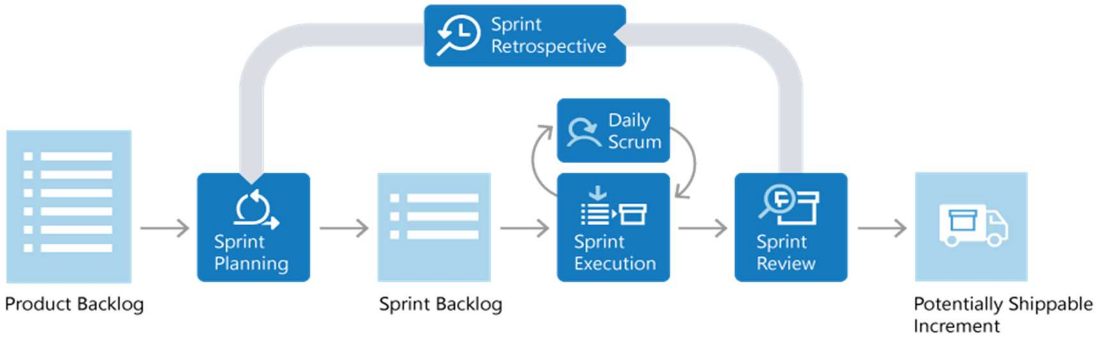


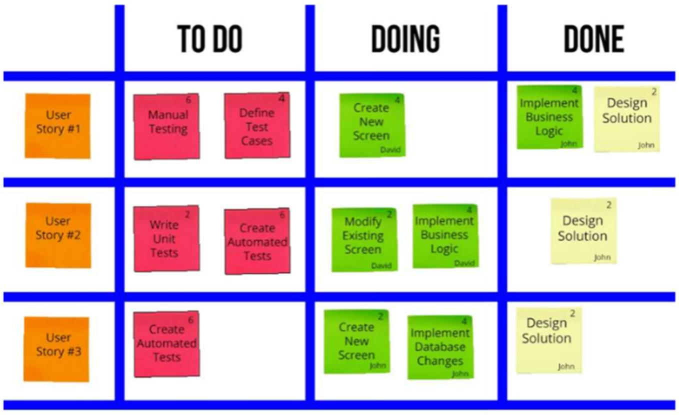
Figure 9 - The Scrum lifecycle. Source: Microsoft (Boer, 2017)

[15] A hybrid methodology is a combination of two or more different methodologies to create a new customized one. On the other hand, a non-hybrid methodology is a pure (or original) methodology (PMI, 2017).

[16] cf. PMI-ACP® Exam Prep: A Course in a Book for Passing the PMI Agile Certified Practitioner (PMI-ACP) Exam. Second Edition. Pages 41–59 (Griffiths, 2015).

The purpose of Scrum is to create usable, and potentially releasable increments that deliver value to customers (Sutherland, 2015). To do so, it makes use of practices and tools such as:

- Ceremonies – Scrum has four meetings known as ceremonies that bring structure to each sprint: sprint planning, daily stand-up, sprint demos, and sprint retrospective;
- Backlog Grooming – At the end of one sprint, the team meets to ensure the work is ready for the next sprint. The goal of the grooming is to ensure the backlog has just relevant items that meet the project's priorities;
- User Stories – A user story describes a software feature[17] from the customer's perspective. It includes the type of user, what they want, and why they want it. The development team uses these stories to create code that will meet the requirements in the stories;
- Timeboxing – This is a fixed period of time during which a team works towards completing a goal. Instead of allowing the team to work until the goal is reached, the team stops working when time ends;
- Scrum Board – The goal of a Scrum board is to visualize the flow of the work within a sprint. It is usually divided into three categories: to do, doing, and done (cf. Figure 10). The team daily updates the board throughout the entire sprint;



	TO DO	DOING	DONE
User Story #1	6 Manual Testing 4 Define Test Cases	4 Create New Screen David	4 Implement Business Logic John 2 Design Solution John
User Story #2	2 Write Unit Tests 6 Create Automated Tests	2 Modify Existing Screen David 4 Implement Business Logic David	2 Design Solution John
User Story #3	6 Create Automated Tests	2 Create New Screen John 4 Implement Database Changes John	2 Design Solution John

Figure 10 - Scrum Board example. Source: Agile Velocity (Hawks, 2010)

- Burndown Chart – A burndown chart represents all the project's work. The backlog is usually on the vertical axis, with time along the horizontal axis (cf. Figure 11). The remaining work can be measured in hours, story

[17] A software feature is a unit of functionality of a system that satisfies a requirement, represents a design decision, and provides a potential configuration option (Apel & Kästner, 2009).

points, ideal days, or other metrics. The chart shows the team how things are going on and supports the decision-making process.

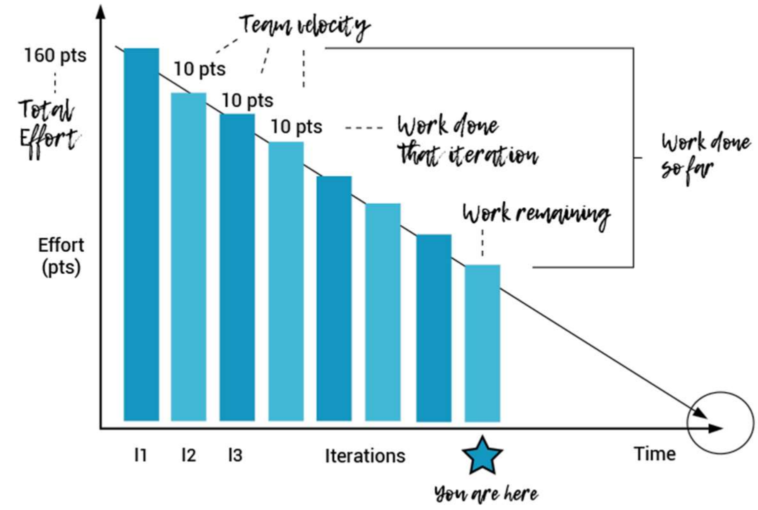


Figure 11 - Burndown Chart example. Source: Scaling Your Business (Gonçalves, 2019)

2.5.1.2. Previous research on Scrum

Due to the strong focus of Scrum on people, there is plenty of academic studies involving aspects related to Scrum teams. The researchers Hossain, Babar, and Paik (2009) studied the limitations of applying Scrum practices in a Global Software Development (GSD[18]) environment. The goal was to identify the factors that restrict the application of Scrum on projects in which the teams are globally distributed and explore strategies to deal with these challenges. The research concluded that the use of Scrum might be limited by a series of factors, such as:

- Lack of synchronous communication – Scrum planning and retrospective meetings are usually long events. Thus, it may be difficult to conduct effective sessions with distributed teams, especially when experiencing time zone differences;
- Cultural and linguistic diversity – These differences can inhibit the team members of raising their opinions and point of views, which may lead to misunderstandings among them;
- Slow and unreliable communication networks – Using different types of communication tools can lead to poor information transmission. Thus, providing good communication bandwidth and adequate tools is vital;

[18] A GSD is a software development effort undertaken dispersed along several software development centres, usually located in different countries or continents (Vanzin et al., 2005).

- Lack of collaborative tools – Not having collaborative tools like virtual task boards and shared issue trackers are challenging factors in GSD projects;
- Large teams in multiple sites – Managing large teams distributed at multiple sites is also considered a challenging undertaking.

Another Scrum study performed and published by the researchers Rising, and Janoff (2000) relates to the use of Scrum by small software development teams. The study analysed the experiences of three different teams (limited to 10 members) with Scrum in a company that fabricates communication products. The scenario was: Team A built a new platform simulator for the use of a type of telephone; Team B created a new product for a call centre market; Team C developed a new feature for a telephone switching system.

The results of the interaction of these three small teams using Scrum showed that throughout the project progress was made, even when requirements were not stable; problems were clearer and visible to everyone; team communication improved; a straighter collaboration established between the teams and the customers helped create a trusted environment. The researchers also concluded that Scrum is appropriate for projects where it is hard to define requirements up front and, that with the use of the methodology, difficult conditions can be anticipated.

2.5.2. Extreme Programming (XP)

2.5.2.1. Main concepts and practices

Extreme Programming (XP) is an Agile software development framework that aims to produce higher quality software and higher quality of life for the development team. XP is the most specific of the Agile methodologies regarding software engineering practices (Beck, 2004).

Kent Beck, Extreme Programming’s creator, designed the framework to deliver the largest amount of valued code within the shortest time. Therefore, XP developers work within strict timeframes. Extreme Programming has the highest degree of customer involvement among Agile methods. The customer usually participates at all stages of work in person. He is not just a stakeholder, but part of the team. XP also has the shortest iterative cycle (most common is one week) compared to other Agile methodologies. Like in Scrum, the Extreme Programming lifecycle is iterative and incremental and repeats several times until the final version of the software (cf. Figure 12).

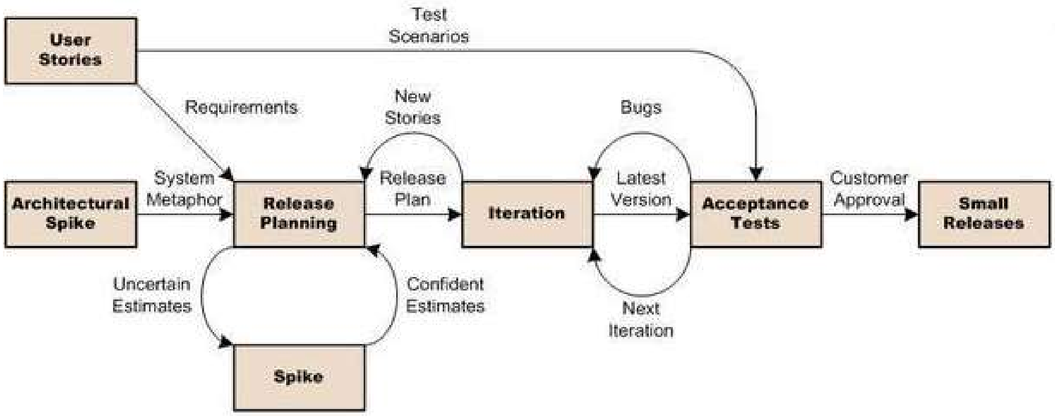


Figure 12 - The Extreme Programming lifecycle. Source: ANTOSSOFTWARETOPICS (S., 2016)

The core of XP is the interrelated set of its 12 practices. Beck (2004) claims that, while it is possible to do these practices in isolation, XP teams believe each practice reinforces the others, and all should be done in conjunction to minimize risks often faced in software development. These practices are best described as:

1) Planning Game – This practice creates an emotional distance from planning by treating it as a game with pieces, players, goals, and rules;
2) Small Releases – XP works with short cycles that provide code for a small set of functions (the User Stories selected by the team for the iteration);
3) Metaphor – XP uses metaphors to explain the design and build a shared view. Metaphors create perspectives that stakeholders understand and help explain the product. The goal is to speed up the understanding;
4) Refactoring – This is the process of clarifying and simplifying the design of existing code, without changing its behaviour. XP teams extend code from iteration to iteration and refactoring helps to keep the code clean;
5) Test-Driven Development (TDD) – TDD process relies on the repetition of a short development cycle. Requirements turn into test cases, and then the code is written to pass the tests. It is opposed to the techniques that allow adding code that does not meet the requirements;
6) Simple Design – XP believes that what does not bring business value is considered waste. Thus, the team is encouraged to design the ideal solution as simple as possible;
7) Pair Programming – A technique in which two developers work together at one workstation. One writes code while the other reviews each line of code as it is typed in. The two developers switch roles frequently;

8) Collective Ownership – XP considers code to belong to the project, not to an individual. As the developers build functionalities, they may modify any part of the code;
9) Continuous Integration – At regular intervals, continuous integration yields releases that grow in functionality. It is how the project leader measures progress, identify and mitigate risks on an ongoing basis;
10) 40-hour week – A practice and metaphor for the team to achieve a sustainable pace by working no more than eight hours per day (or forty hours per week). Productivity does not increase with hours worked, because tired people are less productive;
11) Whole Team – This practice suggests that people who work together in interlinking ways make the project more effective;
12) Coding Standard – XP supports that defining standards for coding leverages consistency. It permits the developers to read code easily and make parts of the system refer to other parts more fluently.

2.5.2.2. Previous research on Extreme Programming

Layman, Williams, and Cunningham (2004) conducted a research aiming to evaluate the consequences of adopting the Extreme Programming methodology. Performed at Sabre Airline Solutions (the largest systems provider for air bookings in North America), they compared two releases of the same system: the 3rd release (called old release) and the 9th release (the new release). The team used a traditional software development methodology in the old release, which started in early 2001 and took 18 months to complete. In this period, the team became experienced in Extreme Programming and tailored its practices to be compatible with their environment. The development of the new release began in late 2003 and took three and a half months. Some of the facts and differences between the two releases were:

- The number of stakeholders more than doubled between releases, which led to the generation of more requirements;
- In the old release, the team sat in semi-private zones, but in the new release the team occupied an open office environment, and the developers used pair-programming stations;

- In the new release the team worked under pressure to add features into the system as the release deadline approached, which may have led to lower code quality;
- A non-mandatory code review policy was in place in the old release. Pair programming took the place of code reviews in the new release;
- Defect prevention also changed between releases, with customer tests being integrated into the new release.

The findings of the researchers suggest that adopting XP methodology may improve developer productivity and product quality. However, it is important to note that the results of the study are limited by the particular conditions in which it was performed. Table 5 presents a summary of the research findings.

Table 5 - Extreme Programming case study findings

#	**Hypothesis -** The use of a set of XP practices led to improvements in...	**Case study evidence?**
1	Internal code structure	No – increase in overall design complexity
2	Pre-release quality	Yes – 65% reduction in defects found in tests
3	Post-release quality	Yes – 30% fewer problems were reported
4	Programmer productivity	Yes – 50% increase in coding output
5	Customer satisfaction	Not evaluated
6	Team morale	Not evaluated

2.5.3. Lean Software Development (LSD)

2.5.3.1. Main concepts and practices

Lean Software Development was inspired by a manufacturing process named Lean[19], whose inventors considered that defects, waiting, processing, inventory, overproduction, transportation, and unneeded movement were the primary sources of problems with product delivery. For this reason, waste elimination is the central concept of Lean (Wilson, 2009).

In 2003, the researchers Mary and Tom Poppendieck merged Lean manufacturing and IT principles. By adapting the core principles of Lean to the software industry, they created the Lean Software Development (LSD)

[19] Lean was invented in the middle of the 20th century in Japan by Toyota employees. Initially named as Toyota Production System (TPS), it was renamed later to Lean Manufacturing (Wilson, 2009).

framework. The results of this work were published in their book "Lean Software Development: An Agile Toolkit". Although LSD is not the most popular Agile methodology, it is among the most used ones due to the benefits of waste elimination and workflow optimization (Griffiths, 2015).

Lean Software Development implements a lot of Agile practices. Griffiths (2015) highlights the importance of tailoring such practices to the reality of each project so that through adaptation they can be more effective. He describes these practices as:

- Automation – Cuts cycle times by doing things faster than doing manually;
- Visual Controls – A system of signs and data displayed to a faster visualization of the flow of work and instantly identification of problems (e.g., Kanban board);
- Virtual Kanban Systems – An electronic system that keeps track of the work and provides information when new work can start;
- Kaizen Events – Short term projects with a specific goal of improvement. They are week-long events led by a facilitator with the project team;
- Daily stand-up meetings – Short meetings held every day, no longer than 15 minutes, whose goal is not meant to solve problems, but to make participants aware of the status. If further discussion is needed, a longer meeting with appropriate parties may be arranged;
- Retrospectives – A retrospective is a meeting conducted at the end of an iteration. The intent is that the team reflects on what happened in the iteration and identifies areas for improvement;
- Cumulative Flow Diagram – A CFD is a chart (cf. Figure 13) used to track the progress of work in certain states (or phases) over time.

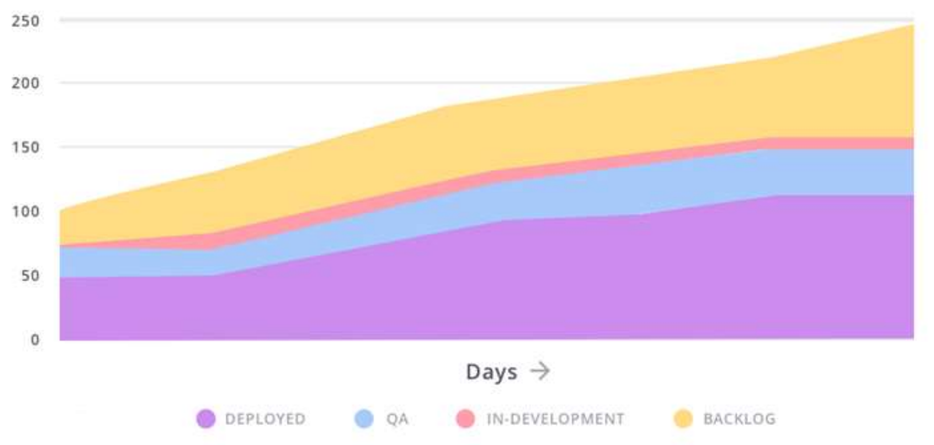


Figure 13 - Cumulative Flow Diagram example. Source: Zepel (Jaikrishnan, 2019)

2.5.3.2. Previous research on Lean Software Development

In 2011, Middleton and Joyce published a study about how Lean principles could be used in software development projects. The hypothesis evaluated was that applying Lean ideas might improve the capability of the development processes. The study began in 2009 in London, UK, and evaluated the performance of a small software development team (nine members only) from BBC Worldwide, a subsidiary of BBC Studios, while using Lean practices and tools with a variety of key users and stakeholders. The hypothesis would be verified by the reduction of lead time, defect rates, and variability, combined with evidence of continuous improvement.

The results showed that, within the one-year research, the lead time decreased by 37%, and the consistency of delivery improved by 47%. Errors reported by users dropped 24%. The use of visual management tools, team-based problem-solving approach, small batches, and process control improved the overall results when compared with traditional approaches previously used by the team. The hypothesis was supported by the study, considering all the quantitative and qualitative data collected.

The researchers concluded that the performance of the development team improved by adopting Lean practices. A faster delivery focused on creating value to the customer also reduced technical and market risks. The study concluded as well that the methodology might not work properly over existing organisational standards due to the necessity of customizations.

2.5.4. Kanban

2.5.4.1. Main concepts and practices

Anderson (2010) describes Kanban as a visual framework that shows what to produce, when to produce, and how much to produce. He explains that Kanban was inspired by the Toyota Production System (TPS) when, in the 1940s, Toyota improved its engineering process by modelling it like supermarkets inventory management processes, offering just enough products to meet demand. By optimizing the flow between the market and the customer, Kanban improves efficiency in the production line.

Toyota brought these principles to its factory floors. The team uses a card (Kanban in Japanese) to show available capacity and pulls the requests from the backlog. Because of this, Kanban is also known by “pull system”. The same idea applies to software development. Work-in-progress (WIP) takes the place of inventory, and new work is added only when there is "space" on the board since Kanban matches the WIP to the team's capacity. According to Anderson (2010), stay on the path of continuous delivery is the main objective of Kanban when applied to software development.

Figure 14 shows an example of a Kanban board with seven phases. The board is customized according to the needs of the team and shows all the stages of the process. The work moves from left to right on the board. The Kanban board should be placed near the project team so that each team member can see it and update the status of the work as per their needs.

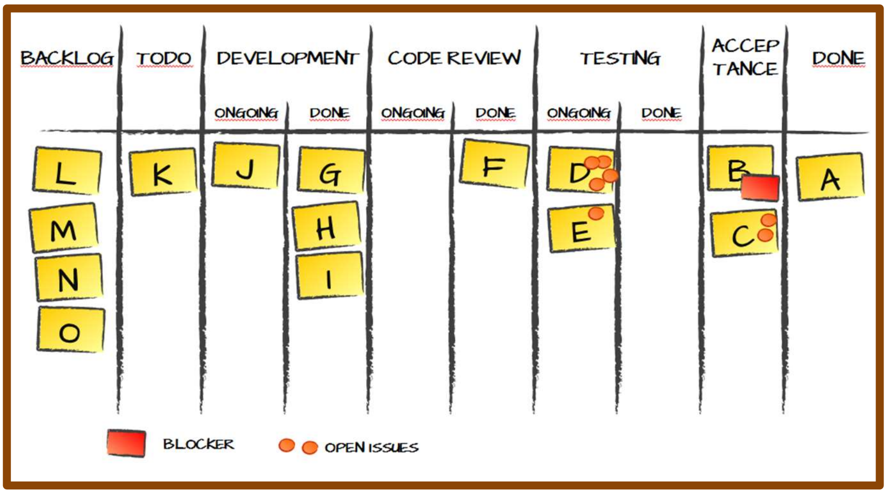


Figure 14 - Kanban Board example. Source: Software Management Project (Brodzinski, 2015)

Anderson (2010) elected five practices as the most consistently used in successful implementations of Kanban. These practices are:

1) Visualize the workflow – A visual representation of the workflow promotes better comprehension of the current situation. By making the work visible (blockers and open issues included), it is easier to anticipate problems and improve collaboration;
2) Limit work-in-progress (WIP) – WIP limits determine the minimum and maximum amount of work for each step on the board or the entire workflow. By limiting the WIP is possible to increase focus and speed;
3) Manage / enhance the flow – The flow of work throughout the Kanban board is monitored and improved. It is desirable a constant flow, showing that the team is frequently creating value. The team then analyse problems in the flow and implement changes;

4) Make process policies explicit – For the collaborative change to occur in the Kanban system, the processes need to be explicit. The team is encouraged to modify the board as needed to make the process clear;
5) Continuously improve – Once the Kanban system is in place, the team can identify issues and suggest improvements. Teams measure their effectiveness by tracking the flow, measuring cycle time, and increasing the quality of work.

2.5.4.2. Previous research on Kanban

Ahmad, Markkula, and Ovio (2013) conducted a comprehensive literature review of the trend of Kanban utilization in software development, aiming to identify its benefits and challenges. In spite of an apparent lack of scientific research involving Kanban usage in this area in 2013, the researchers were able to select 19 relevant papers (among 492) to analyse in more detail. The results of the study provided significant insights for future studies on Kanban in software engineering.

As major benefits of using Kanban, the researchers identified improved lead time to deliver software, improved quality of software, improved communication and coordination, increased consistency of delivery, and decreased "bugs". The challenges are related to organisational issues and the lack of knowledge and training. In addition, some best practices were identified and listed to provide a guide for teams interested in using Kanban.

The findings of the study were described as:

- Simplicity, focus on the flow of the work, and no mandatory use of fixed iterations are key factors for Kanban adoption;
- Customer satisfaction, ideal lead-time delivery, improved quality of the software, early feedback, low number of errors, improved communication, and increased team motivation are the main benefits of using Kanban;
- Kanban works well when blended with other Agile practices, even though other papers have concluded that it is not easy to work with hybrid methods;
- Training people on Kanban and changing organisational culture are the great challenges to introduce the methodology;

- A good strategy to adopt Kanban recommends incremental transitions (beginning by the Kanban principles) and adequate team mentoring. Daily meetings are considered a key practice for the flow of information.

Despite the relevant findings, the study indicated that there was room for more rigorous and extensive scientific research on Kanban in software development.

2.5.5. A summary of the four Agile methodologies

Regarding the main aspects of each methodology studied, we can affirm that:

- Scrum is mainly a managerial framework. It is about the activities the team do when they are not coding. Scrum explicitly prescribes a model, according to which the team plan the work and analyse how things went daily and during the retrospectives. Scrum has specific roles that facilitate the development process and ensure it is followed;
- Extreme Programming (XP), on the other hand, focus on software engineering practices. Higher quality of software and quality of life for the development team are the priorities. Being the most specific of the Agile methodologies regarding engineering practices, Extreme Programming may require some tailoring to suit when it comes to the use of its full set of practices;
- One of the key ideas of Kanban is to refrain from producing a surplus. Kanban achieves this by using visual tools to capture and show how features move through the development cycle. It gives the project team insight into the development process and helps managers address issues in real-time;
- Lean Software Development strives to reduce waste and maximize value to the customer. Waste could allow the building of the wrong feature, something that would never be used. Lean relies on the idea that the leadership should trust the project team is making its best effort. Like in Extreme Programming, Lean Software Development has prescriptions regarding engineering practices.

Having different methodologies that address distinct "problems", while remaining loyal to the same values and principles, is one of the greatest advantages of Agile over other software development approaches since it brings more flexibility to the organisations (Griffiths, 2015).

We elaborated a comparative view summarizing the main information of each Agile methodology studied (cf. Table 6).

Table 6 - Comparative summary of the four Agile methodologies studied

	Scrum	**XP**	**LSD**	**Kanban**
Main concept	Productivity	Engineering	Waste elimination	Production flow
Principles	Transparency, Inspection, and Adaptation	Simplicity, Communication, Feedback, Courage, and Respect	Eliminate waste, Amplify learning, Empower the team, Deliver fast, Defer decisions	Incremental change, Respect current processes, Encourage leadership acts
Development approach	Iterative & incremental	Iterative & incremental	Iterative	Incremental (Iterations are optional)
Suggested iteration time	Two to four weeks	One to three weeks	None	None
Main practices	Daily stand-ups, Demos, Retrospective, Backlog Grooming, Timeboxing	Planning Game, Pair Programming, Test-driven development, Refactoring	Automation, Kaizen Events, Daily stand-up meetings, Retrospectives	Visualize the workflow, Limit WIP, Make process policies explicit, Continuously improve
Main tools	Scrum Board, Burndown Chart	Task Board, User Stories	Visual Controls, Cumulative Flow Diagrams	Kanban Board

Chapter 3 – Research Methodology

This research encompasses two complementary parts. The first part focused on collecting and analysing a great deal of valuable information on Agile, aiming to elaborate a conceptual model of the decision process for an Agile approach. For this, different techniques were applied, including a comprehensive literature revision complemented with an exploratory study, integrating an interview and focus group sessions. This qualitative phase allowed us to understand the problem itself, verify the difficulties to be solved, and gather the requirements to sketch a conceptual decision model that represents the computer system to be proposed and developed.

In this first part, two research techniques were applied: an analytical research, since in this type of investigation the researcher uses facts and available information on the researched subject as a source for the accomplishment of critical analyses; and a theoretical-conceptual research, which is related to the creation of a new concept or model. Combined, these methods compose a conceptual-analytical approach (Kothari, 2008).

The second part of this research aimed to build a prototype of the computer system that implements the conceptual decision model previously elaborated. Practices derived from Agile methodologies were used to design and develop the prototype of this system. These practices were chosen according to the specific needs of this research. Since it is a scientific work, the level of documentation required is higher than what is usually adopted in a professional project that applies an Agile methodology, and more formal as well. In this way, some adaptations in documenting these practices were necessary, but without compromising the final result.

The research methodology applied in this second part of the work is classified as artefact-building, since in the construction of a new system (artefact), the aspects relating to its usefulness are emphasized and a methodology directed to the construction of the product is used (Järvinen, 2000).

3.1. Part 1 – Literature Revision & Exploratory Study

In the literature revision, characteristics, concepts, best practices, main applications, success cases, previous studies, and selection techniques on

Agile methodologies were analysed. Books whose authors are Agile researchers, specialists, and practitioners provided valuable data. Also, conference proceedings, research results, articles in periodicals, specialized journals, and classroom materials, having Agile as the subject, provided meaningful information. Some examples of this literature are: Waterfall Vs V-Model Vs Agile: A Comparative Study on SDLC; Kanban in software development: A systematic literature review; Exploring Extreme Programming in Context: An Industrial Case Study; Using Scrum in Global Software Development: A Systematic Literature Review; Applying Lean Principles in Software Development Process – A Case Study.

In complement, an exploratory study integrating one interview and three focus group sessions were carried out. An exploratory study may involve gathering data through interviews with subject matter experts and debates with groups of people who have experience with the focused subject. The main goal is to capture an overview of the "problem" that, usually, have limited or no previous study, aiming to make discoveries (Kothari, 2008).

The interview and the focus groups were held to collect the opinions of the participants – professionals familiar with Agile methodologies and software development processes – on the aspects that may support the choice of an Agile approach in software development. Additionally, in all of these activities, the participants were asked to provide their views about the essential requirements for the system to be (conceptualized and) developed, which provided a core vision of the high-level requirements for the system.

For a qualitative analysis of the data obtained, we used Leximancer[20], a software that analyses document contents (textual data), allows context analysis, and visually displays the relationship among the extracted information (summarized in themes and concepts).

As a result of the first part of the research, the high-level requirements for the system to be built were documented, and a conceptual model was elaborated to support the decision process and the choice of an Agile methodology in software development projects. Data of past software development projects were used to validate the conceptual model (and confirm the consistency of the resulting recommendations).

[20] Available in *https://www.leximancer.com*

Figure 15 depicts the work performed and the outputs (in blue) of the first part of the research.

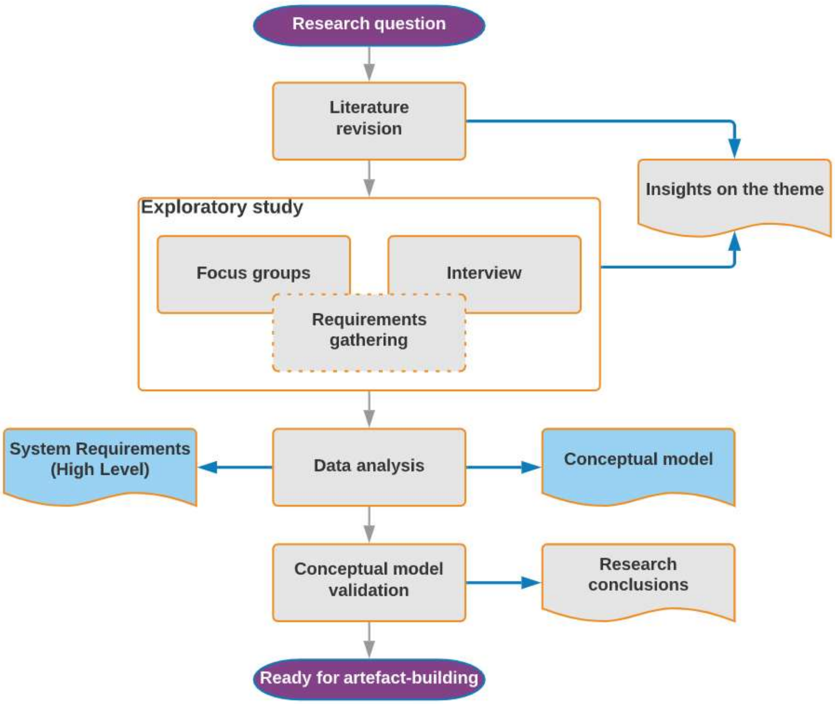


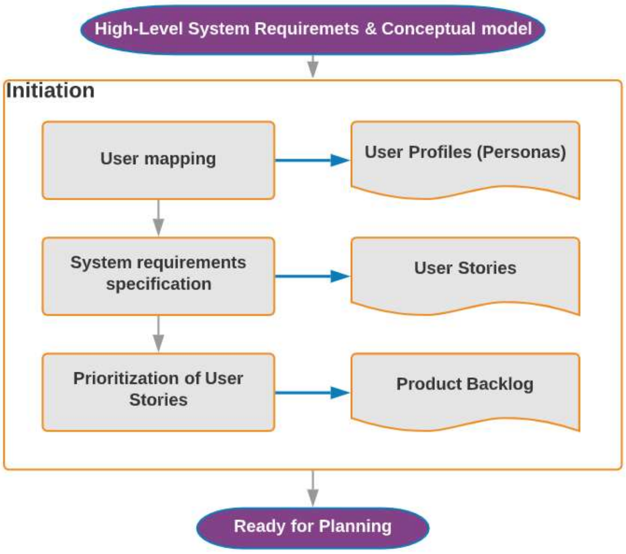
Figure 15 - Literature Revision & Exploratory Study

3.2. Part 2 – Prototype Development

The second part of our research (the prototype development) is comprised of three phases: i. Initiation; ii. Planning; and iii. Execution. These phases are detailed as follow.

3.2.1. Initiation phase

In the Initiation phase (cf. Figure 16), the potential users for the system are identified and described, focusing on the benefits that they might obtain from it. An Agile technique called Personas was used. Also in this phase, the high-level system requirements gathered in the first part of the study are detailed and described as User Stories. As the last activity of the phase, a Product Backlog containing a prioritised list of the User Stories was created. It shows the "just enough features" to be available in the first version of the system.

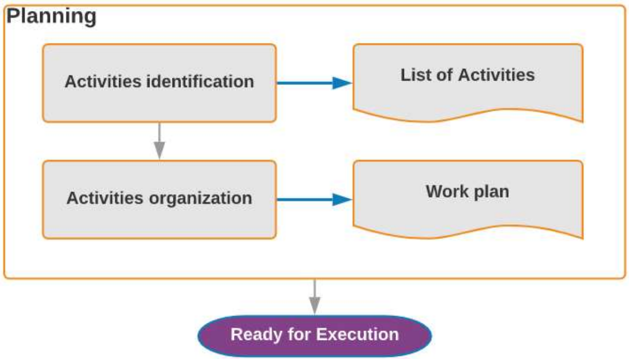

High-Level System Requiremets & Conceptual model
Initiation
User mapping
User Profiles (Personas)
System requirements specification
User Stories
Prioritization of User Stories
Product Backlog
Ready for Planning

Goals: Identify the system users; describe and prioritise the system requirements to be implemented.

Figure 16 - Prototype Development - Initiation phase

3.2.2. Planning phase

In Planning (cf. Figure 17), the activities necessary to design the system and build the prototype are identified and organised in a way that shows what needs to be done, in which sequence. To organise these activities and allow a proper management of its execution, an electronic Kanban board was created using Trello[21,], a task management online software application.



Goals: Identify the activities necessary to transform the system requirements into system functionalities; elaborate a work plan for the prototype development.

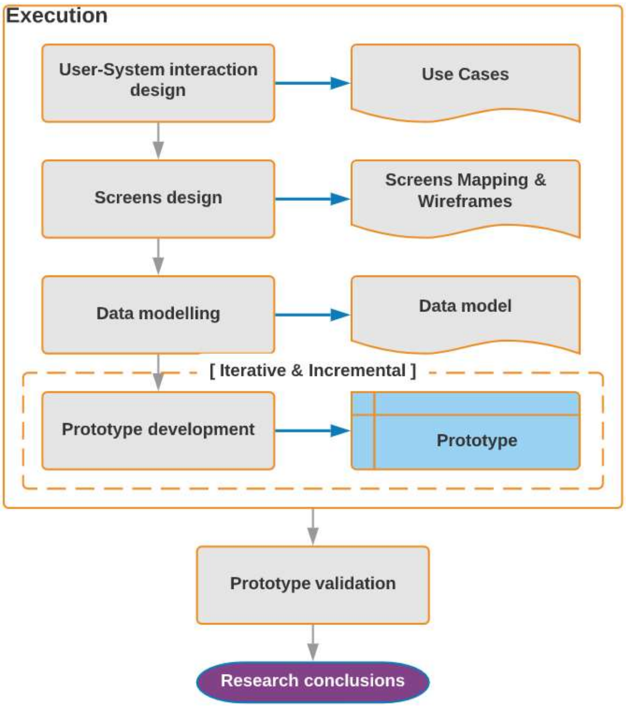
Figure 17 - Prototype Development - Planning phase

3.2.3. Execution phase

In Execution (cf. Figure 18), the system features are designed and modelled using free web-based design tools (Lucidchart[22], MockFlow[23], and Draw.io[24]). In the prototype development, the features were built in an

[21] Available in *https://trello.com*
[22] Available in *https://www.lucidchart.com*
[23] Available in *https://mockflow.com*
[24] Available in *https://www.draw.io*

iterative and incremental way, meaning that each subsequent iteration adds new features until the prototype contains just enough resources for its basic operation. The last activity of the phase was the validation of the prototype by key stakeholders.



<u>Goals</u>: Design and model the system; build the prototype (in an iterative and incremental way); validate the prototype.

Figure 18 - Prototype Development - Execution phase

Chapter 4 – Exploratory Study

4.1. Interview and Focus Groups

The specific objectives of the interview with the Agile specialist and the focus groups with Agile developers and practitioners were:

1st – Identify the factors that may support the choice of an Agile approach in software development;

2nd – Identify the aspects that may help in pointing out the best Agile methodology to be used among Scrum, Extreme Programming, Kanban, or Lean Software Development, in case Agile is a suitable approach;

3rd – Capture the essential functional and practical requirements for the information system proposed.

The interview and focus groups were held in different cultural and organisational environments and involved people with a diverse level of knowledge and experience with Agile methodologies. The goal was to capture various perceptions of the Agile aspects under investigation.

The script with the questions (cf. Appendix A – Focus Group Script) elaborated for the focus groups was created after the analysis of relevant information, other studies, and previous work about Agile (e.g., Agile Software Development, Principles, Patterns, and Practices; Agile Practice Guide; Scrum: The Art of Doing Twice the Work in Half the Time; Extreme Programming Explained: Embrace Change; Kanban: Successful Evolutionary Change for Your Technology Business; Lean Software Development: An Agile Toolkit). These questions aimed to stimulate the debate among participants focusing on:

- The conditions that favour the use of an Agile approach;
- The characteristics and behaviours that leaders, teams, and members of development teams should have to work with an Agile methodology;
- The adequate environmental factors (physical or cultural) for a software development project to use an Agile methodology;
- The characteristics that the product of the project (the software) may have that contribute to the use of an Agile methodology.

The script with the questions elaborated for the interview with the Agile specialist also covered the mentioned scope, but it contains an additional set of questions to explore the knowledge of the interviewee on the factors that may influence the choice of the Agile methodology, among the four studied.

In this way, only the aspects that might be addressed differently by each methodology were explored in the interview since those that are similar do not help in differentiating the methodologies. These questions are a bit more elaborated and can be found in detail in Appendix B. The questions aimed to collect the impressions and opinions of the interviewee on

How each methodology addresses…:

- … changes in requirements and in what way each type of approach influences development cycles (iterations);
- … the use of software engineering practices and whether there is anyone that emphasizes this aspect;
- … customer engagement and the need for higher interaction and collaboration between the project team and business representatives;
- … the delivery approach in the sense of continuous delivery, deliver as fast as possible, and minimum viable product (MVP[25]);
- … the prioritisation of system requirements.

The following sections provide more details about the interviewee and participants of the focus groups, as well as on the key insights and findings. The detailed transcripts of the interview and the focus groups can be found from Appendices C to I.

4.1.1. Interview with an Agile specialist

The interview was conducted with the Agile specialist Tiago Palhoto. Tiago is 20 years experienced in software development. He is an Agile coach and author of book and articles about Agile. He currently works as a consultant for the European Commission (EC) in Brussels, Belgium, and is responsible for scaling Agile practices in the EC.

4.1.2. Focus groups

Three focus groups were held in different cultural and organisational environments whose participants have diverse degrees of experience in Agile. The objective was to capture various perceptions of the theme. The information concerning origin, composition, and background of each group is

[25] A Minimum Viable Product (MVP) is a product that has just those features (and no more) that allow the user to use the product (Ries, 2011).

presented next; however, participants' names and the identification of the organisation they belong are omitted due to non-disclosure agreements:

- The first focus group gathered four middle management professionals trained in Agile from a Humanitarian organisation with an office in Brussels, Belgium. They have been working on the implementation of Agile practices in some of its projects;
- The second focus group brought together six IT project managers from a Portuguese organisation of the Banking sector. They are experienced in Agile (on average two and a half years). The organisation uses Agile extensively in systems development projects;
- The third focus group was attended by five software developers from a Brazilian company of the Telecommunications sector, with strong experience in Agile (on average five years). The organisation has been extensively using Agile methodologies in IT projects for many years.

4.1.3. Data analysis

We analysed the data gathered in the interview and focus groups with the help of the software Leximancer, which allowed a qualitative analysis of these non-structured data, highlighting meaningful concepts and the relationship among them. From the analysis of these concepts, it was possible to identify i. the factors that may support the choice of an Agile approach in software development; ii. the characteristics of software development projects and organisational environments (aspects) that help to indicate the best Agile methodology to use (among the four studied); iii. the essential high-level requirements for the system proposed.

In Leximancer, we generated concept maps[26], linking the related concepts, and also ranked concept lists, containing the most used names, words, and the relevance of them. The concept maps are presented next (cf. Figures 19, 20, and 21), followed by the main insights obtained from them. The ranked concept lists can be found in Appendices J, K, and L.

4.1.3.1. Factors that may support the choice of an Agile approach in software development

[26] A Concept Map is a visual display of concepts and their relationships to each other (Leximancer Pty Ltd, 2018).

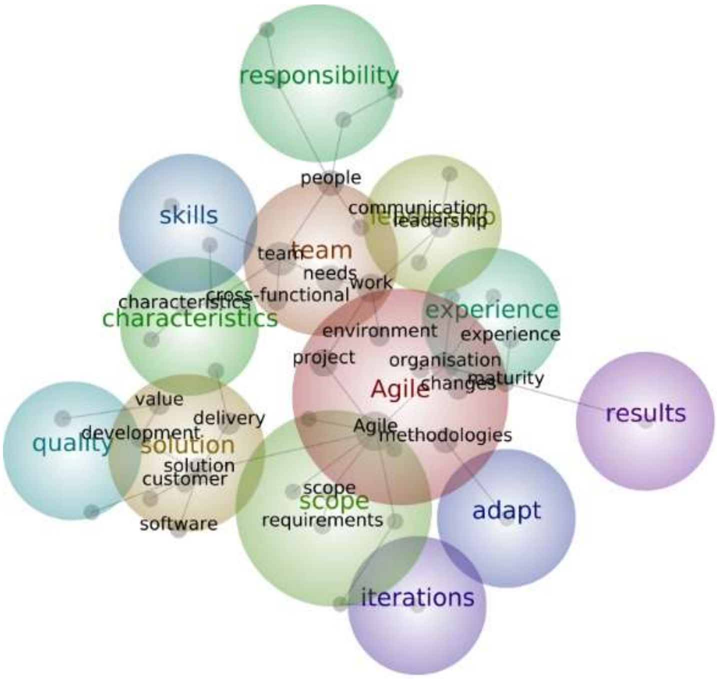


Figure 19 - Map of the concepts related to the choice of an Agile approach

This map shows the conceptual structure of the data collected in the exploratory study (regarding the choice of an Agile approach). The interactive nature of the map allowed us to explore the concepts and their connections to each other. This resource permitted us to see how the different concepts are connected, facilitating our analysis processes. The main insights obtained are described next (in topics), organised by aspect evaluated.

1st Aspect: Conditions that favour the use of an Agile approach.

- When the maturity of the organisation in Agile is high. If the maturity level is low, it is key to have a group of people who already know Agile, ready and willing to take the first step, and empowered to play a leading role in the consolidation of the Agile principles;
- When the degree of uncertainty on the system requirements is high;
- When the organisation has a project-based structure and less bureaucratic barriers;
- When is possible for the project teams to work closely (co-located);
- When the organisation has the ability to implement changes quickly and effectively (change is managed at the organisational level);
- When the scope of the software development project changes frequently (a high volume of change requests);

- When there is a necessity of generating continuous value through constant delivery of software increments. The team must be able to focus on delivering software frequently;
- When there is room in the organisation or in the project for decision-making to happen together.

2nd Aspect: Characteristics and behaviours that leaders should have to work with Agile.

- Pilot the Agile adoption and utilization, setting guidelines and the vision, supporting all changes related to these processes;
- Delegate and empower the team so that they can make their own decisions, letting them stand out and acting more as a coach and facilitator, trusting they are the best people to get the job done;
- Move from a command and control approach to trusting the people who are really going to do the work;
- Tolerance, knowing that Agile means experimentation and will lead to mistakes. Understand that these mistakes are healthy and should occur as soon as possible so the team can anticipate issues and act on them;
- Create and ensure that working conditions are created so that people are proud and enjoy what they do;
- Promote cross-functional (anti-silos) multidisciplinary teams, and a product-oriented approach more than a project-oriented one;
- Know very well your team and stay close to it to be familiar with important themes and act to unblock issues as quickly as possible;
- Be versatile at the decision-making level, being able to justify and support the reasons for changes and transitions;
- Communicate clearly;
- Shield the team from external interference as much as possible;
- Be willing to face frequent changes in the course of the project, knowing that more changes will come, and then dealing with them naturally;
- Promote a continuous improvement environment, often questioning what can be done to extend the benefits already obtained.

3rd Aspect: Characteristics and behaviours a development team and the members of a development team should have to work with Agile.

The team:

- Should be small, no more than 12 people, and co-located;
- Must have a good degree of autonomy and be self-organised;
- Must be cross-functional and present a multiplicity of skills;
- Should be experienced in Agile and constantly seek for self-improvement;
- Should seek to know what other Agile teams, inside or outside the organisation, have been doing;
- Should have low turn-over, avoiding the team's disruption;
- Should seek the alignment of expectations and values within the team (working agreements);
- Must show commitment, engagement, and focus on the project activities, minimizing context switch.

The team member:

- Is accountable, responsible, transparent, and willing to share information;
- Has team spirit, has a sense of unity, seeks cohesion, has good behaviour;
- Feels free to raise problems and risks without fear;
- Has an open mindset and quickly adapt to new work environments.

4th Aspect: Environmental factors for a software development project to use Agile.

The physical environmental factors suggest:

- The possibility of collocating the project team and business representatives. If it is not possible, ensure to have the tools to enhance communication and collaboration as much as possible;
- To have open workspaces that promote proximity among people, allowing them to interact more and solve problems faster.

The cultural environmental factors suggest:

- A more horizontal and less vertical organisational structure, with a lower level of hierarchy;
- A safe environment where people feel safe to express themselves;
- Maturity and openness to embracing organisational change processes;
- An environment free of a command-and-control culture;
- A product-oriented environment, with teams linked to a product that will evolve over time;
- An environment where accountability is shared by everyone.

5th Aspect: Characteristics a product may have that favour Agile.

- A perspective of development and maintenance of an evolving product able to be built incrementally and iteratively (delivered in chunks);
- Minimum viable product-oriented, having just enough features to work;
- A short time-to-market product;
- A software that does not require a lot of documentation, either resulting from the development process or for proper operation;
- A simple and easy-to-use solution.

4.1.3.2. Aspects that help to indicate the best Agile methodology to use

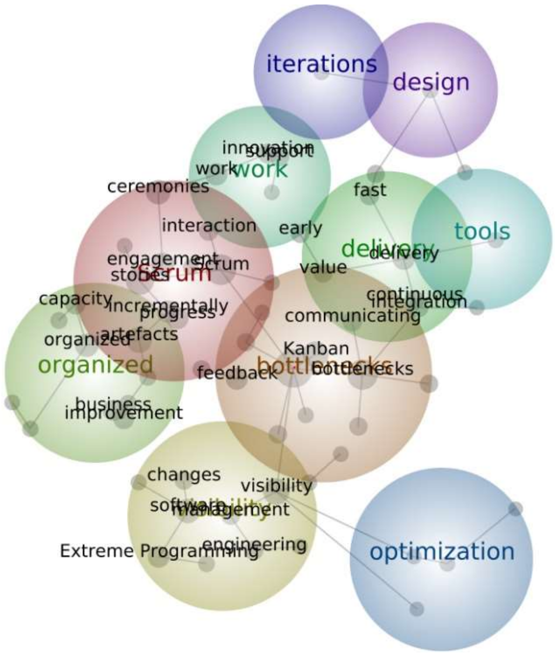


Figure 20 - Map of the concepts related to choosing an Agile methodology

Next, we describe the main insights obtained about how each Agile methodology studied addresses the aspects that are approached in different ways by each one. First, we identify the aspect, and then we present (in topics) the insights.

1st Aspect: How each methodology addresses changes in requirements.

- Scrum is the one that best enables the principles related to inspections, adaptation, and flexibility through short cycles. It is the one that effectively allows dealing with changes in system requirements;
- Extreme Programming is not that related to changes in system requirements. It is more related to the optimization of the team's work (development) regardless of the level of change requirements;

- Kanban and Lean Software Development are also not too related to changes in system requirements. They are focused on reducing waste, avoiding jumping tasks.

2nd Aspect: How each methodology addresses the use of software engineering practices.

- Extreme Programming was created to optimize and improve software code. It is the one that gives more emphasis to engineering practices and is not that related to management aspects;
- Lean Software Development and Kanban are close. LSD is focused on the optimization of the process, which usually can be seen through a Kanban board. They cover a bit of management, but also a lot of software engineering practices;
- Scrum does not focus on such practices. It is more comprehensive.

3rd Aspect: How each methodology addresses customer engagement in the project.

- Extreme Programming promotes this engagement with the customer by having someone fulfilling this “role” as a member of the project team;
- Scrum has a large set of artefacts and ceremonies that promote greater interaction with the client. It also clearly shows the moments this should occur;
- Lean Software Development and Kanban promote customer involvement while identifying the value streams, understanding the process, and how it is possible to optimize them.

4th Aspect: How each methodology addresses the concepts of continuous delivery, deliver as fast as possible, and minimum viable product (MVP).

- Scrum promotes continuity by delivering and showing product evolution;
- Lean Software Development does not since it is more focused on optimizing processes;
- Extreme Programming also promotes continuity by the way it approaches tests and constant integration of code, key factors in continuous delivery;
- As for "Delivery as fast as possible", Lean Software Development and Kanban are key tools to implement it since they have a set of recommendations and practices to optimize the delivery process, aiming at delivery as fast as possible;

- Scrum does not approach this directly since the time-boxed iterations do not mean the team has to hand over something at the end of the sprint;
- Extreme Programming also contribute to faster delivery, although both Lean and Kanban are the ones that can effectively do that;
- About a minimum viable product (MVP), Scrum is a perspective of that. The iterative approach allows the team to early start delivering value. Scrum formalizes the MVP way of thinking;
- Kanban also promotes this because of its focus on an improved workflow, which contributes to define and deliver a minimum viable product;
- As for Extreme Programming and Lean Software Development, they are more practical tools that can help to implement a minimum viable product.

5th Aspect: How each methodology works to prioritise features and activities.

- Scrum instantiates this principle very well by the way it handles and manages the backlog (which is native). The role of the Product Owner makes this clear in Scrum;
- Kanban also works this way. It is optional to prioritise the job, but the concern is pulling work and (often) putting them in order;
- Lean Software Development can give some indications regarding not the prioritisation of system requirements, but in the way the changes in the requirements are handled.

4.1.3.3. Essential (high-level) requirements for the system proposed

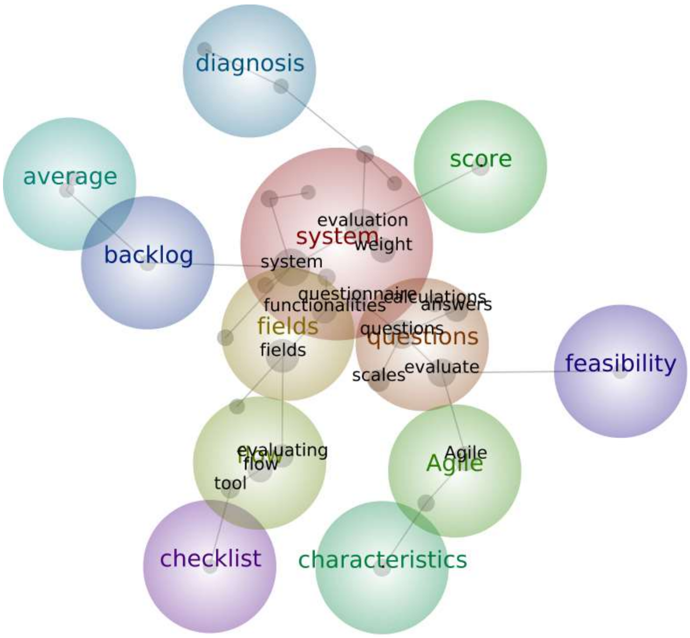


Figure 21 - Map of concepts related to the essential requirements for the system proposed

Unique aspect: The essential requirements for the system to be developed.

- Essentially an assessment-based solution. A questionnaire with questions and answers to capture and process the data required for the Agile suitability assessment and to provide a recommendation in the end;
- The assessment should be divided into two parts: the first evaluating the Agile approach suitability, and the second assessing and indicating the best methodology to use;
- The questionnaire should have a group of questions with specific weights, balancing the contribution of each aspect evaluated within the Agile or methodology suitability assessment;
- The questionnaire should support different types of answers, e.g., binary responses types (Yes or No), rating scaled ones (1 to 5), etc.;
- The assessment could support the view from different types of perspectives: from the software development perspective (the performing layer), and the business or leadership perspective (the directive layer), having transversal questions;
- If the idea is to have something absolute, the system could point out Agile or not, but if the idea is to have something subjective (for further evaluation), the system could provide a confidence interval as a result (e.g., 70% pointing to Agile and 30% to a traditional approach);
- Regarding the Agile methodology assessment, the questions could add in favour of one methodology or another and not point out exclusively a single option. The answer should lead to a result more prone to a good probability rather than a complete certainty;
- The rules to decide which methodology should be used might consider a non-exclusive approach, pointing out more than one possible option among the four methodologies.

4.1.3.4. Summarized results

The results of our exploratory study indicate that to work with Agile an organisation should be familiar with its practices or, at least, have professionals with some experience in its methodologies. It is important to have a project management culture in place and be able to implement changes quickly. Leaders should support the use of Agile practices and actively participate in Agile projects. They also should trust the team and let them self-organise. Development teams should be small (maximum of 12 people) and experienced (with high technical competence).

About the environmental factors to use Agile, they relate to the likelihood of changes (the higher, the more adequate), the level of access to business representatives (the higher, the better to work with Agile), and geographic distribution (the closer people are, the better to use Agile). Regarding project characteristics, software criticality (better is low to medium), the delivery approach (ideally multiple and frequent deliveries of parts of the software), and the level of documentation and formalization (the lower, the better to work with Agile) are the key aspects to consider.

Concerning the aspects that help to point out the best Agile methodology to use among Scrum, Extreme Programming, Kanban, and Lean Software Development, we can affirm that they relate to the emphasis of each methodology and also to how each one addresses: changes in system requirements; the delivery of the software; customer engagement; and the prioritisation of activities and system requirements in the project.

All these aspects should be assessed through a questionnaire capable of producing a positive or negative indication to use an Agile approach. In the case of a positive recommendation, this assessment should also have a set of questions to indicate the most suitable methodology among the four frameworks studied.

4.1.4. Brief conclusions from the Exploratory Study

Our results unveiled the essential aspects to consider within an Agile suitability assessment and the ideal way of conducting this assessment.

We have checked that, to evaluate the Agile suitability, we need to capture the maturity of the organisational environment, understand the reality of the company and the actual practical situation in terms of people. It is important to comprehend the vision of the organisation (from an environmental and behavioural perspective) at a strategic level and the level of development teams (the ones involved in software development). To understand the situation in terms of delivery and the difficulties that teams face are also essential factors to consider while assessing the Agile suitability.

Regarding the approach to perform the assessment, the exploratory study pointed out that it should be done in a perspective of questioning the business goals or the problem to be solved (the major motivations for the project), and not so much in applying the Agile tools and techniques available. The approach should be more problem-resolution oriented. The questionnaire could have three or four sections, one of them clearly related to the organisational context, and the others to the level of people, teams, and leadership competencies and characteristics.

We conclude that the findings of our exploratory study are aligned with the main Agile concepts (captured in the literature revision cf. items 2.3.4, 2.3.5, 2.4.2, 2.4.3, and 2.5), and provide the necessary support to elaborate a consistent conceptual model that supports the option for an Agile methodology. This conceptual model is the basis that will guide the development of the prototype of the decision support system proposed.

4.2. Conceptual Model

The key factors and aspects identified in the exploratory study were translated into questions aiming to assess the suitability of a development approach (Agile or non-Agile) and an Agile methodology (among the four studied). The result is a conceptual model that implements an Agile suitability assessment for a given software development context, recommending the use of an Agile approach when is the case, and indicating an Agile methodology to be used (when Agile fits).

The idea is to have a two-step assessment. The first step identifies (according to participants' recommendation) whether an Agile approach is appropriate or not. When Agile is appropriate, the second step identifies which of the four methodologies studied (Scrum, Extreme Programming, Kanban or Lean Software Development) is more suitable to the software development project in question (cf. Figure 22).

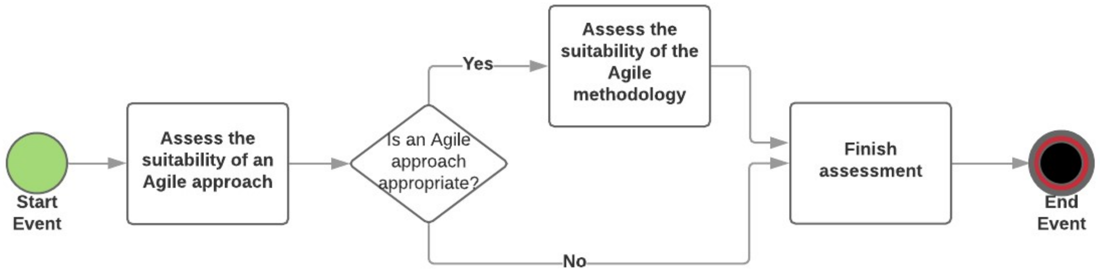

Start Event
Assess the suitability of an Agile approach
Is an Agile approach appropriate?
Yes
No
Assess the suitability of the Agile methodology
Finish assessment
End Event

Figure 22 - Agile suitability assessment flow (steps of the assessment)

4.2.1. Step 1 – Agile approach suitability assessment

Our findings suggest three groups of information to consider while assessing the Agile approach suitability (step 1) in the context of software development. These three groups represent categories to evaluate:

- Organisational Environment category – Includes the questions to assess the organisational context where the software will be developed;
- People category – Considers the questions that aim to assess the characteristics, behaviours, and competencies of leaders and teams involved in software development;
- Project Environment category – Contains questions whose goal is to evaluate project management capabilities, the context for the project under assessment and its product-related aspects.

The relevant aspects identified to be considered within each of these categories are depict in the following mind map (cf. Figure 23).

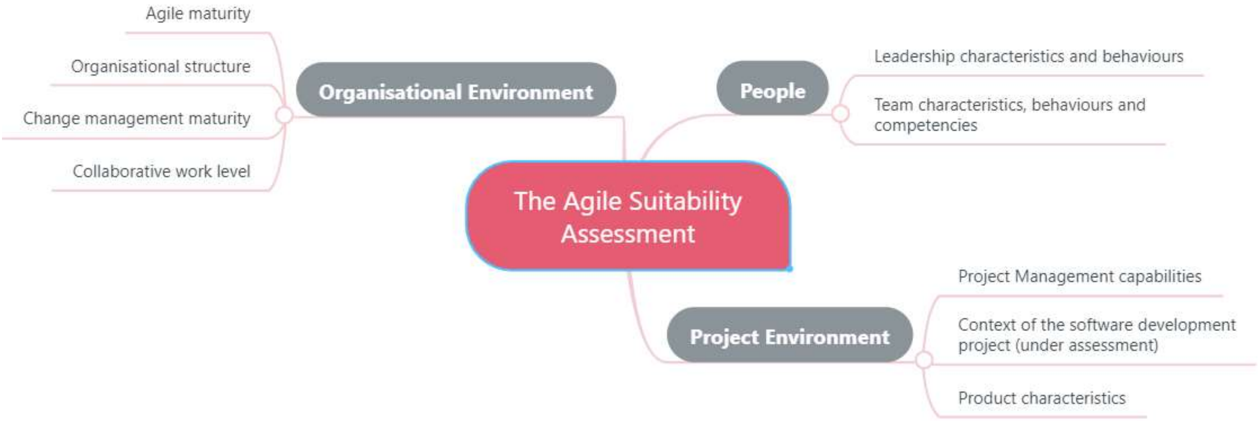

Agile maturity
Organisational structure
Change management maturity
Collaborative work level
Organisational Environment
People
Leadership characteristics and behaviours
Team characteristics, behaviours and competencies
The Agile Suitability Assessment
Project Management capabilities
Project Environment
Context of the software development project (under assessment)
Product characteristics

Figure 23 - Relevant aspects to assess by category of evaluation

To evaluate each of the nine relevant aspects identified (distributed into three categories), we elaborated questions to quantify (through a scoring scale) how much each aspect contributes to the use of an Agile approach. Regarding the possible answers to each question, for some of them, we used the support of the theory (throughout existing models and patterns) to elaborate the list of possible answers. In the details of each question, we explain the source of the answers and how many points (in brackets) each one contributes to using Agile approaches. The higher the number of points, the higher the contribution to the use of an Agile approach. This rule applies to all answers to the questions of the Agile approach suitability assessment.

4.2.1.1. Organisational Environment category questions

For the assessment of the Organisational Environment category, we propose four questions, one per aspect to be evaluated: Agile maturity; Organisational structure; Change management maturity; and Collaborative work level.

Question 01: Regarding the level of Agile maturity, which is the sentence that best describes the reality of the organisation?

To support the creation of the possible answers to this question we used as source the organisational Agile maturity level classification defined by Kourounakis et al. (2015), as originally described by the authors in the Agile@EC Guide.

Possible answers:

A. There is no history of using Agile practices. (0)
B. Sporadic development cycles and loose Agile practices. (25)
C. Agile practices are established, but there is a lack of common approach and consistency between teams. (50)
D. Following Agile principles repeatedly across all projects to achieve business and customer goals. (75)
E. Adopting Agile approaches at a strategic Enterprise level. (100)

Question 02: In relation to the organisational structure, select the sentence that best describes or is closer to the reality of the organisation.

To help in creating the possible answers to this question, we used the organisational structure types of projects defined by the Project Management Institute in the PMBOK Guide (PMI, 2017).

Possible answers:

A. The organisation is structured around projects. Most of the people are involved in project work. The Project Manager has high level of authority and controls project resources. (100)
B. Some of the people in the organisation are involved in project work. The organisation has full-time Project Managers and project administrative staff. The Project Manager has considerable authority over projects. (75)

C. The organisation has the role of Project Manager. However, the Project Manager does not have full authority over the project, project staff, or project budget. (50)

D. The Project Manager acts more like a Project Coordinator. Its ability to make or enforce decisions is low, and most of the authority remains with the Functional Manager. (25)

E. The organisation is structured around primary functions. The employee has one function and reports to the Functional Manager. The Functional Manager assigns and manages the employees work and handles administrative tasks. (0)

Question 03: Regarding the maturity of the organisation to implement changes when they are necessary, select the sentence that best describes or is closer to the reality of the organisation.

To support the creation of the possible answers to this question we used the Prosci's Change Management Maturity Model (ECM Lab, 2004). The model describes different levels of change management capabilities across organisations.

Possible answers:

A. People are not aware of the practices that exist to manage changes, and the organisation does not have any guideline established for managing the people side of change. (0)

B. Elements of change management can be seen in isolated parts of the organisation. The effort to manage the people side of change is sporadic and decentralized. (25)

C. There are groups applying a more structured process to manage changes, but this is still restricted to particular teams or areas of the organisation. (50)

D. The organisation has a defined common approach or standards for managing changes and tailors them to the specific needs of each project or process change. (75)

E. Change management competency is part of the daily practices in the organisation. (100)

Question 04: Regarding collaborative work in the organisation, indicate your level of agreement with each statement below.

In this question, we assess the collaborative work level in the organisation trough four components of collaborative work: coordination; cooperation; collaboration; and communication. They were defined by Patel, Pettitt, and Wilson (2012) in the research "Factors of collaborative working: A framework for a collaboration model". Each component is measured by a Likert scale[27]. For each of the different statements related to the four components, the possible answers and points (in brackets) are: *A. Fully disagree* (0); *B. Partially disagree* (25); *C. Do not agree nor disagree* (50); *D. Partially agree* (75); *E. Fully agree* (100).

Statements:

A. Most people in the organisation work together effectively, support each other, communicate well, and do their share. – coordination
B. Problem-solving occurs effectively within a group of people with different skill sets, and consensus is reached on most of decisions. – cooperation
C. Most people in the organisation share ideas and activities informally without being asked to do so. – collaboration
D. Communication is effective and transversal within the organisation, characterized by mutual trust. – communication

4.2.1.2. People category questions

For the People category evaluation, we propose two questions regarding Leadership and Team characteristics, behaviours, and competencies. Both questions are evaluated using Likert scales. For each question, there are ten statements. All of these statements were created supported by the results obtained in the exploratory study.

Question 01: Regarding the leadership characteristics and behaviours within the organisation or at the level of the projects, indicate your level of agreement with each statement below.

For each of the ten statements, the possible answers and points are:

[27] When responding to a Likert item, respondents specify their level of agreement or disagreement on a symmetric agree-disagree scale (East Carolina University, 2015).

A. Fully disagree (0); *B. Partially disagree* (25); *C. Do not agree nor disagree* (50); *D. Partially agree* (75); *E. Fully agree* (100).

Statements:

A. The leadership of the organisation or the project encourages and supports the use of Agile approaches in software development.

B. The leadership has confidence that the project team can transform the customer needs into a successful product, trusting the team has the best people to get the job done.

C. The leadership delegates and empowers the team so that they can make their own decisions, letting the team stand out and acting more as a coach and facilitator.

D. The leadership understands that Agile means experimentation and that it may lead to mistakes. They believe that these mistakes can be healthy to the project and that they should occur as soon as possible so the project team can anticipate issues and act on them.

E. The leadership incentives and promotes cross-functional and multidisciplinary teams.

F. The leadership knows very well your teams and stay close to them to be familiar with important themes and act to unblock issues as quickly as possible.

G. The leadership is versatile at the decision-making level, being able to justify and support the reasons for changes and transitions.

H. The leadership shields the team from external interference as much as possible.

I. The leadership is willing to face frequent changes in the course of the project, knowing that more changes will come, and then dealing with them naturally.

J. The leadership promotes a continuous improvement environment, often questioning what can be done to extend the benefits already obtained with new ways of work.

Question 02: Regarding the project team characteristics, behaviours and competences, indicate your level of agreement with each statement below.

For each of the ten statements of this question, the possible answers and points are:

- Statements A to D and F to J: *A. Fully disagree* (0); *B. Partially disagree* (25); *C. Do not agree nor disagree* (50); *D. Partially agree* (75); *E. Fully agree* (100).
- Statement E: *A. Up to 12* (100); *B. 13 to 20* (75); *C. 21 to 45* (50); *D. 46 to 80* (25); *E. More than 80* (0).

Statements:

A. The project team is experienced in Agile and constantly seek for self-improvement. They seek to know how other Agile teams (inside or outside the organisation) have been working.

B. The project team has the autonomy to make their own decisions about how to undertake the project work (self-organising team).

C. The project team has all competencies needed to accomplish the work without depending on others not part of the team (cross-functional team).

D. The project team has frequent access to at least one business / customer representative to ask questions and get feedback on the work.

E. The alternative that includes the size of the team that will undertake the project is (if not sure, choose the most likely option).

F. The team member is committed, engaged, and focused. He/she avoids (as much as possible) context switch.

G. The team member is accountable, responsible, transparent, and willing to share information.

H. The team member shows team spirit, has a sense of unity, seeks cohesion, and has good behaviour.

I. The team member feels free to raise problems and risks without fear.

J. The team member has an open mindset and quickly adapt to new work environments.

4.2.1.3. Project Environment category questions

For the assessment of the Project Environment category, we propose three questions per aspect evaluated: Project Management capabilities, the context of the software development project (under evaluation), and Product characteristics.

Question 01: Regarding the level of Project Management capabilities, which is the sentence that best describes the reality of the project?

To support the creation of possible answers to this question we used the assessed levels of project management maturity defined by Mullaly and Thomas (2010) in the paper "Re-thinking project management maturity: perspectives gained from explorations of fit and value."

Possible answers:

A. There is no formal implementation of Project Management. The processes to be used and the effectiveness of the results will come from the experience and expertise of individuals and the team. (0)

B. There are some Project Management capabilities defined and used at an organisational level, but they are incomplete or inconsistently applied. The project will have some level of formality, but not comprehensive nor fully applied. (25)

C. There is a comprehensive Project Management process in place and consistently applied to all projects within the organisation. (50)

D. There is a consistently defined Project Management process fully integrated into the management capabilities of the organisation and its lifecycle. (75)

E. There is a comprehensive, fully integrated approach to manage projects, which exists within an ongoing cycle of continuous improvement. (100)

Question 02: For each of the aspects related to the project under evaluation listed below, select the answer that best represents or is closer to the reality of the project.

In this question we considered essential to assess three aspects that lead to the use of Agile approaches, which are the uncertainty of system requirements, the need for co-locating the project team, and the necessity of tools and techniques that enhance communication and collaboration. Therefore, we divided this question into three sub-questions.

To create the possible answers to these sub-questions, we combined the results of the exploratory study with insights obtained from the literature revision. The main source of theoretical data was the “Agile Practice Guide”, published by the Project Management Institute (2017).

- For the first sub-question, the possible answers and points are: *A. Up to 5%* (0); *B. 6% to 15%* (25); *C. 16% to 30%* (50); *D. 31% to 45%* (75); *E. More than 45%* (100).

- For the second and third sub-questions, the possible answers and points are: *A. Not at all* (0); *B. It is indifferent* (25); *C. It would be nice* (50); *D. It would be important* (75); *E. It is mandatory* (100).

Sub-questions:

1. What is the degree of uncertainty on the system requirements before the project start or in the initial stages of the project?
2. Is there a real necessity for co-locating the project team and business / customer representatives to undertake this project?
3. Is there a need for the project to use tools (digital or not) and techniques to enhance the communication and collaboration between the project team and stakeholders?

Question 03: For each of the aspects related to the software resulting from the project under evaluation, select the answer that best represents or is closer to the reality of the project.

In this question, we assess two important characteristics of the software resulting from Agile projects, which are how critical a software failure is, and the delivery approach. Thus, we divided the question into two sub-questions.

In the definition of the answers for these sub-questions, we used as main source the Agile Suitability Filters, described in the Agile Practice Guide (PMI, 2017), which brings the concepts developed by the PMI regarding Agile suitability models (cf. item 2.4.4.4).

- For the first sub-question, the possible answers and points are: *Time or business opportunities* (100); *Discretionary funds* (75); *Essential funds* (50); *Single life* (25); *Many lives* (0).
- For the second sub-question, the possible answers and points are: *Yes* (100); *Very likely* (75); *It is indifferent* (50); *Very unlikely* (25); *No* (0).

Sub-questions:

1. Regarding the possible impact of defects presented by the software resulting from the project, what are the most likely related losses?
2. Can the product of the project be built and delivered in smaller chunks in different moments during the project?

4.2.1.4. The Agile approach suitability assessment

In our research, we conclude that the three categories identified are equally important in determining the Agile approach suitability. However, we also conclude that each aspect to be assessed within a category should contribute to the outcome of the evaluation with its own degree of importance since a specific aspect may influence more (or less) the rating of your category.

According to our findings, regarding the Organisational Environment, Agile maturity and Collaborative work are a little more important than Organisational structure and Change Management maturity when assessing the Agile suitability. Concerning the People category, we conclude that both leadership and team aspects influence the Agile adequacy the same way. In relation to the Project Environment, our findings suggest that the most significant aspect relates to the Context of the software development project (meaning the degree of uncertainty, co-location, and the need for tools/techniques to enhance communication and collaboration), followed by the Product characteristics, that we conclude to be more relevant than the Project Management capabilities to determine the Agile suitability.

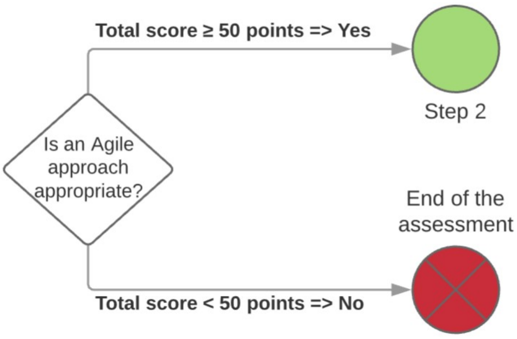
Therefore, we propose a weighted distribution to the questions formulated within each category (cf. Table 7).

Table 7 - Map of questions and distribution of weights by assessment category

Agile Approach Suitability Assessment Structure		[Weight]
Category	Organisational Environment	[33,33%]
Category questions	Question 01 – Agile maturity	[30%]
	Question 02 – Organisational structure	[20%]
	Question 03 – Change Management maturity	[20%]
	Question 04 – Collaborative work	[30%]
Category	People	[33,34%]
Category questions	Question 01 – Leadership characteristics and behaviours	[50%]
	Question 02 – Team characteristics, behaviours and competencies	[50%]
Category	Project Environment	[33,33%]
Category questions	Question 01 – Project Management capabilities	[15%]
	Question 02 – Context of the software development project (under assessment)	[50%]
	Question 03 – Product characteristics	[35%]

Note: The sum of the weights of all categories equals 100% of the assessment. The sum of the weights of all questions in each category equals 100% of the weight of the category.

The calculation of the Agile approach suitability assessment considers the points assigned to the answer selected in each question and its corresponding weight. A total score equal to or greater than 50 points indicates a result in favour of using an Agile approach, leading the evaluation process to step two. Otherwise, the assessment comes to an end (cf. Figure 24).


Total score ≥ 50 points => Yes
Step 2
Is an Agile approach appropriate?
End of the assessment
Total score < 50 points => No

The higher the score of the assessment, the higher the suitability of the project evaluated to use an Agile approach.

Figure 24 - Is an Agile approach appropriate?

The detailed equations for calculating the Agile approach suitability assessment are available in Appendix N, as well as the percentages of each question, sub-question, and statement considered in the calculations.

4.2.2. Step 2 – Agile methodology suitability assessment

The results from our exploratory study suggest five aspects to consider in order to perform the Agile methodology suitability assessment (step 2) in the context of an Agile software development effort.

These five aspects concentrate on the characteristics addressed in different ways by each of the four methodologies under study since all of them share a lot of Agile principles. Therefore, only focusing on aspects that have some differences is possible to identify which one is best suited for a given project. These aspects are presented in the mind map (cf. Figure 25) in order of relevance, from the most (1) to the least relevant (5).

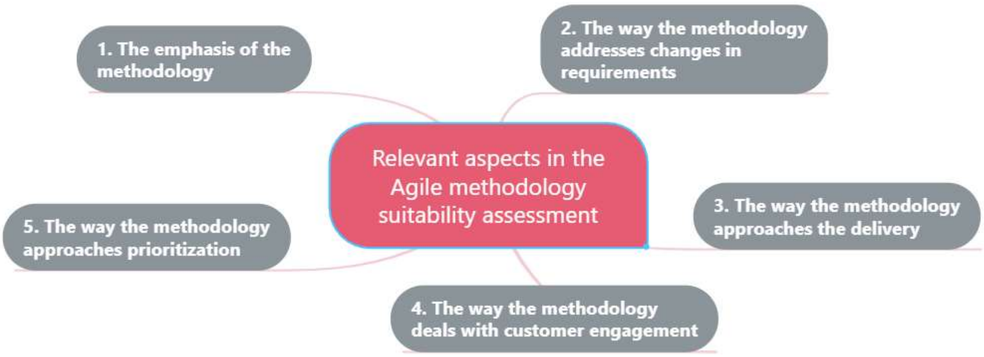

1. The emphasis of the methodology
2. The way the methodology addresses changes in requirements
Relevant aspects in the Agile methodology suitability assessment
3. The way the methodology approaches the delivery
5. The way the methodology approaches prioritization
4. The way the methodology deals with customer engagement

Figure 25 - Relevant aspects of the Agile methodology suitability assessment

To assess these aspects in the context of a software development project, we propose five questions, one per aspect evaluated. All questions formulated derive from the results of our exploratory study. The possible answers for each of the questions were elaborated in a way that allows the evaluator to select the answer that best matches the approach for the aspect assessed. Each answer may contribute fully, partially, or do not contribute to the use of one of the four methodologies.

The possible answers to each question were created with inputs from the theory (e.g., Scrum, XP, Kanban, and LSD principles, practices, tools, and techniques) and the results of the exploratory study as well (cf. item 4.1.3.2). In the details of each question, we show how much each answer contributes to the use of one of the four Agile methodologies.

4.2.2.1. Agile methodology suitability assessment questions

Question 01: Based on the known information about the software development project, select the alternative that best reflects or is closer to the emphasis that the Agile methodology to be applied must have so that it meets the needs and objectives of the project.

In this question, we propose four answers. Each one fully contributes to the use of one methodology. In this way, the selected methodology receives the maximum score (100), and the others are not scored (0).

Possible answers:

A. The project requires mainly a managerial framework. The focus is on productivity, ensuring the software is built quickly and with minimum complexities. An iterative and incremental approach is required to periodically release software increments.

   Points by methodology: Scrum = 100; XP = 0; Kanban = 0; LSD = 0.

B. The project needs a framework whose priority is the high quality of the interim and final software. To have this, the methodology must focus on applying sophisticated software engineering practices and work with short development cycles.

   Points by methodology: Scrum = 0; XP = 100; Kanban = 0; LSD = 0.

C. The project requires a methodology with a focus on improving the workflow and refraining from producing a surplus throughout full transparency of work and real time communication of capacity. It is key to have a clear view of the development process and address issues as quickly as possible.

Points by methodology: Scrum = 0; XP = 0; Kanban = 100; LSD = 0.

D. The project needs a framework whose focus is on streamlining the development process, cutting away all activities that do not directly contribute to the final software. The methodology does not necessarily have to work with fixed-length development cycles, but relies on an experienced team to apply best software engineering practices to deliver the project.

Points by methodology: Scrum = 0; XP = 0; Kanban = 0; LSD = 100.

Question 02: Based on the known requirements for this project and on what is known about the stakeholders as well, select the alternative that best reflects or is closer to the expected reality of the project in respect to the frequency the system requirements may change.

In this question, we propose four answers. Each one contributes fully and partially to the use of one methodology. This is justified by the fact that some of the frameworks similarly address changes in system requirements. Thus, the answers were created putting together those that work in a similar way but giving total emphasis to one methodology and partial to another. Any option selected will favour the use of two methodologies, one fully (100 points) and another one partially (50 points).

Possible answers:

A. Changes in requirements are expected, but the project approach will not embrace these changes once the planning for the development cycle (iteration) is done nor during an ongoing cycle. If any big change is needed, the project team typically stops the current cycle and starts a new one with new requirements.

Points by methodology: Scrum = 50; XP = 100; Kanban = 0; LSD = 0.

B. Changes in requirements are expected, but while working with stable fixed-length development cycles (iterations), embrace these changes during cycles will be accepted when strictly necessary, with the project maintaining the planned workload throughout the reprioritisation of features.

Points by methodology: Scrum = 100; XP = 50; Kanban = 0; LSD = 0.

C. Frequent changes in requirements and project scope are the norm. These changes will typically be accepted, implemented, and delivered as fast as possible since one of the project's priority is to deliver value to the customer as soon as possible.
   Points by methodology: Scrum = 0; XP = 0; Kanban = 50; LSD = 100.

D. Frequent changes in requirements are expected. These changes will be accommodated in the product backlog according to the priority and will be executed as soon as there is available capacity in the team.
   Points by methodology: Scrum = 0; XP = 0; Kanban = 100; LSD = 50.

Question 03: Based on the known systems requirements for the project and the general project objectives, select the alternative that best matches the context of the project regarding the delivery approach.

In this question, we propose two answers. Each option contributes fully to the use of two methodologies. The reason is that both methodologies similarly approach the software delivery method. Therefore, the answers were created describing the two different approaches. The answer selected favours the use of two methodologies and disfavours the other two.

Possible answers:

A. The project requires an approach that enables frequent delivery of parts of the software in a planned manner. The scope of the releases will mostly contain business features defined by the client. There is flexibility for changing requirements, but the negotiation for the implementation of such changes must occur to maintain the balance of the backlog, minimizing risks for the continuity of the delivery and the quality of the software.
   Points by methodology: Scrum = 100; XP = 100; Kanban = 0; LSD = 0.

B. The project requires an approach that enables the delivery of parts of the software as fast as possible. There must be high flexibility for the implementation and rapid availability of changes. For this, it is essential to have an optimized and constantly revised workflow, minimizing risks of creating bottlenecks or performing tasks that do not directly contribute to the software.
   Points by methodology: Scrum = 0; XP = 0; Kanban = 100; LSD = 100.

Question 04: Based on the needs and objectives of the project and on what is known about the stakeholders as well, select the alternative that reflects or more closely matches what is required or expected in terms of customers or business representatives engagement in the project.

In this question, we propose three answers. The first option indicates Extreme Programming as adequate, scoring 100 points. The second option describes the way Scrum engages the customer in the project, scoring 100 points. The last answer option defines the Kanban and LSD approach regarding the involvement of business representatives in the project. Therefore, in this question, the scoring method is particular since it may score XP or Scrum exclusively, but when the third alternative is chosen, both Kanban and Lean Software Development are scored (100 points).

Possible answers:

A. The customer or business representatives will be actively engaged in the project on a daily basis, integrating the project team.

Points by methodology: Scrum = 0; XP = 100; Kanban = 0; LSD = 0.

B. The customer or business representatives will interact frequently with the project team and participate in important project activities (like iteration planning and reviews), providing frequent feedback on software increments.

Points by methodology: Scrum = 100; XP = 0; Kanban = 0; LSD = 0.

C. The customer or business representatives will be mainly involved in the project to identify value streams and to contribute to the optimization of the development process.

Points by methodology: Scrum = 0; XP = 0; Kanban = 100; LSD = 100.

Question 05: Based on the project needs and objectives and also on what is known about requirements, select the alternative that represents or more closely matches what is expected regarding how business requirements, system features, or project activities should be prioritised in the project.

In this question, we propose four answers. The selected methodology is fully scored (100), and the others are not (0).

Possible answers:

A. The customer or business representative will define the priority of the features within a development cycle, but the project team will perform the work in the sequence they deem to be adequate for the objectives set for the cycle.

Points by methodology: Scrum = 100; XP = 0; Kanban = 0; LSD = 0.

B. The project team will work on the planned features for the development cycle in the order defined by the customer or business representative. The project team will not be allowed to modify the order of the work.

Points by methodology: Scrum = 0; XP = 100; Kanban = 0; LSD = 0.

C. The prioritisation of the job will be done by whoever is in charge of it, in accordance to the project needs. The main concern is pulling work and often reprioritise remaining tasks to keep a paced workflow.

Points by methodology: Scrum = 0; XP = 0; Kanban = 100; LSD = 0.

D. The prioritisation of the work will be done by whoever is in charge of it, according to the needs of the project. The main concern is to avoid unnecessary work and prioritise tasks that directly add value to the software.

Points by methodology: Scrum = 0; XP = 0; Kanban = 0; LSD = 100.

4.2.2.2. The Agile methodology suitability assessment

We conclude that each aspect that helps to identify the most suitable methodology should contribute differently to the outcome of the assessment since all aspects do not have the same influence. We propose a weighted distribution to the formulated questions according to the relevance of each aspect assessed (cf. Table 8).

Table 8 - Agile methodology assessment map of questions and weights

Agile Methodology Suitability Assessment Structure	[Weight]
Question 1 – The emphasis of the methodology	[30%]
Question 2 – The way the methodology addresses changes in Requirements	[25%]
Question 3 – The way the methodology approaches the delivery	[20%]
Question 4 – The way the methodology deals with customer engagement	[15%]
Question 5 – The way the methodology approaches prioritisation	[10%]

Note: The sum of the weights of all questions equals 100% of the assessment.

The calculation of the Agile methodology suitability assessment considers the points assigned to the answer selected in each question and the weight of the question. The methodology with the highest score is the one most suited for the project under assessment. The assessment is then concluded.

The equation for calculating the Agile methodology suitability assessment is available in Appendix N.

4.2.3. Conceptual model validation

To validate the conceived conceptual model, we used data from completed projects that: i. used one of the four Agile methodologies studied; ii. used a traditional (non-Agile) development methodology. The idea is to compare the approach (Agile / non-Agile) and the methodology suggested by the model (when the approach suggested is Agile) with the approach and methodology used in these projects. The higher the matches, the greater the confidence that the conceptual model is adequate to evaluate the suitability of Agile for software development projects.

We used data from 56 real software development projects undertaken between January 2012 and October 2018. Only completed projects whose evaluation by the customer (at the end of the project) obtained at least classification "Satisfied (4)", in a scale that varies from "Very Unsatisfied (1)" to "Very Satisfied (5)", were considered in order to have more assertiveness in the validation.

Table 9 summarizes the main data of these projects.

Table 9 - Consolidated data of the projects used to validate the conceptual model

Countries	Industries	Project Types	Methodologies
United States (19) Canada (9) Brazil (8) Germany (7) United Kingdom (5) Belgium (4) Portugal (4)	Financial Services (11) Internet (8) Energy-Oil & Gas (6) Computer-Software (5) Food (5) Govern (5) Media-all types (5) Health care (4) Telecommunications (2) Other (2) Hospitality (1) Manufacturing (1) Mining (1)	Processes / Efficiency (31) Sales / Revenue (20) Compliance (4) Research & Development (1)	Scrum (29) Kanban (11) Non-Agile (7) Extreme Programming (6) Lean Software Development (3)

4.2.3.1. Main characteristics of the sample and way of obtaining

The sample of project data used to validate the model presents the following main characteristics:

- 50% of the projects were executed in North America, 36% in Europe, and 14% in South America;
- Financial Services (20%) and Internet (14%) are the two industries that concentrate most of the projects of the sample;
- Increase Processes and/or Efficiency, or Sales and/or Revenue were the main reasons for undertaking these projects, accounting together 91% of projects;
- Data about the development approaches and Agile methodologies used in these projects are represented in Figures 26 and 27.

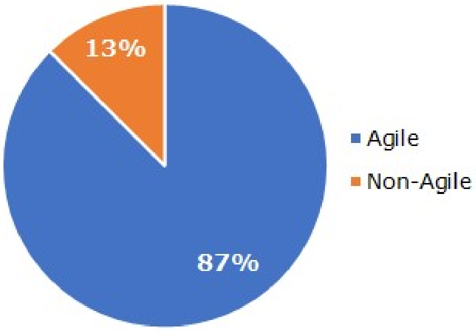


Figure 26 - Development approaches used

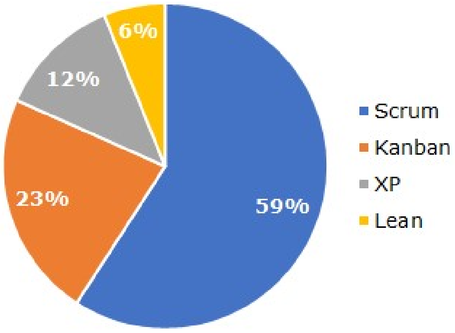


Figure 27 - Agile methodologies applied (when the approach is Agile)

Data from 48 projects (86% of the sample) came from a recognized global IT consulting company, which did not authorize the disclosure of its name nor the identification of the clients of such projects due to non-disclosure agreements. A form containing the questions and answers defined (cf. items 4.2.1 and 4.2.2) was created in Microsoft Excel. A template of this form is available in Appendix M. All 48 forms were completed by representatives of the company's Project Management Offices (PMO's). We also count on the help of a Senior Manager from the company to support and clarify any respondents' doubts while answering the questionnaire. The eight remaining forms (completing 56 projects) were filled in by the dissertation's author, who used the information on past software development projects he managed (with the consent of the clients).

4.2.3.2. Steps of the validation process

After having the 56 questionnaires completed, the analysis of the responses occurred in two stages. First, we verified the answers of the first part of the questionnaire regarding the projects' Agile approach suitability. We separated the ones that used a traditional approach (N=7) from those that used an Agile approach (N=49) and started the validation by the first group to check the results the conceptual model would suggest for these seven projects. After consolidating the results, we repeated the process for the 49 projects that used an Agile approach to see which approach the model would recommend. These two activities completed the validation of step 1 of the conceptual model (Agile Approach Suitability Assessment).

The second stage of validation was to verify which Agile methodology the conceptual model suggested for each of the 49 projects that employed an Agile approach. This completed the validation of step 2 (Agile Methodology Suitability Assessment).

4.2.3.3. Results

We calculated the results of the questionnaires with the help of an Excel spreadsheet. Each result was compared to the methodology used by the project. As already explained, the objective was to obtain as many matches as possible in order to confirm the effectiveness of the concept model.

Regarding the assessment of the suitability of an Agile approach (step 1 of the model), Table 10 shows the results obtained for the non-Agile projects.

Table 10 - Results of the Agile approach suitability assessments (non-Agile projects)

Proj. ID	Country	Industry	Project Type	Methodology Used	Assessment Result		
					Score	Approach Suggested	
P01	Brazil	Energy-Oil & Gas	Compliance	Non-Agile	28	Non-Agile	✓
P09	United Kingdom	Financial Services	Processes / Efficiency	Non-Agile	45	Non-Agile	✓
P23	Belgium	Govern	Compliance	Non-Agile	44	Non-Agile	✓
P26	Brazil	Telecommunications	Processes / Efficiency	Non-Agile	51	Agile	⊗
P30	United States	Govern	Compliance	Non-Agile	35	Non-Agile	✓
P34	United States	Energy-Oil & Gas	Compliance	Non-Agile	33	Non-Agile	✓
P44	Germany	Health care	Processes / Efficiency	Non-Agile	47	Non-Agile	✓

Within the seven evaluated projects that did not use Agile methodologies, six assessments (86%) obtained results confirming the use of a non-Agile approach. Only one evaluation pointed out that an Agile approach could be applied; however, it was verified that the score obtained (51 points) by this project (P26) is very close to the suggested threshold (50 points).

In relation to the results of the Agile projects, the assessments confirmed that all 49 projects (100%) pointed out the suitability for an Agile approach. The lowest score obtained was 50, by the project P08, and the highest score (88) was presented by the project P47. The complete list with the results obtained by each project is available in Appendix O (cf. Table 29).

Therefore, based on the results obtained, we conclude that the model is adequate to evaluate the suitability of an Agile approach, and we do not believe it is necessary to modify the scoring format suggested by the model.

Regarding the Agile methodology suitability assessments (step 2 of the conceptual model), Table 11 presents the consolidated results.

Table 11 - Consolidated results of the Agile methodology suitability assessments

No. of projects	Methodology used	Methodology recommended by the model
29	Scrum	✓ Scrum » 27 recommendations ⊗ Extreme Programming » 02 recommendations
11	Kanban	✓ Kanban » 10 recommendations ⊗ LSD » 01 recommendation
06	Extreme Programming (XP)	✓ Extreme Programming » 06 recommendations
03	Lean Software Development (LSD)	✓ LSD » 03 recommendations

The assessments showed that:

- From the 29 projects that applied Scrum, 27 (93%) assessments pointed out that the methodology is adequate. Only two assessments (7%) suggested a different framework (Extreme Programming). However, it does not invalidate the selection method proposed since Scrum and XP are methodologies that share many characteristics. The lowest score obtained was 70, by the projects P35 and P54, and 15 projects obtained the maximum score (100 points);
- From the 11 projects that used Kanban, 10 (91%) assessments indicated that Kanban is adequate. Only one assessment suggested Lean Software Development as a framework more suitable. This is also predictable since

Kanban and LSD are similar. The lowest score obtained was 65 by the projects P06 and P37. Four projects (P18, P42, P46, P56) scored 100;

- From the six projects that applied Extreme Programming (XP), the assessment of all of them pointed out that the methodology is adequate. The lowest score (75) was obtained by the project P17, and four projects (P12, P21, P48, and P51) got the highest score (100);
- From the three projects that used Lean Software Development (LSD) as methodology, all of them confirmed the methodology to be suitable. All projects scored the maximum number of points (100).

A detailed table showing the score obtained by each of the 49 projects that used an Agile methodology is available in Appendix O (cf. Table 30).

From the 49 evaluated projects that used Agile methodologies, the model assertively confirmed 46 results (94%). The three assessments that presented different results suggested an alternative framework with characteristics similar to the framework used in the project.

Given the assertiveness of the results obtained by the model when compared to the approaches (step 1) and development methodologies (step 2) used in real projects, we conclude that the conceptual model conceived to assess the Agile suitability for a software development project is adequate and can be used as basis for the construction of a decision support system for this purpose.

4.3. Requirements Gathering

The IEEE (1990) defines "Requirement" as a condition or capability needed by a user to solve a problem or achieve an objective. To uncover these conditions and capabilities in the early stages of software development, the development process usually starts with the requirements gathering.

Requirements gathering is the process of discovering the degree the software meets the purpose it is intended by identifying its requirements and documenting them in a form that is amenable to analysis, communication, and subsequent implementation (Nuseibeh & Easterbrook, 2000). Many techniques may be used to gather system requirements (observation, interviews, focus group, etc.), and though Agile does not prescribe any specific practice for uncovering requirements, in this work we utilized User Stories, as it will be presented in more details in chapter five.

According to Griffiths (2015), because system requirements are linked to the needs of future users, it is essential to determine in advance what these needs are. It helps to clarify the system purpose, avoiding something that will not meet users' needs to be built. In this work, two needs were identified as drivers for requirements definition (and prototype development):

- Need 1 (N01): To know if an Agile approach is appropriate for a given software development project;
- Need 2 (N02): To know the most appropriate Agile methodology for a given software development context (when Agile is adequate).

Having these two needs as drivers, all requirements identified in the scope of this work must (somehow) be linked to one or both needs.

4.3.1. High-level system requirements

From the analysis of the results of the interview and focus groups (guided by the needs to be met with the future system), the following high-level (essential) system requirements were identified (cf. Table 12).

Table 12 - High-level system requirements

ID	High-level System Requirements	Related Need(s)
R01	The process to determine the Agile suitability should be done through a solution that allows the assessment of the information collected, like a questionnaire.	N01/N02
R02	The Agile suitability assessment should take into account the management perspective (directive layer).	N01
R03	The Agile suitability assessment should take into account the solution delivery perspective (performing layer).	N01/N02
R04	The assessment result should indicate a favourable (or not) recommendation for Agile's utilisation.	N01
R05	In case Agile is suitable, the assessment result should also provide a recommendation on which Agile methodology is most appropriate.	N02
R06	The assessment result should be presented immediately, enabling the decision-maker to know the outcome right after the assessment.	N01/N02
R07	The system must allow the evaluator to maintain the information related to the assessments.	N01/N02
R08	The system must allow the evaluator to view / print previous assessments.	N01/N02

In the next chapter (cf. item 5.1.2), these high-level system requirements are detailed and described using User Stories.

Chapter 5 – Prototype Development

This chapter presents the work to design the system proposed and develop the prototype of this system. This work is organised in three phases: Initiation, Planning, and Execution. The activities performed and the Agile practices and techniques applied in these phases are described next.

5.1. Initiation

The main objective of the Initiation phase was to identify and describe the potential system users and requirements. Three Agile tools and techniques were used to reach these objectives:

- The Agile technique "Personas" was applied in the identification and description (mapping) of the potential system users;
- "User Stories" were used to detail and describe the high-level system requirements identified during the requirements gathering;
- A "Product Backlog" containing a prioritised list of User Stories was created to provide an overview of the detailed requirements and their priorities.

5.1.1. Users mapping with Personas

One of the key steps to develop a new system is to identify its target audience. This can be difficult since, at first glance, only the end user may be seen as a beneficiary, but this is not always the case. To identify to whom the system is intended, it is important to focus on who may benefit from it, and how. To do so, Agile recommends using Personas.

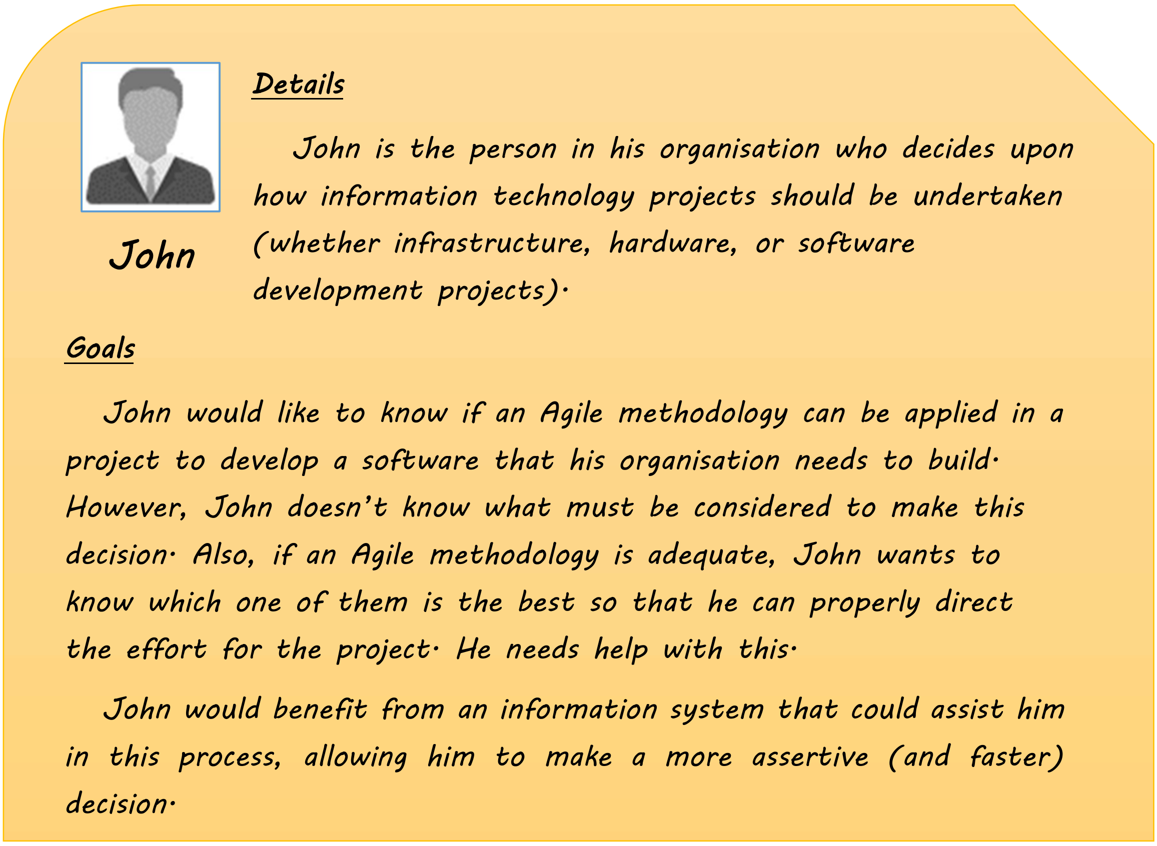
Griffiths (2015) explains that Personas are representations of the key stakeholders on the project. The technique brings the system users closer to the development team, which helps drive the development effort towards a solution more suitable to the user interests. Personas can identify real people or profiles. In this work, Personas describe the different type of people (roles) who might benefit from the future system. A typical way to describe a Persona is using a "card" divided into three parts:

- Identification – Name and picture (or an image) of the Persona;
- Details – Persona's relevant characteristics and behaviours (only relevant details are included);
- Goals – Reasons that could drive the Persona to use the system; benefits he/she wants to achieve; and problems the system could help to solve.

Four personas were identified as potential users of our system, giving the objective of the system proposed: the decision-maker; the development leader; the Agile consultant; and the Agile researcher.

According to Dam and Siang (2018), one way of using Personas is to capture a clear vision of the role the users play in their organisation. An examination of these roles can help to create better design decisions. This approach is known as "Personas role-based perspective". One way of applying this approach involves giving personas fictional names, in an attempt to "humanize" a certain role. This is aligned with the Personas goal of "bringing the system users closer to the development team." In this work, the four Personas were named as:

- *John*, the decision-maker (cf. Figure 28);
- *Angela*, the development leader (cf. Figure 29);
- *Peter*, the Agile consultant (cf. Figure 30);
- *Anthony*, the Agile researcher (cf. Figure 31).



John

Details

John is the person in his organisation who decides upon how information technology projects should be undertaken (whether infrastructure, hardware, or software development projects).

Goals

John would like to know if an Agile methodology can be applied in a project to develop a software that his organisation needs to build. However, John doesn't know what must be considered to make this decision. Also, if an Agile methodology is adequate, John wants to know which one of them is the best so that he can properly direct the effort for the project. He needs help with this.

John would benefit from an information system that could assist him in this process, allowing him to make a more assertive (and faster) decision.

Figure 28 - Persona card - John

Angela

Details

Angela is the leader of the systems development area in her organisation and has as main attributions to guide the developers in their daily activities, and ensure the correct execution of the company's development processes.

Goals

Each software development project that Angela's area executes contains an essential step, which is the determination of the most appropriate methodology for the project. This is an important choice that Angela needs to make in order to increase the chances of succeeding in her projects.

Having an information system that could help Angela in this decision-making process would be of great benefit to her.

Figure 29 - Persona card - Angela

Peter

Details

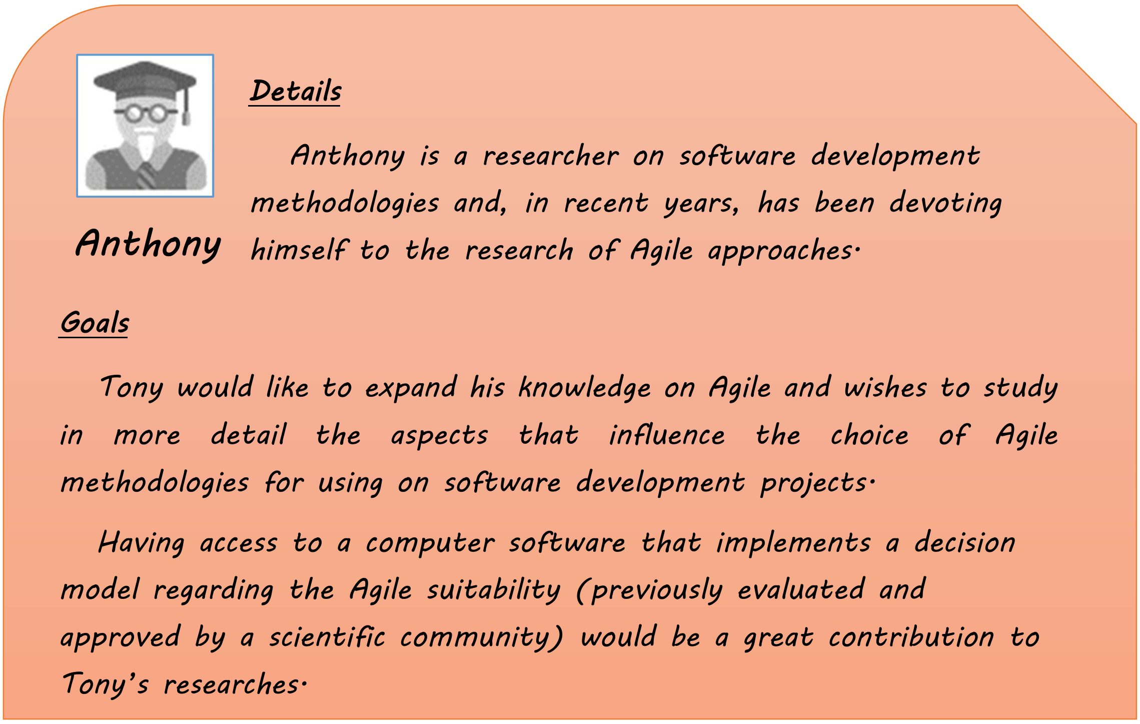
Peter is an Agile specialist and works for an IT consultancy company. Because of his extensive knowledge on Agile methodologies and software development he is usually assigned to work with clients on Agile projects.

Goals

As an Agile consultant, Peter needs to help his clients with all types of Agile matters, including deciding when to use an Agile methodology.

Although Peter is fully capable of indicating when a system would be better developed using Agile, having access to an information system that could support him in this process would be an asset so that he could carry out his job more efficiently.

Figure 30 - Persona card - Peter



Anthony

Details

Anthony is a researcher on software development methodologies and, in recent years, has been devoting himself to the research of Agile approaches.

Goals

Tony would like to expand his knowledge on Agile and wishes to study in more detail the aspects that influence the choice of Agile methodologies for using on software development projects.

Having access to a computer software that implements a decision model regarding the Agile suitability (previously evaluated and approved by a scientific community) would be a great contribution to Tony's researches.

Figure 31 - Persona card - Anthony

5.1.2. System requirements specification with User Stories

As presented in chapter four, in the interview and focus groups, participants were asked to describe the essential requirements for the system to be developed. At that phase of the work, the objective was to collect their vision without concerning about capturing requirements in detail. This section aims to explore these high-level requirements, detailing and describing them.

According to Griffiths (2015), once the high-level system requirements are gathered, they need to be organised in a way that allows the development team to understand, estimate, plan and track them. Although there is no unique way to describe system requirements in Agile, a simple and common approach is through User Stories. A User Story describes a software feature from the client's point of view, and the functionalities in the resultant software meet the requirements described in the User Stories (Sutherland, 2015).

Although there is more than one way to write User Stories, they are frequently written in the format: "As a <role>, I want / need <functionality>, so that <benefit>." In addition to the description, every User Story must have one or more acceptance criteria. The main advantage of this template is that it helps to identify who is the user and what is the benefit intended for every piece of functionality (Sutherland, 2015).

After breaking down the high-level requirements into User Stories, they are written on index cards or entered into a requirements management tool. The User Stories are then prioritised by value (Griffiths, 2015). Regarding the prioritisation, the author says that one of the most used techniques in Agile is the "MoSCoW" scheme, whose name derives from the first letters of the following labels:

- Must have: Stories that cannot be de-scoped. They usually represent the features of a Minimum Viable Product (MVP);
- Should have: Stories that can be de-scoped as a last resort;
- Could have: Stories that can be de-scoped without bringing significant impact to the system objectives;
- Would like to have (but not this time): De-scoped stories.

5.1.2.1. User Stories description and prioritisation

To describe and prioritise the User Stories, we used the same data usually contained in a User Story index card. All User Stories described are linked to the correspondent high-level system requirement (cf. table 12):

- User Story Title: A short title to identify the User Story;
- User Story Description: Description of the User Story using the format "As a <role>, I want / need <functionality>, so that <benefit>.";
- Priority: Relative priority of the User Story using MoSCoW technique;
- Acceptance Criteria: The condition to consider the User Story completed.

19 User Stories were identified (cf. Tables 13 to 20):

Table 13 - User Stories of the high-level system requirement R01

R01: The process to determine the Agile suitability should be done through a solution that allows the assessment of the information collected, like a questionnaire.			
User Story Title	User Story Description	Priority	Acceptance Criteria
Agile suitability assessment questionnaire	As an evaluator, I want to have a questionnaire available through a browser so that I can make an Agile suitability assessment online.	Must have	The questionnaire must work free of errors in a Web Browser (most recent version).
Different types of answers to the questionnaire's questions	As an evaluator, I need the questionnaire to contain questions allowing different types of answers (single choice, Likert scale), so that it is possible to have a range of possible questions and answers.	Must have	The question whose type of answer is: • Single choice - Must allow selecting only one answer; • Likert scale - Must allow selecting only one answer (level of agreement / disagreement).

Questions with specific weights in the questionnaire	As an evaluator, I want the questionnaire to allow the questions to have specific weights so that the outcome of the assessment considers the relevance of each characteristic assessed through a question.	Must have	The assessment result must consider the weight of the question in the final result.
Weight of the questions hidden in the questionnaire	As an evaluator, I want the questionnaire not to show the weight of a question, so that this information does not compromise my assessment process.	Must have	The questionnaire must not show the weight of the questions.
Saving a partial assessment	As an evaluator, I want the questionnaire might be saved as is so that I can complete my assessment later.	Must have	The questionnaire must allow being saved partially assessed.

Table 14 - User Stories of the high-level system requirement R02

R02: The Agile suitability assessment should take into account the management perspective (directive layer).			
User Story Title	User Story Description	Priority	Acceptance Criteria
The assessment must consider the Management perspective	As an evaluator, I need the assessment to consider the different aspects evaluated from the point of view of the Management so that the assessment considers the inputs from the directive layer.	Must have	The assessment result includes these aspects in the calculation (according to the decision model defined).

Table 15 - User Stories of the high-level system requirement R03

R03: The Agile suitability assessment should take into account the solution delivery perspective (performing layer).			
User Story Title	User Story Description	Priority	Acceptance Criteria
The assessment must consider the Solution Delivery perspective	As an evaluator, I need the assessment to consider the different aspects evaluated from the point of view of the Solution Delivery so that the assessment considers the inputs from the performing layer.	Must have	The assessment result includes these aspects in the calculation (according to the decision model defined).

Table 16 - User Stories of the high-level system requirement R04

R04: The assessment result should indicate a favourable (or not) recommendation for Agile's utilisation.			
User Story Title	User Story Description	Priority	Acceptance Criteria
Showing the score of Agile adherences to the evaluated context	As an evaluator, I want the solution to point out when Agile is suitable, showing the score of adherences obtained in the evaluated project scenario so that I can decide or help someone to decide for an Agile approach.	Must have	The assessment result must show the final score. A score equal or higher than 50 indicates Agile suitability.

Table 17 - User Stories of the high-level system requirement R05

R05: In case Agile is suitable, the assessment result should also provide a recommendation on which Agile methodology is most appropriate.			
User Story Title	User Story Description	Priority	Acceptance Criteria
The system points out Scrum as the most suitable methodology	As an evaluator, I want the solution to indicate when SCRUM is the most suitable methodology for a certain project scenario so that I can decide or help someone to decide in favour of using SCRUM.	Must have	The assessment result must indicate SCRUM as suitable and the score of the assessment.
The system points out XP as the most suitable methodology	As an evaluator, I want the solution to indicate when XP is the most suitable methodology for a certain project scenario so that I can decide or help someone to decide in favour of using XP.	Must have	The assessment result must indicate XP as suitable and the score of the assessment.
The system points out Kanban as the most suitable methodology	As an evaluator, I want the solution to indicate when KANBAN is the most suitable methodology for a certain project scenario so that I can decide or help someone to decide in favour of using KANBAN.	Must have	The assessment result must indicate Kanban as suitable and the score of the assessment.
The system points out LSD as the most suitable methodology	As an evaluator, I want the solution to indicate when LSD is the most suitable methodology for a certain project scenario so that I can decide or help someone to decide in favour of using LSD.	Must have	The assessment result must indicate LSD as suitable and the score of the assessment.

Table 18 - User Stories of the high-level system requirement R06

R06: The assessment result should be presented immediately, enabling the decision-maker to know the outcome right after the assessment.			
User Story Title	User Story Description	Priority	Acceptance Criteria
Viewing the assessment result on the system screen	As an evaluator, I want to see the assessment result on the screen, so that I can check the result immediately.	Must have	After finishing the assessment, the result(s) must be displayed on the system screen.
Printing the assessment result	As an evaluator, I want to print the assessment result, so that I can have a printed version.	Should have	After finishing the assessment: 1. A version of the result(s) formatted for printing must be printed; 2. A version of the result(s) formatted for printing must be saved in PDF.
Mailing the assessment result	As an evaluator, I want to e-mail me / someone else the assessment result, so that I / other recipients receive the result by e-mail.	Could have	After finishing the assessment, a version of the result(s) formatted for mailing must be sent by e-mail, and the message must be received by a valid e-mail address.

Table 19 - User Stories of the high-level system requirement R07

R07: The system must allow the evaluator to maintain the information related to the assessments.			
User Story Title	User Story Description	Priority	Acceptance Criteria
Updating project information	As an evaluator, I want the system to allow me to modify the project information (name and summary) so that I can adjust this information as needed.	Should have	Project information updated in the database.
Deleting a previous assessment	As an evaluator, I want the system to allow me to delete a previous assessment so that I can eliminate unnecessary data from the system.	Must have	Assessment information deleted from the database.

Table 20 - User Stories of the high-level system requirement R08

R08: The system must allow the evaluator to view / print previous assessments.			
User Story Title	User Story Description	Priority	Acceptance Criteria
Viewing a previous assessment	As an evaluator, I want the system to allow me to view a previous assessment so that I can consult the information of the assessment.	Must have	Assessment report loaded on the system screen showing all information related to it.
Printing a previous assessment	As an evaluator, I want the system to allow me to print a previous assessment so that I can have a printed version.	Should have	1. A version of the result(s) formatted for printing must be printed; 2. A version of the result(s) formatted for printing must be saved in PDF.

5.1.3. Prioritisation of User Stories with Product Backlog

The Project Management Institute (2017) defines a Product Backlog as a list of new features, changes to existing features, bug fixes, or other related products that an Agile team delivers as part of an Agile effort. In Agile development, these features, bugs, changes, and other products are usually represented by User Stories.

This topic aims to prioritise the User Stories according to the importance of each one within the development of the proposed system. This prioritisation supports the definition of the features of the system included in the minimum viable product (MPV) to be developed, that is, a version containing just enough features that allow the use of the system.

The Product Backlog is structured as:

- User Story Title: The short title identifying the User Story;
- Priority: Relative priority of the User Story (MoSCoW technique);
- Order: Priority of the User Story in relation to the other User Stories.

Table 21 - The Product Backlog

Product Backlog		
User Story Title	Priority	Order
Agile suitability assessment questionnaire	Must have	1
Different types of answers to the questionnaire's questions	Must have	2
Questions with specific weights in the questionnaire	Must have	3
Weight of the questions hidden in the questionnaire	Must have	4
Saving a partial assessment	Must have	5
The assessment must consider the Management perspective	Must have	6
The assessment must consider the Solution Delivery perspective	Must have	7
Showing the score of Agile adherences to the evaluated context	Must have	8
The system points out Scrum as the most suitable methodology	Must have	9
The system points out XP as the most suitable methodology	Must have	10
The system points out Kanban as the most suitable methodology	Must have	11
The system points out LSD as the most suitable methodology	Must have	12
Viewing the assessment result on the system screen	Must have	13
Deleting a previous assessment	Must have	14
Viewing a previous assessment	Must have	15
Updating project information	Should have	16
Printing the assessment result	Should have	17
Printing a previous assessment	Should have	18
Mailing the assessment result	Could have	19

The User Stories from priority 1 to 15 are the ones chosen to be included in the scope of the minimum viable product (cf. blue rows in Table 21). The User Stories from priority 16 to 19 are out of the initial scope and may be included in a future release (cf. yellow rows in Table 21).

5.2. Planning

The Planning phase aims to identify and organise the activities to build the software in an Agile manner. Since that in terms of the development the scope of this work is limited to the construction of a functional prototype, the main output of this phase is a Work Plan for the system design and prototype development, with the support of an electronic Kanban board.

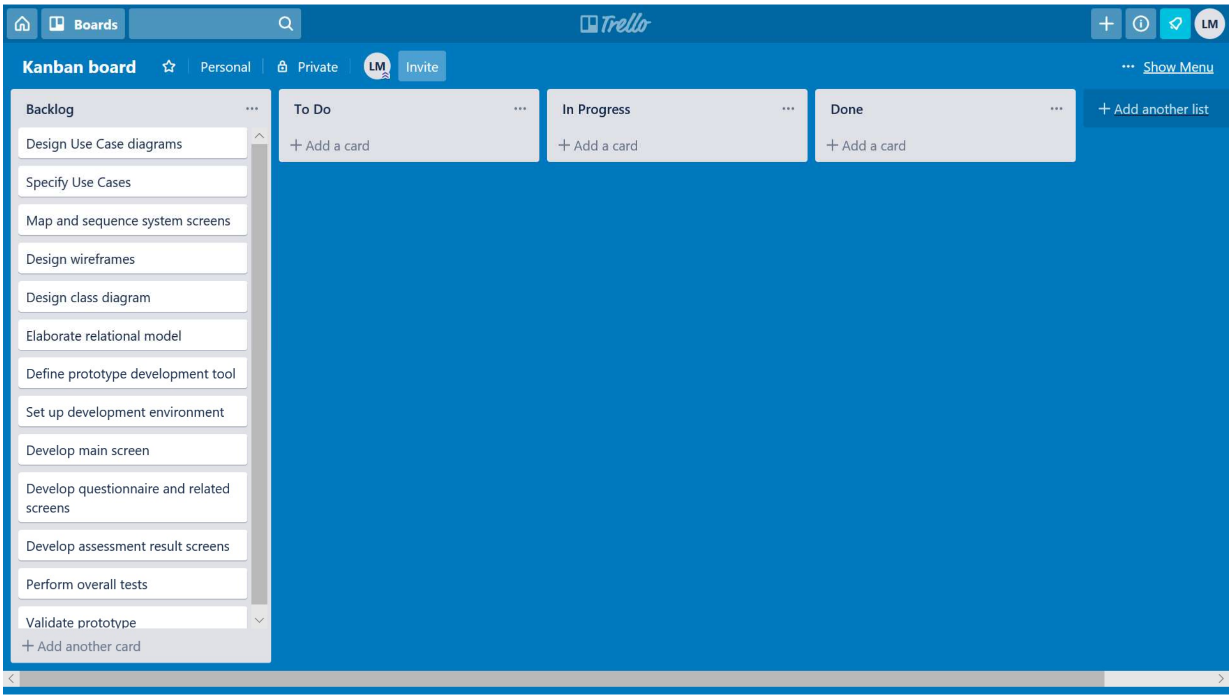
5.2.1. Activities identification

This section identifies the activities necessary to transform the system requirements into features. This includes the activities related to system design and modelling, and prototype development and validation:

- User-System interaction design
  - Design the Use Case diagrams
  - Specify the Use Cases
- Screens design
  - Map and sequence the system screens
  - Design the wireframes (sketches of the system screens)
- Data modelling
  - Design the class diagram
  - Elaborate the relational model
- Prototype development (iterative and incremental)
  - Define a prototype development tool
  - Set up the development environment
  - Develop the main screen
  - Develop the questionnaire and related screens
  - Develop the assessment result screens
  - Perform overall tests
- Prototype validation
  - Validate the prototype

5.2.2. Organisation of activities using a Kanban board

In a Kanban board, the work items move from left to right (cf. item 2.5.4).



Backlog
- Design Use Case diagrams
- Specify Use Cases
- Map and sequence system screens
- Design wireframes
- Design class diagram
- Elaborate relational model
- Define prototype development tool
- Set up development environment
- Develop main screen
- Develop questionnaire and related screens
- Develop assessment result screens
- Perform overall tests
- Validate prototype

To Do

In Progress

Done

Figure 32 - Kanban board (Trello)

The Kanban board shows all the stages of the work and is designed according to the needs of the project. It contains a first stage called "Backlog", where all activities planned are initially placed. The following columns are: "To Do", where all the work items ready to be done must be; "In Progress" showing the ongoing work items; and "Done" containing the work already performed. Our Kanban board (cf. Figure 32) has these four stages and was built using the web-based application Trello.

The activity with the highest priority, or which must be performed first, appears on the top of the correspondent column.

5.3. Execution

The Execution phase aims the construction of the prototype, which includes design, modelling, development, and test activities.

Agile design and modelling refer to the techniques that generate artefacts typically light weighted and barely enough to capture the system design without considering many details (Griffiths, 2015). Some types of Agile models that are created during a project include Use Case diagrams; screens design; and data models. In Agile, regardless of the technique, the focus is creating valuable but not extraneous documentation. This was the approach adopted in this study.

5.3.1. User-System interaction design

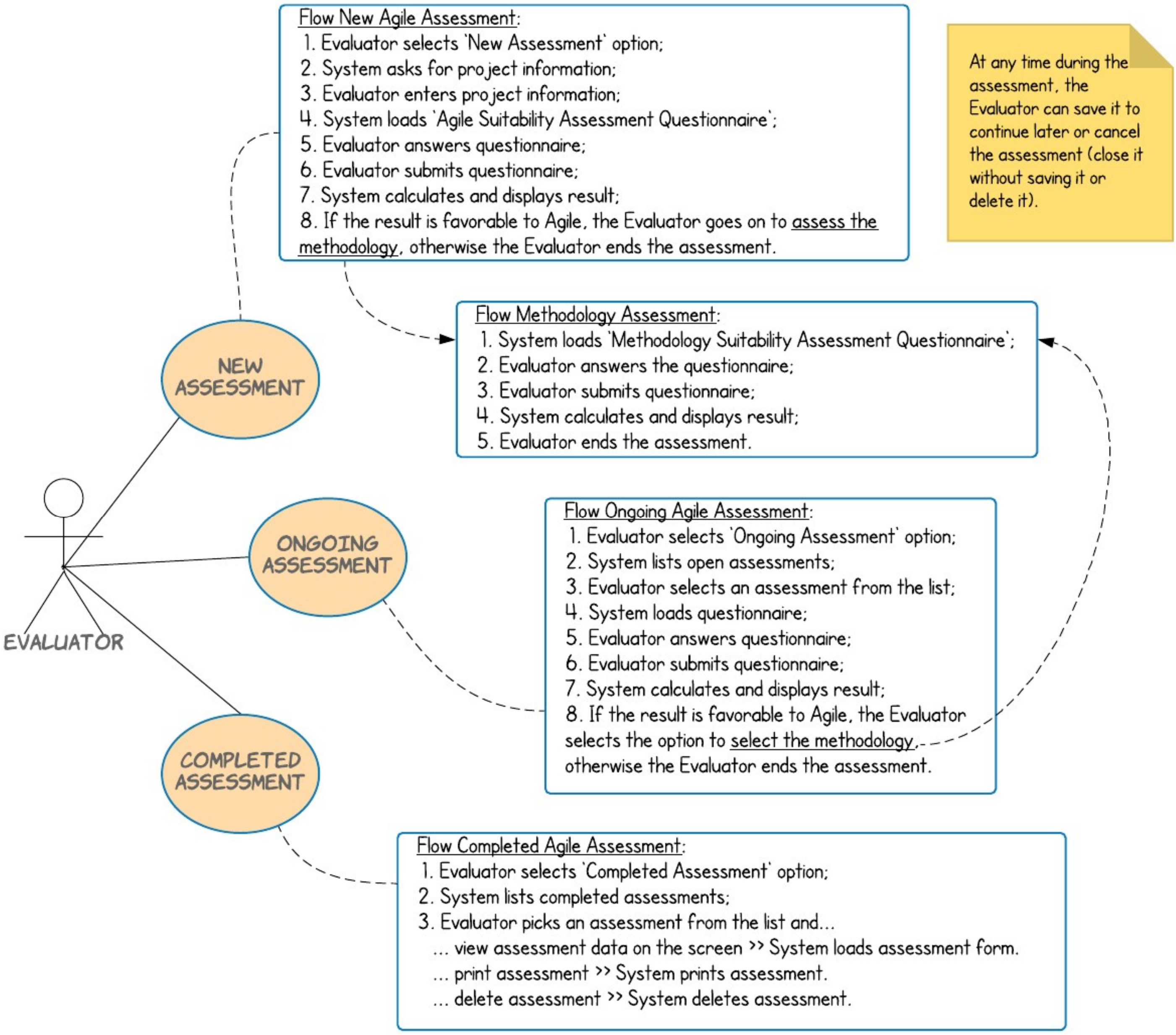
Use Cases describe the interactions between external actors and the system to obtain specific goals (Cockburn, 2001).

Three Use Cases were defined to describe the way the actor (the evaluator) will interact with the system proposed:

- Use Case 1: New Assessment;
- Use Case 2: Ongoing Assessment;
- Use Case 3: Completed Assessment.

To complement the representation of the interaction between the evaluator and the system, we decided to describe the steps involved in the process in the Use Case diagram. The idea is not going into a lot of detail, but instead, depict an overview of the relationship between actor and system with just enough information to support the development (cf. Figure 33).

5.3.1.1. Use Case diagram and specifications



Flow New Agile Assessment:

1. Evaluator selects 'New Assessment' option;
2. System asks for project information;
3. Evaluator enters project information;
4. System loads 'Agile Suitability Assessment Questionnaire';
5. Evaluator answers questionnaire;
6. Evaluator submits questionnaire;
7. System calculates and displays result;
8. If the result is favorable to Agile, the Evaluator goes on to assess the methodology, otherwise the Evaluator ends the assessment.

At any time during the assessment, the Evaluator can save it to continue later or cancel the assessment (close it without saving it or delete it).

Flow Methodology Assessment:

1. System loads 'Methodology Suitability Assessment Questionnaire';
2. Evaluator answers the questionnaire;
3. Evaluator submits questionnaire;
4. System calculates and displays result;
5. Evaluator ends the assessment.

Flow Ongoing Agile Assessment:

1. Evaluator selects 'Ongoing Assessment' option;
2. System lists open assessments;
3. Evaluator selects an assessment from the list;
4. System loads questionnaire;
5. Evaluator answers questionnaire;
6. Evaluator submits questionnaire;
7. System calculates and displays result;
8. If the result is favorable to Agile, the Evaluator selects the option to select the methodology, otherwise the Evaluator ends the assessment.

Flow Completed Agile Assessment:

1. Evaluator selects 'Completed Assessment' option;
2. System lists completed assessments;
3. Evaluator picks an assessment from the list and...

... view assessment data on the screen >> System loads assessment form.
... print assessment >> System prints assessment.
... delete assessment >> System deletes assessment.

Figure 33 - Use Cases - Agile Suitability Assessment

5.3.2. Screens design

5.3.2.1. Screens mapping

The purpose of the mapping is to identify all screens that will be necessary to implement the system's functionalities, as well as to define the sequence of them and the moments in which they must be triggered. For this, we elaborated a flow diagram (cf. Figure 34).

In an Agile project, this type of design is usually done with the help of a whiteboard (a digital picture is taken for recording purposes) or drawn on a sheet of paper. However, given the need for higher formalization required by this work, we decided to elaborate the diagram with the help of the web-based tool Draw.io.

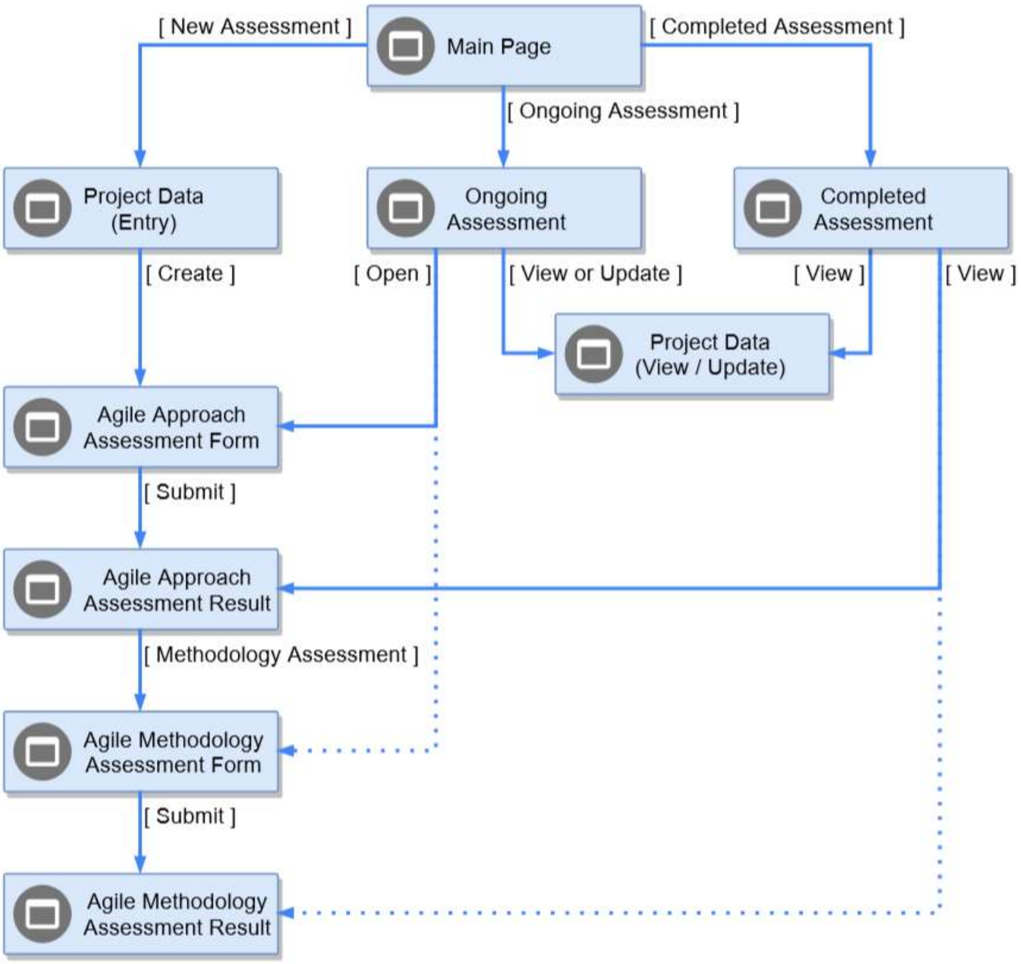


Figure 34 - Screens mapping

Drawing a diagram like this allowed us to have the insights to start producing the sketches of our system screens.

5.3.2.2. Wireframes

Wireframes are a low-fidelity design of system interfaces and screens, conveying just enough information to illustrate their core idea. In the case of an application system, wireframes might show the steps of a process, its functionalities, and behaviours of elements (Brown, 2011).

Just as in the screen mapping, in Agile, wireframes are usually designed using low-tech resources, but for the same reason, here we used an online tool (MockFlow) to draw the system screens sketches (cf. Figures 35 to 43).

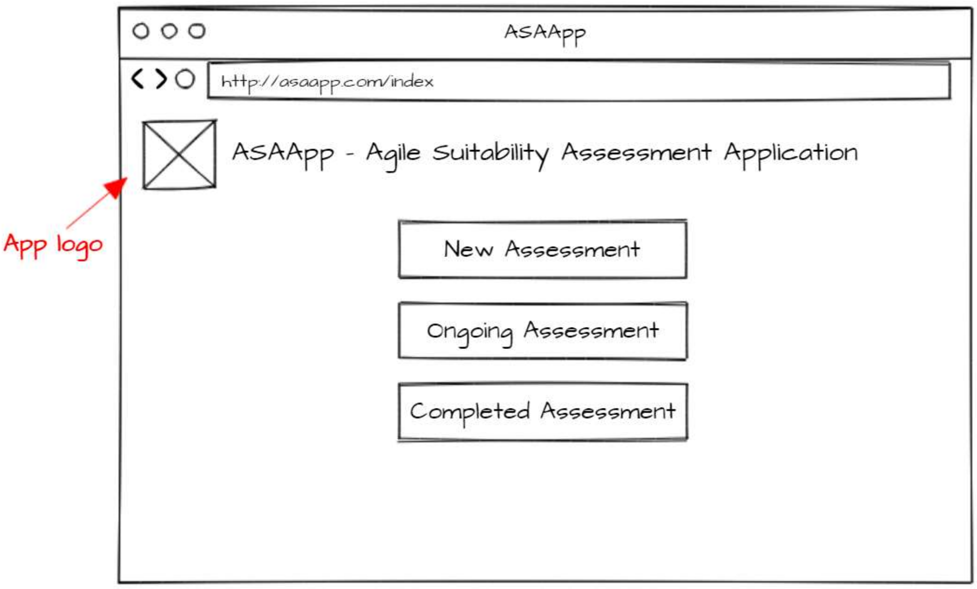


Figure 35 - Main page

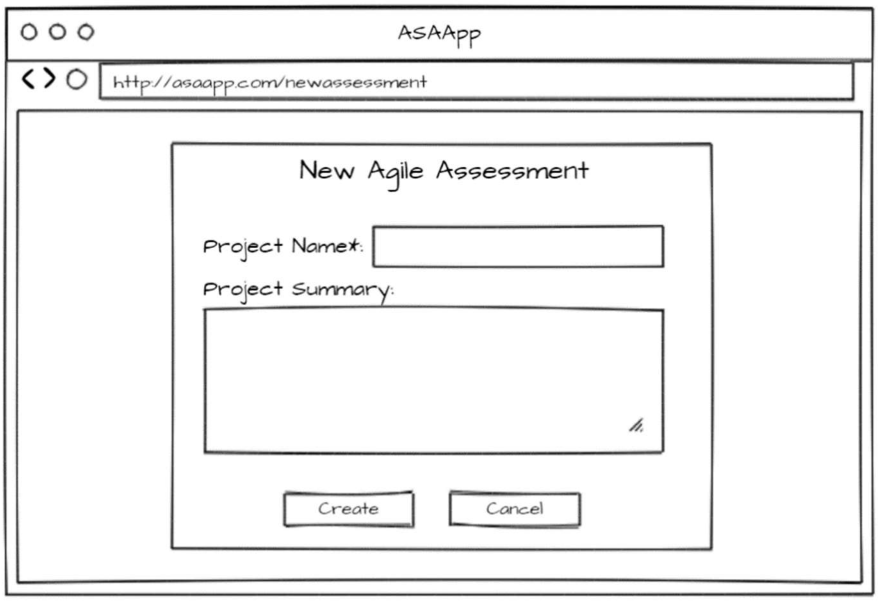
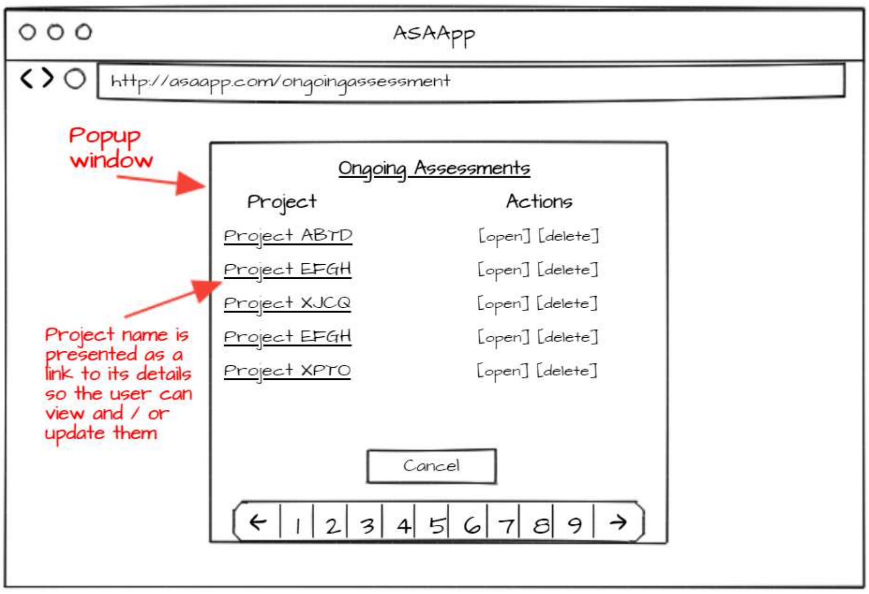


Figure 36 - Project Information (Entry Data)

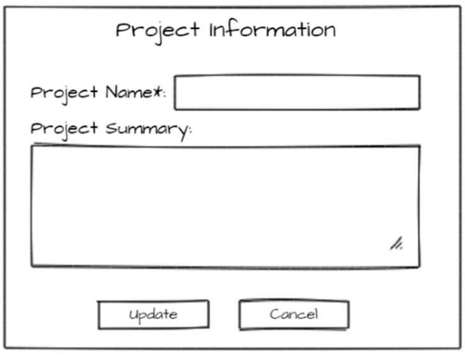


Ongoing Assessments

Project	Actions
Project ABTD	[open] [delete]
Project EFGH	[open] [delete]
Project XJCQ	[open] [delete]
Project EFGH	[open] [delete]
Project XPTO	[open] [delete]

Project name is presented as a link to its details so the user can view and / or update them

Figure 37 - Ongoing Assessments

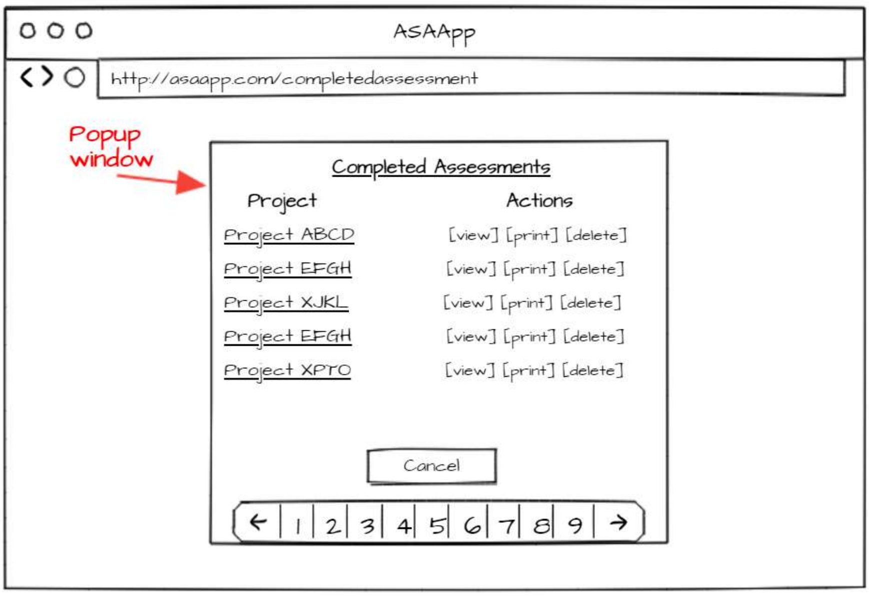


Project Information

Project Name*:

Project Summary:

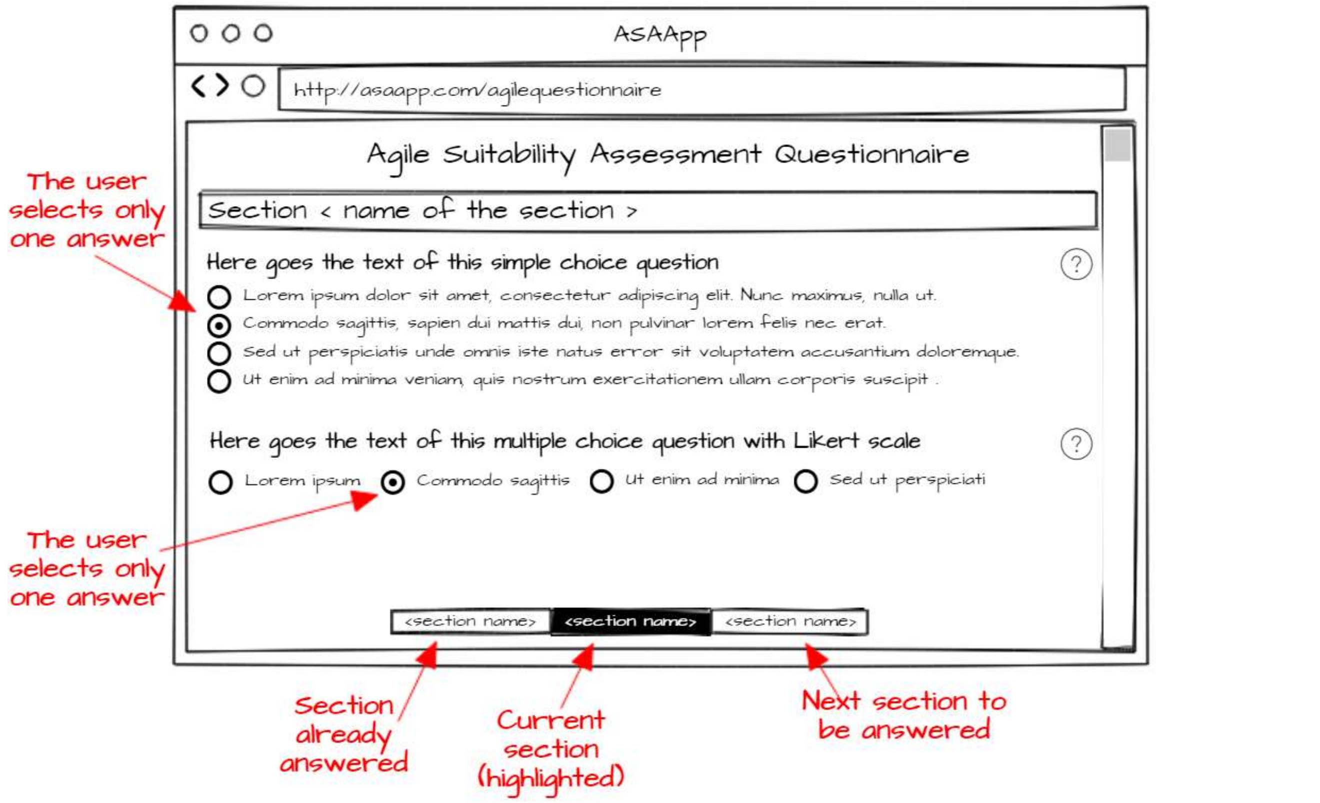
Figure 38 - Project Information (View / Update Data)



Completed Assessments

Project	Actions
Project ABCD	[view] [print] [delete]
Project EFGH	[view] [print] [delete]
Project XJKL	[view] [print] [delete]
Project EFGH	[view] [print] [delete]
Project XPTO	[view] [print] [delete]

Figure 39 - Completed Assessments



Agile Suitability Assessment Questionnaire

Section < name of the section >

Here goes the text of this simple choice question

○ Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut.
● Commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat.
○ Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque.
○ Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit .

Here goes the text of this multiple choice question with Likert scale

○ Lorem ipsum ● Commodo sagittis ○ Ut enim ad minima ○ Sed ut perspiciati

The user selects only one answer

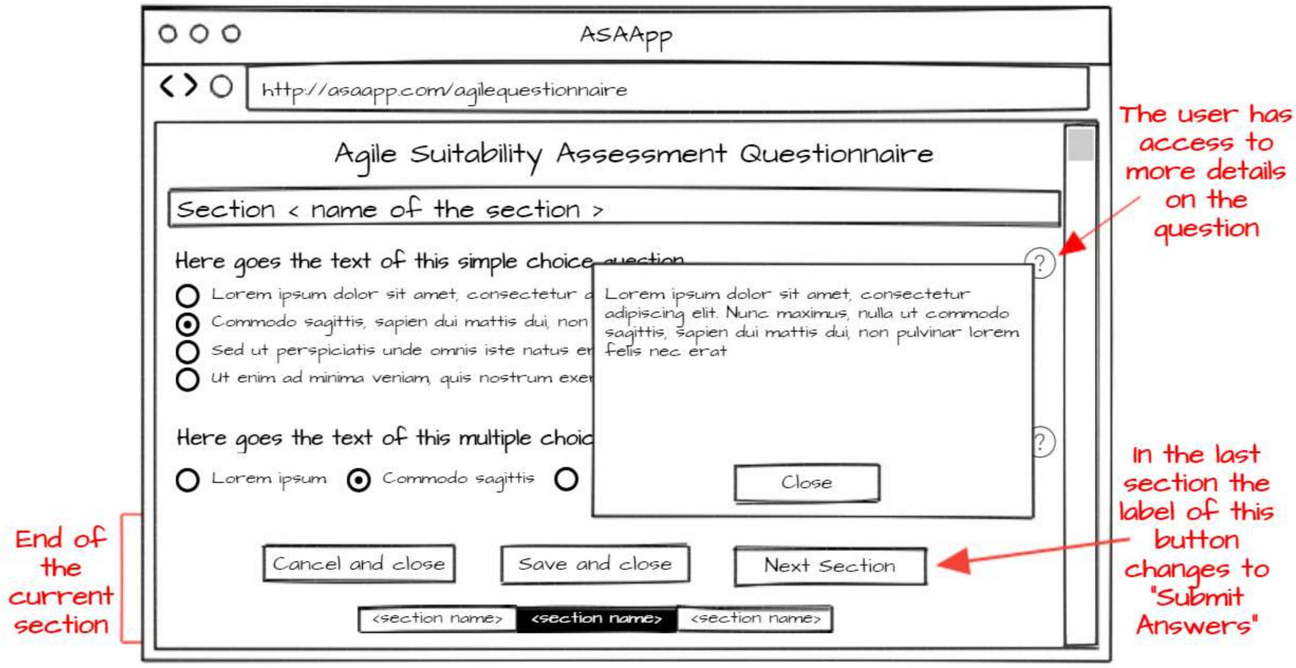
The user selects only one answer

Section already answered

Current section (highlighted)

Next section to be answered

Figure 40 - Agile/Methodology[28] Suitability Assessment Questionnaire (1)



Agile Suitability Assessment Questionnaire

Section < name of the section >

Here goes the text of this simple choice question

○ Lorem ipsum dolor sit amet, consectetur a
● Commodo sagittis, sapien dui mattis dui, non
○ Sed ut perspiciatis unde omnis iste natus er
○ Ut enim ad minima veniam, quis nostrum exe

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Here goes the text of this multiple choic

○ Lorem ipsum ● Commodo sagittis ○

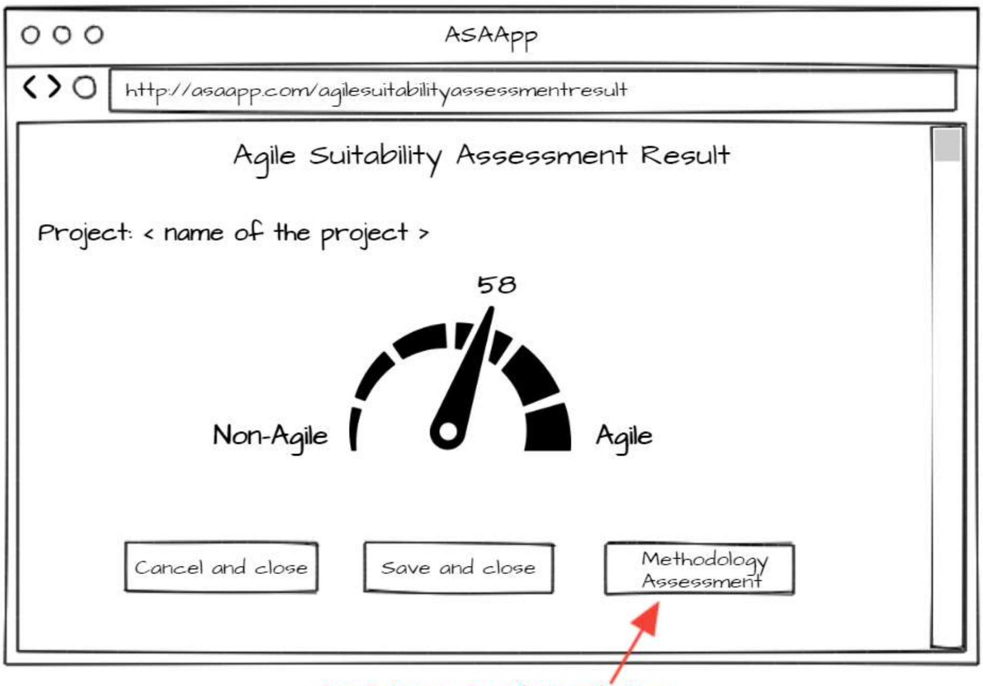
The user has access to more details on the question

In the last section the label of this button changes to "Submit Answers"

End of the current section

Figure 41 - Agile/Methodology Suitability Assessment Questionnaire (2)

[28] The structure of the questionnaire is the same both for assessing the suitability of an Agile approach and assessing the suitability of the Agile methodology.

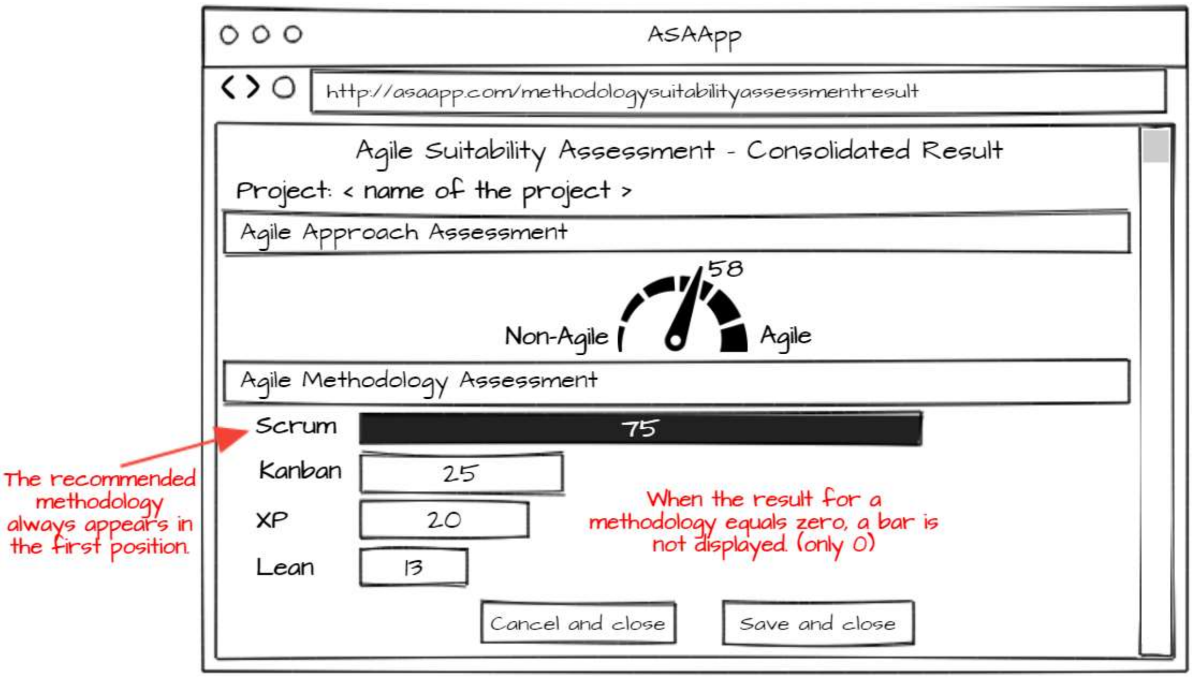


Agile Suitability Assessment Result

Project: < name of the project >

This button is only displayed when the result of the evaluation is equal to or greater than 50

Figure 42 - Agile Suitability Assessment Result



Agile Suitability Assessment - Consolidated Result

Project: < name of the project >

Agile Approach Assessment

Agile Methodology Assessment

Scrum	75
Kanban	25
XP	20
Lean	13

The recommended methodology always appears in the first position.

When the result for a methodology equals zero, a bar is not displayed. (only 0)

Figure 43 - Agile Suitability Consolidated Result

5.3.3. Data modelling

Properly structuring information is an essential condition for the good performance of any information system, whether automated or not (Ramos, 2012). In Agile approaches, this premise is not different. Relational modelling is one of the main forms of structuring information, and to represent this structure, the Unified Modelling Language (UML[29]), created in 1997, is widely used regardless the development approach in place. This section presents the Class Diagram[30] (cf. Figure 44), and the Relational Model[31] (cf. Figure 45) elaborated to structure the information managed by the system.

5.3.3.1. Class diagram

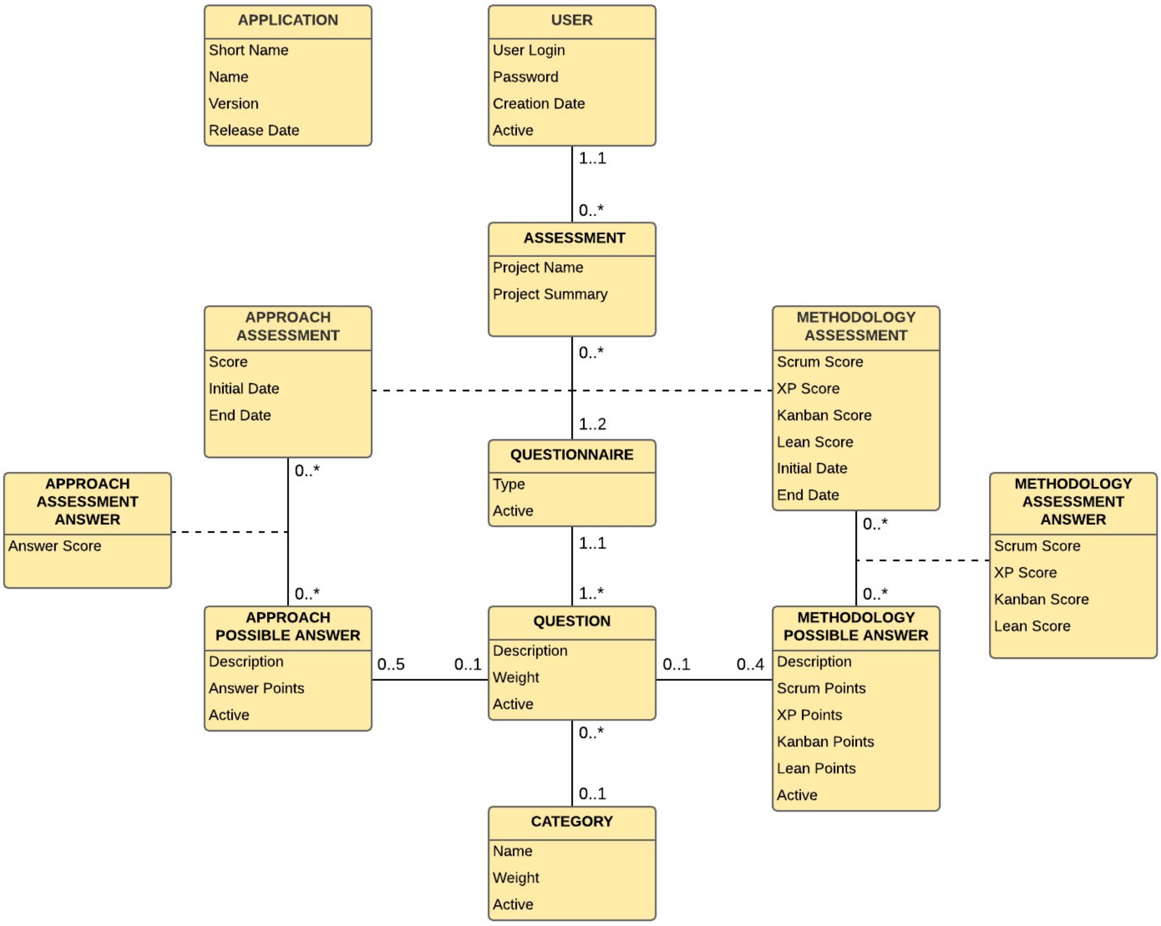


Figure 44 - Class diagram

[29] UML is a diagrammatic language for the design of information systems whose specifications are represented by diagrams that use a set of graphic symbols (Ramos, 2012).

[30] A Class Diagram is a UML diagram representing the structural component that supports an information system (Ramos, 2012).

[31] In a Relational Model information is structured in relationships and represented by tables (Ramos, 2012).

5.3.3.2. Relational model

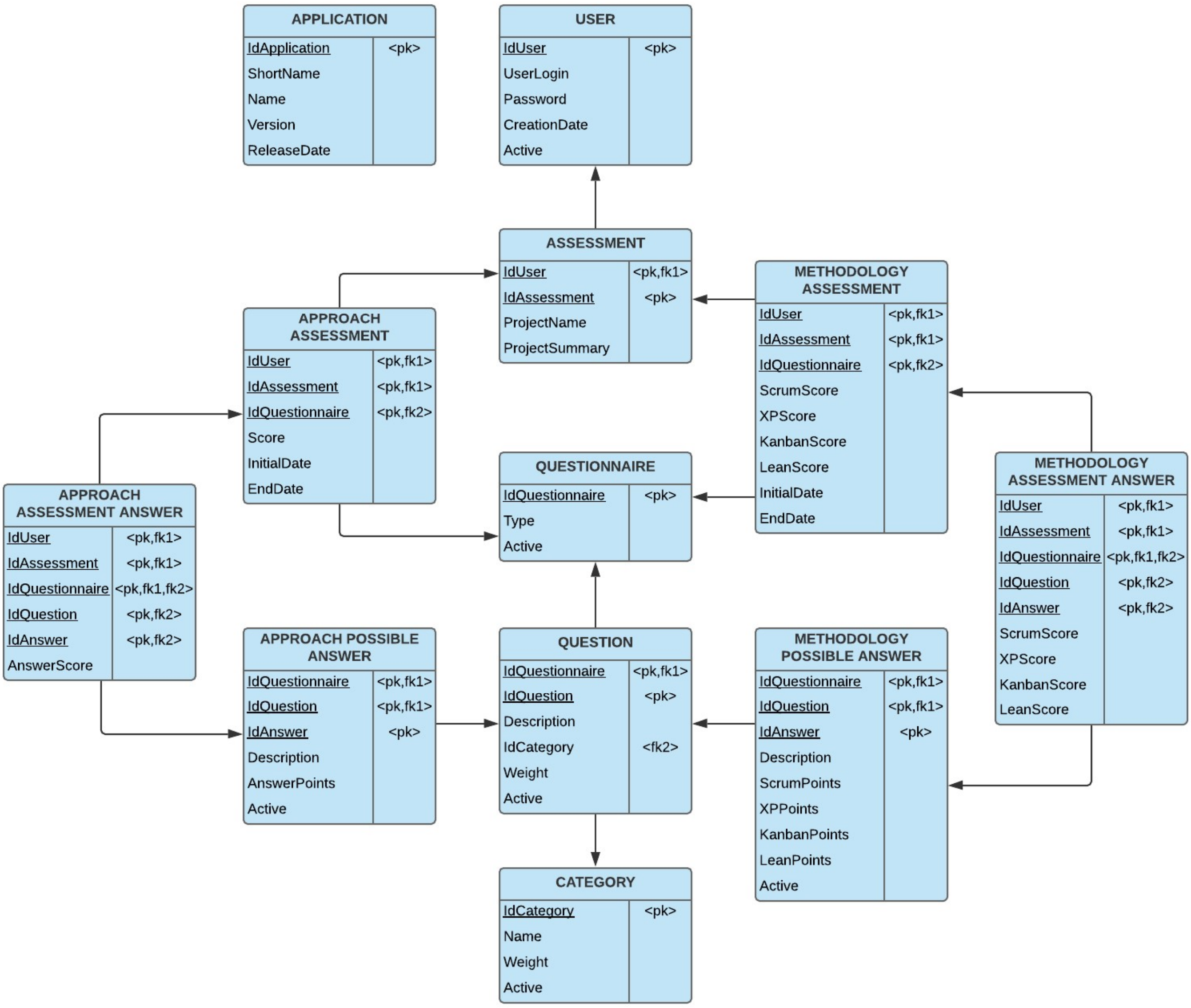


Figure 45 - Relational model

Although Agile projects do not ask for much formalization to model systems data, due to the rigor required by a master's dissertation, we decided to apply formal rules for data modelling, both to elaborate the Class Diagram and the Relational Model. We followed the UML conventions for data modelling and diagramming according to the book "Desenhar Bases de Dados com UML" (Ramos, 2012).

Next, the 12 tables modelled to support the information managed and maintained by the system are briefly explained (cf. Table 22).

Table 22 - System tables

Table name	Table description
1.APPLICATION	It stores the Application (System) identification, version, and release date.
2.USER	It stores the system user data (system access data and creation date).
3.QUESTIONNAIRE	It stores the types of questionnaires available for the Agile suitability assessment. There are only two types: the Agile Approach Suitability Questionnaire and the Agile Methodology Suitability Questionnaire. The table allows the storage of different versions of these questionnaires, but only one questionnaire of each type can be active at the same time (column Active = 'True'). Earlier versions can be stored but disabled (column Active = 'False').
4.QUESTION	It stores the data related to the questions, sub-questions, and statements that are part of the questionnaire.
5.CATEGORY	It stores the data of the evaluation categories of the Agile Approach Suitability Questionnaire. There is no Category concept in the Agile Methodology Suitability Questionnaire.
6.APPROACH POSSIBLE ANSWER	It stores all possible answers for each assessment item in the Agile Approach Suitability Questionnaire.
7.METHODOLOGY POSSIBLE ANSWER	It stores all possible answers for each assessment item in the Agile Methodology Suitability Questionnaire.
8.ASSESSMENT	It stores the basic data of the evaluated project.
9.APPROACH ASSESSMENT	It stores the score obtained and the start and end dates (and time) of the Agile approach suitability assessment.
10.METHODOLOGY ASSESSMENT	It stores the score obtained by each of the four methodologies and the start and end dates (and time) of the Agile methodology suitability assessment.
11.APPROACH ASSESSMENT ANSWER	It stores the answer chosen for a specific assessment item of the Agile Approach Suitability Questionnaire.
12.METHODOLOGY ASSESSMENT ANSWER	It stores the answer chosen for a specific assessment item of the Agile Methodology Suitability Questionnaire.

5.3.4. Prototype development

According to Santos (2018), a software prototype is a model, a simulation of the final product, representing an idea of how it should look and behave. The prototype can demonstrate visual aspects of the interface (hierarchy and organisation of data, the position of elements), navigation (path through which screens or system functionalities will be accessed) and interactions (how the system will respond to each user action). The author explains that prototypes can be classified by the level of fidelity in low, medium, or high.

A low-fidelity prototype is the one that looks like a draft, presenting only sketches of the system screens. There is a low concern with visual aspects, colours, or typography since the objective is to have an idea of the general layout (elements and spaces where they will be placed).

A medium-fidelity prototype has a slightly higher visual refinement but still does not show the final layout of the screens. It can be done in black and white, without even depicting the final images and typography.

A high-fidelity prototype shows the closest visual representation of how the system screens should look after implemented. The visual aspects of the interface (visual identity, style guide) are usually already defined, and it is possible to navigate through the features (at least at demo level).

Since our main objective in developing the prototype of the system is to validate the proposed solution (usefulness, usability, clarity, and intuitiveness of functions), we chose to build a high-fidelity prototype.

5.3.4.1. Prototype scope

To define what would be in the scope of our high-fidelity prototype, we used as a reference the User Stories chosen to compose the first version of the system (or MVP). Therefore, we selected the User Stories that allow the construction of a prototype containing the necessary resources to validate the proposed solution. Table 23 identifies the Stories selected to be prototyped.

Table 23 - User Stories for the prototype

User Story Title	Prototyped
Agile suitability assessment questionnaire	Yes
Different types of answers to the questionnaire's questions	Yes
Questions with specific weights in the questionnaire	Yes
Weight of the questions hidden in the questionnaire	Yes
Saving a partial assessment	No
The assessment must consider the Management perspective	Yes
The assessment must consider the Solution Delivery perspective	Yes
Showing the score of Agile adherences to the evaluated context	Yes
The system points out Scrum as the most suitable methodology	Yes
The system points out XP as the most suitable methodology	No
The system points out Kanban as the most suitable methodology	No
The system points out LSD as the most suitable methodology	No
Viewing the assessment result on the system screen	Yes
Deleting a previous assessment	No
Viewing a previous assessment	Yes

5.3.4.2. Prototype development tool

For the development of the prototype, we chose to use "Angular 7"[32]. Angular 7 is a JavaScript-based front-end development framework (open-source) that allows the creation of Single-Page Applications[33] (SPAs). The framework consists of several components that form a tree structure with parent and child components.

Although Angular 7 is not widely used as a tool for building prototypes, we opted for it since it has some advantages aligned with the objectives and needs of this work, such as:

- Reusable code – Since the idea is to continue this work and develop the system proposed (but not in the scope of this dissertation), using Angular 7 to build the prototype will allow code reuse in the future;
- Comprehensive component library – The framework offers a library with pre-built user interface components, standardizing the layout and operation of some features, facilitating the prototype development;
- Supportive community – Angular 7 has a supportive community of users that offers websites maintained by developers containing up-to-date information concerning tips for all-level programmers and problem-solving, as well as insightful comments, answers, and links to different Angular 7 resources.

As a source code editor tool – to view, edit, run, and debug source code for applications –, we used the open-source Integrated Development Environment (IDE)[34] "Visual Studio Code"[35]. It is a fast and lightweight source code editor extensively used by front-end developers. The Visual Studio Code editor supports the Angular 7 framework.

5.3.4.3. Prototype

In the presentation of our prototype, we describe the sequence of steps of the screens developed.

After the user securely logs in to the system (which we consider unnecessary to represent in the prototype as it is a prerequisite for this type

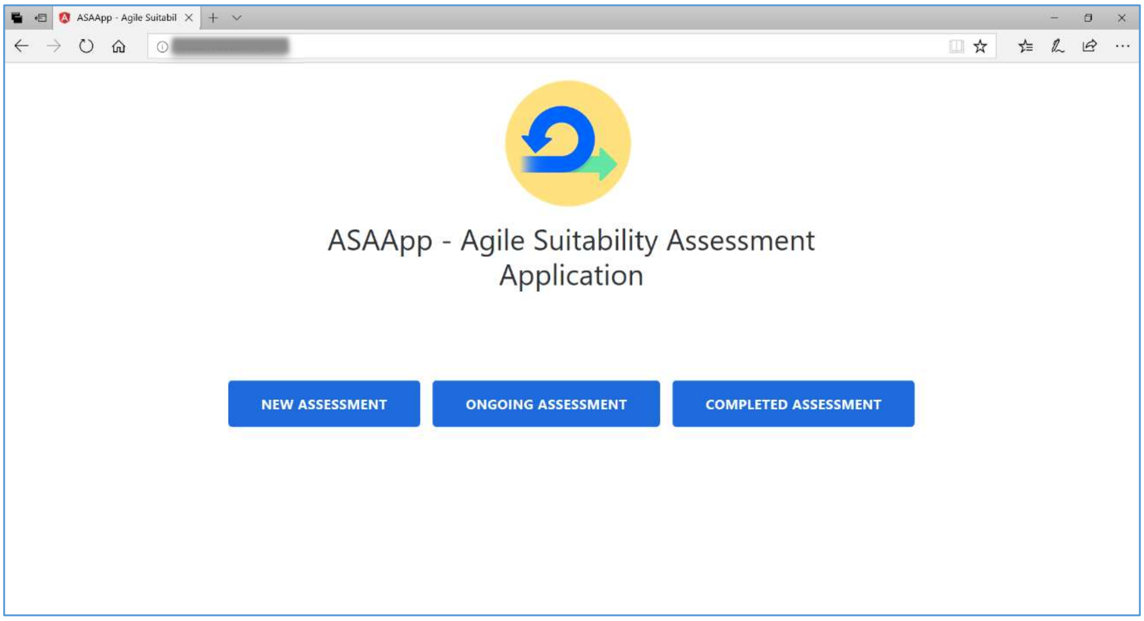
[32] Available in *https://angular.io*

[33] A SPA is an individual component that can be replaced or updated independently, without refreshing the whole page so that it does not need to be reloaded on each user action (Jadhav et al., 2015).

[34] An IDE (Integrated Development Environment) consists of both a complete and unified development methodology and set of computer aids that support the use of the methodology (Konsynski et al., 1984).

[35] Available in *https://code.visualstudio.com*

of solution), the system will display the main page (cf. Figure 46), where the evaluator chooses an action: to perform a "New Assessment"; to retrieve an "Ongoing Assessment"; or to consult a "Completed Assessment".

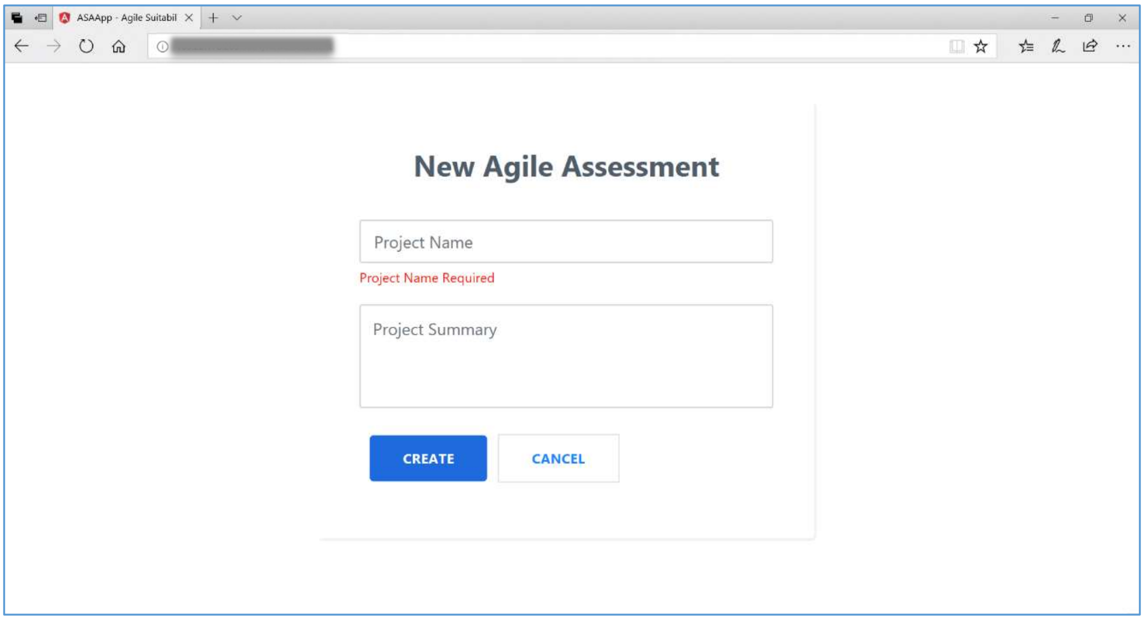


ASAApp - Agile Suitability Assessment Application

Figure 46 - Main page

Option: New Assessment

When opting for a "New Assessment", the system will show a page (cf. Figure 47) where the data of the project to be evaluated must be entered.



New Agile Assessment

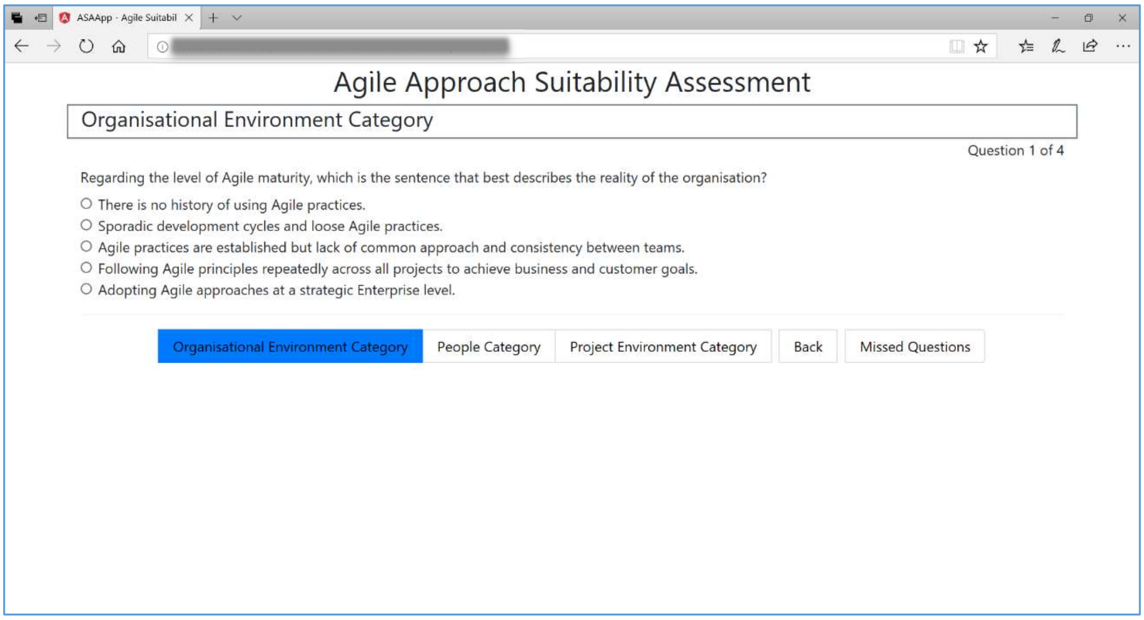
Project Name

Project Name Required

Project Summary

Figure 47 - Project data entry page

After entering and confirming the project data by clicking on the button "Create", a page containing the "Agile Approach Suitability Assessment Questionnaire" will be displayed with the first question (cf. Figure 48).



Agile Approach Suitability Assessment

Organisational Environment Category

Question 1 of 4

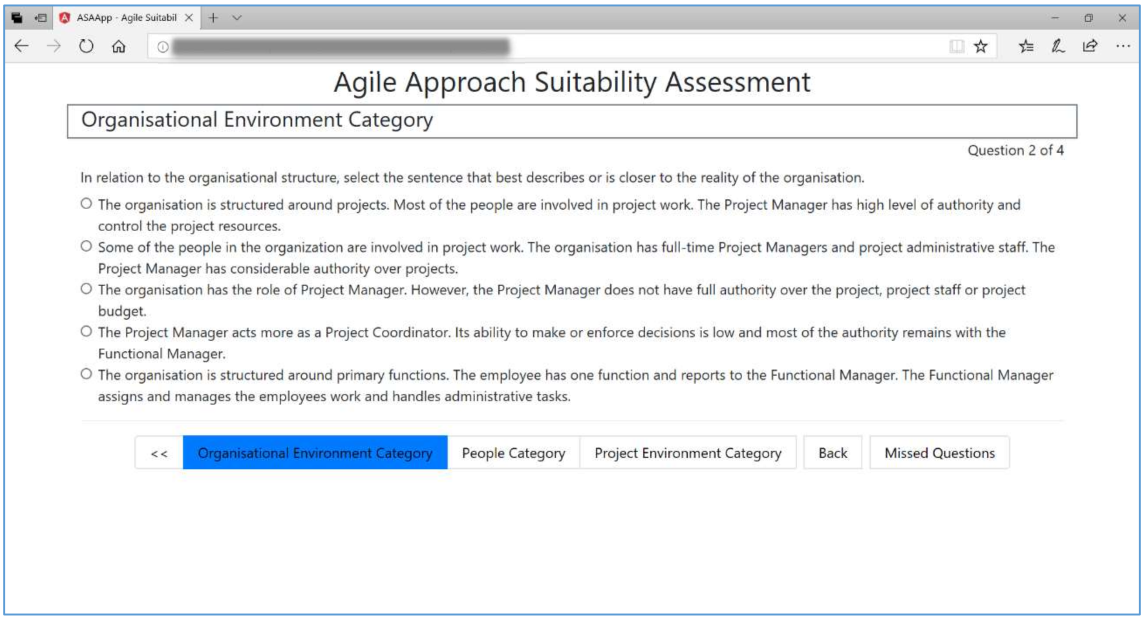
Regarding the level of Agile maturity, which is the sentence that best describes the reality of the organisation?

○ There is no history of using Agile practices.
○ Sporadic development cycles and loose Agile practices.
○ Agile practices are established but lack of common approach and consistency between teams.
○ Following Agile principles repeatedly across all projects to achieve business and customer goals.
○ Adopting Agile approaches at a strategic Enterprise level.

Figure 48 - Questionnaire page - Organisational Environment (question 1 of 4)

The evaluator may start completing the form. Each question of each category of the first part of the assessment is presented in sequence until the last section of the questionnaire (cf. Figures 49 to 56).

<u>Note</u>: Only after selecting the answer, the system will display the button >> so that the user can navigate to the following question.



Agile Approach Suitability Assessment

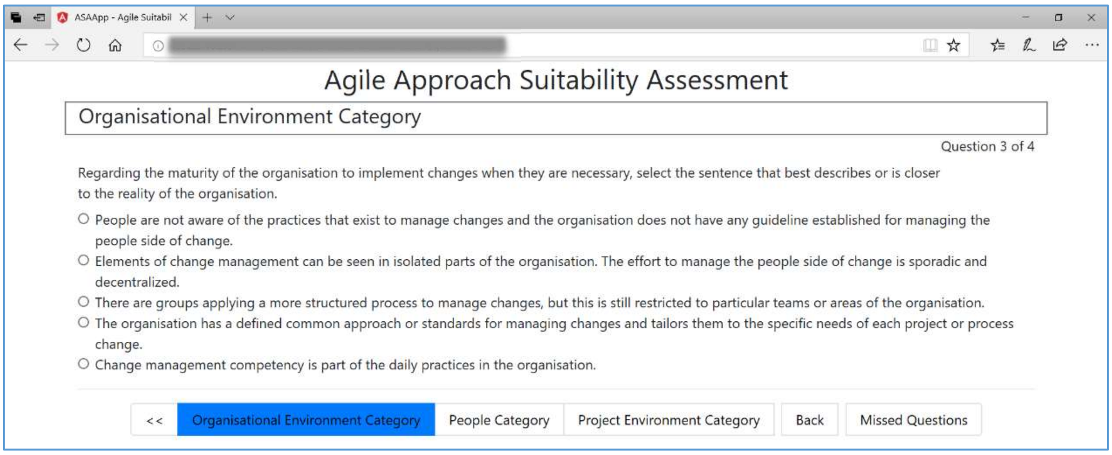
Organisational Environment Category

Question 2 of 4

In relation to the organisational structure, select the sentence that best describes or is closer to the reality of the organisation.

○ The organisation is structured around projects. Most of the people are involved in project work. The Project Manager has high level of authority and control the project resources.
○ Some of the people in the organization are involved in project work. The organisation has full-time Project Managers and project administrative staff. The Project Manager has considerable authority over projects.
○ The organisation has the role of Project Manager. However, the Project Manager does not have full authority over the project, project staff or project budget.
○ The Project Manager acts more as a Project Coordinator. Its ability to make or enforce decisions is low and most of the authority remains with the Functional Manager.
○ The organisation is structured around primary functions. The employee has one function and reports to the Functional Manager. The Functional Manager assigns and manages the employees work and handles administrative tasks.

Figure 49 - Questionnaire page - Organisational Environment (question 2 of 4)



[redacted]

Agile Approach Suitability Assessment

Organisational Environment Category

Question 3 of 4

Regarding the maturity of the organisation to implement changes when they are necessary, select the sentence that best describes or is closer to the reality of the organisation.

○ People are not aware of the practices that exist to manage changes and the organisation does not have any guideline established for managing the people side of change.

○ Elements of change management can be seen in isolated parts of the organisation. The effort to manage the people side of change is sporadic and decentralized.

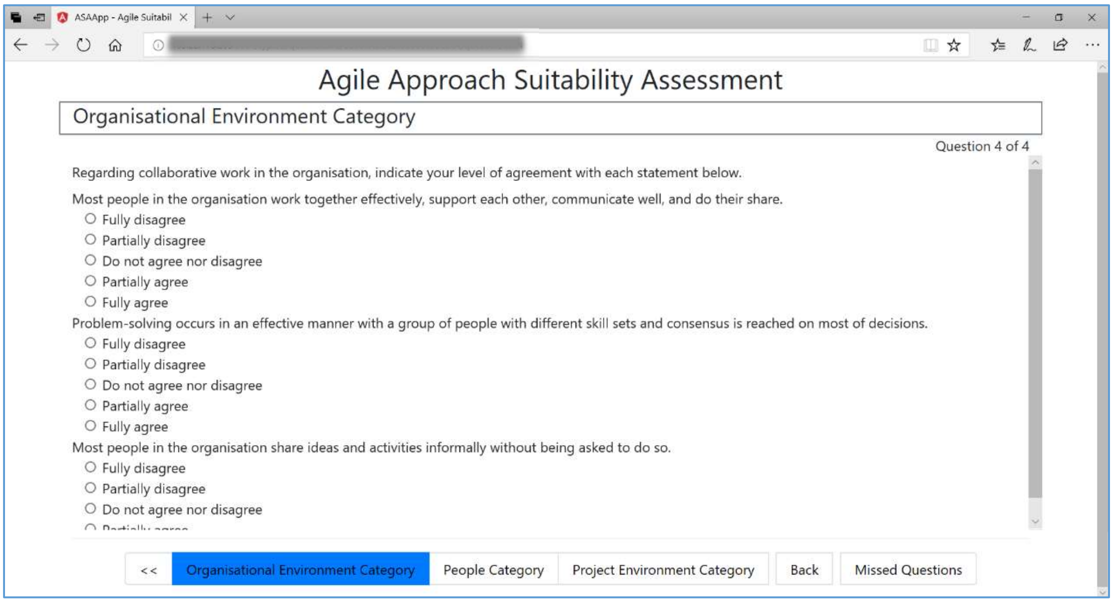
○ There are groups applying a more structured process to manage changes, but this is still restricted to particular teams or areas of the organisation.

○ The organisation has a defined common approach or standards for managing changes and tailors them to the specific needs of each project or process change.

○ Change management competency is part of the daily practices in the organisation.

<<
Organisational Environment Category
People Category
Project Environment Category
Back
Missed Questions

Figure 50 - Questionnaire page - Organisational Environment (question 3 of 4)



[redacted]

Agile Approach Suitability Assessment

Organisational Environment Category

Question 4 of 4

Regarding collaborative work in the organisation, indicate your level of agreement with each statement below.

Most people in the organisation work together effectively, support each other, communicate well, and do their share.

○ Fully disagree
○ Partially disagree
○ Do not agree nor disagree
○ Partially agree
○ Fully agree

Problem-solving occurs in an effective manner with a group of people with different skill sets and consensus is reached on most of decisions.

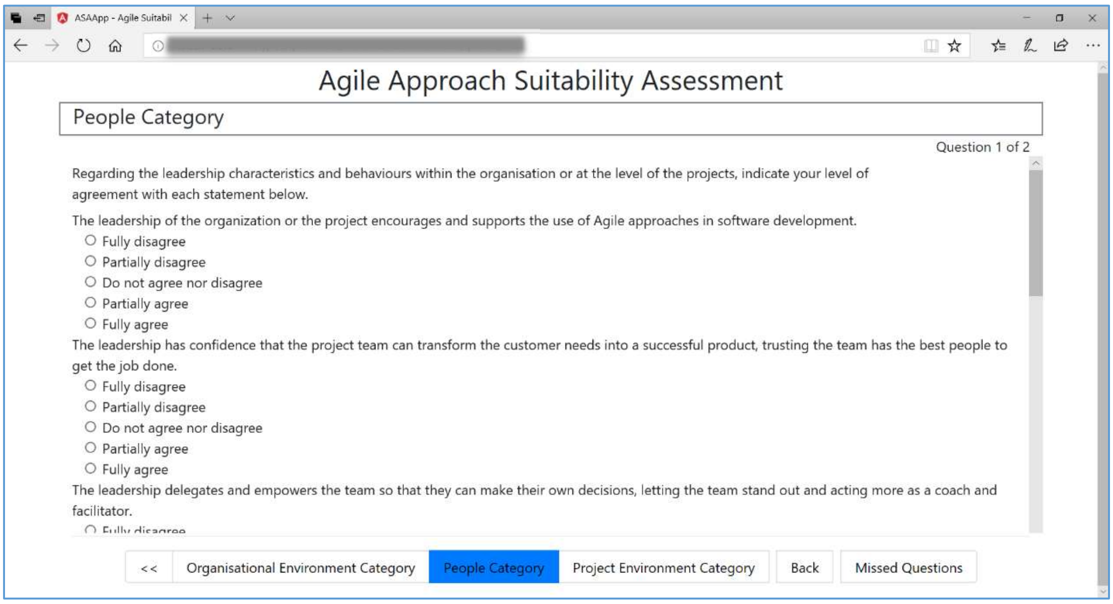
○ Fully disagree
○ Partially disagree
○ Do not agree nor disagree
○ Partially agree
○ Fully agree

Most people in the organisation share ideas and activities informally without being asked to do so.

○ Fully disagree
○ Partially disagree
○ Do not agree nor disagree
○ Partially agree

<<
Organisational Environment Category
People Category
Project Environment Category
Back
Missed Questions

Figure 51 - Questionnaire page - Organisational Environment (question 4 of 4)



[redacted]

Agile Approach Suitability Assessment

People Category

Question 1 of 2

Regarding the leadership characteristics and behaviours within the organisation or at the level of the projects, indicate your level of agreement with each statement below.

The leadership of the organization or the project encourages and supports the use of Agile approaches in software development.

○ Fully disagree
○ Partially disagree
○ Do not agree nor disagree
○ Partially agree
○ Fully agree

The leadership has confidence that the project team can transform the customer needs into a successful product, trusting the team has the best people to get the job done.

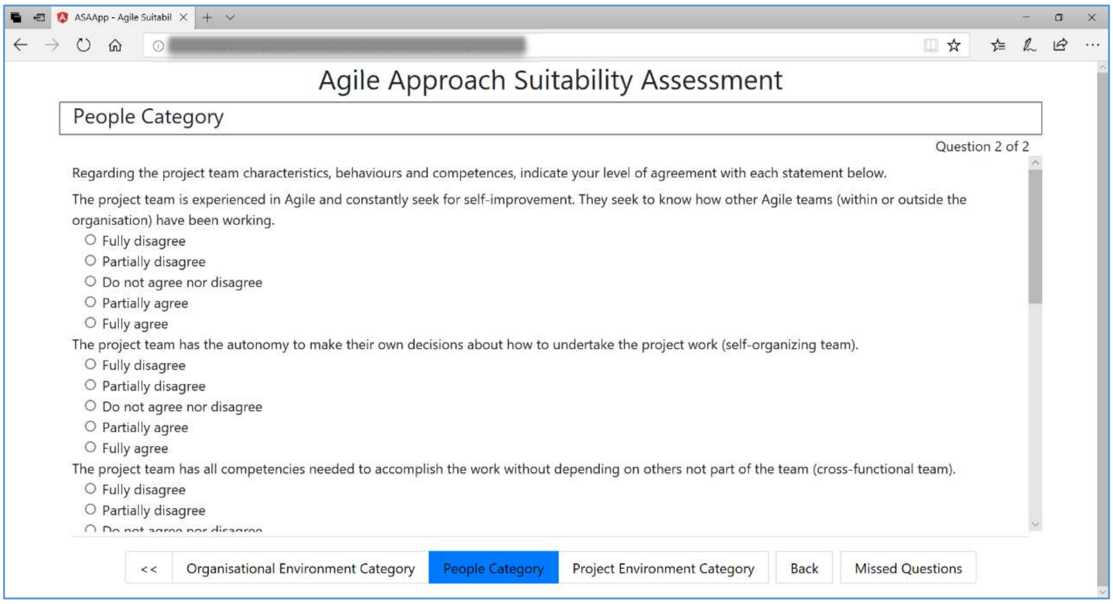
○ Fully disagree
○ Partially disagree
○ Do not agree nor disagree
○ Partially agree
○ Fully agree

The leadership delegates and empowers the team so that they can make their own decisions, letting the team stand out and acting more as a coach and facilitator.

○ Fully disagree

<<
Organisational Environment Category
People Category
Project Environment Category
Back
Missed Questions

Figure 52 - Questionnaire page - People (question 1 of 2)



Agile Approach Suitability Assessment

People Category

Question 2 of 2

Regarding the project team characteristics, behaviours and competences, indicate your level of agreement with each statement below.

The project team is experienced in Agile and constantly seek for self-improvement. They seek to know how other Agile teams (within or outside the organisation) have been working.

○ Fully disagree
○ Partially disagree
○ Do not agree nor disagree
○ Partially agree
○ Fully agree

The project team has the autonomy to make their own decisions about how to undertake the project work (self-organizing team).

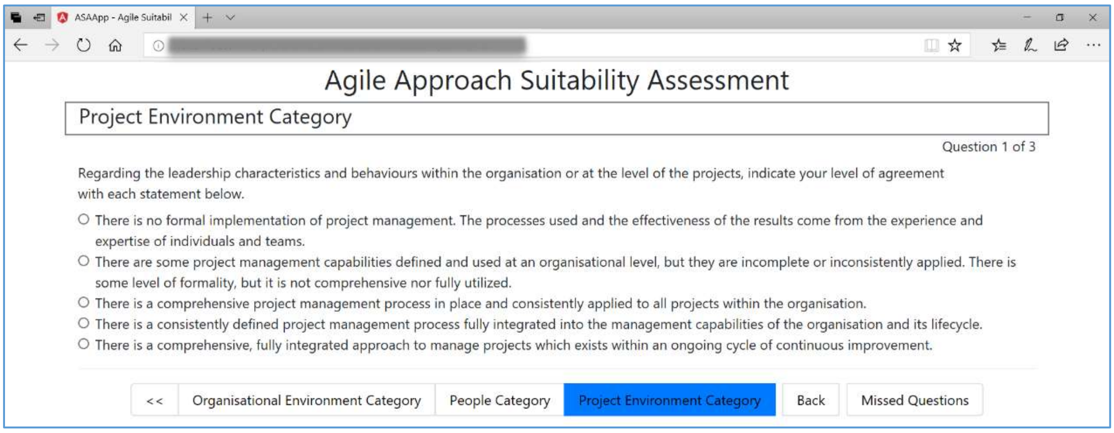
○ Fully disagree
○ Partially disagree
○ Do not agree nor disagree
○ Partially agree
○ Fully agree

The project team has all competencies needed to accomplish the work without depending on others not part of the team (cross-functional team).

○ Fully disagree
○ Partially disagree
○ Do not agree nor disagree

<<
Organisational Environment Category
People Category
Project Environment Category
Back
Missed Questions

Figure 53 - Questionnaire page - People (question 2 of 2)



Agile Approach Suitability Assessment

Project Environment Category

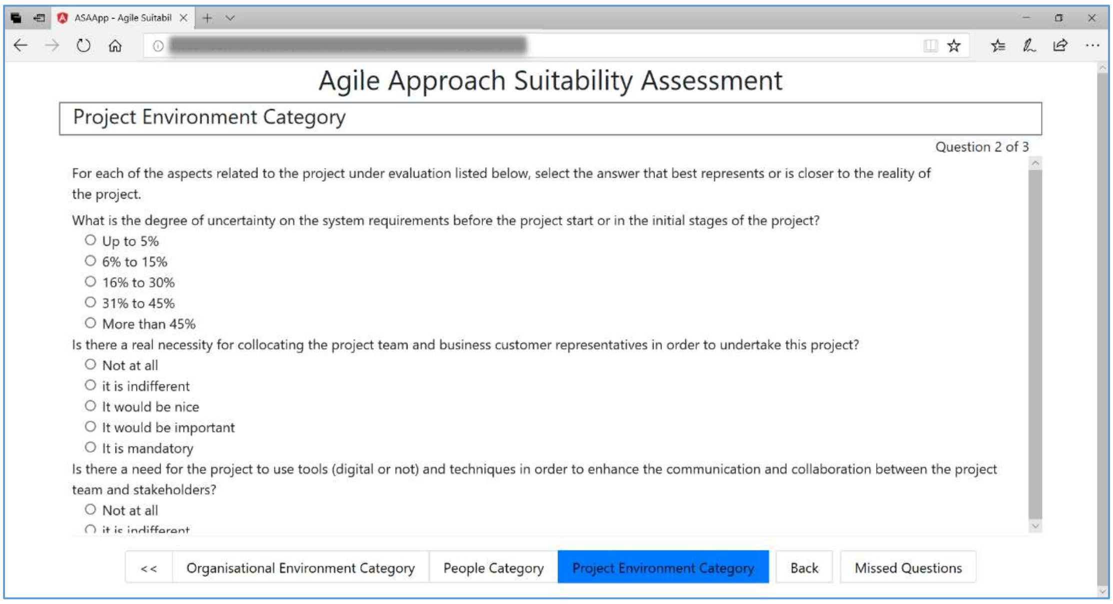
Question 1 of 3

Regarding the leadership characteristics and behaviours within the organisation or at the level of the projects, indicate your level of agreement with each statement below.

○ There is no formal implementation of project management. The processes used and the effectiveness of the results come from the experience and expertise of individuals and teams.
○ There are some project management capabilities defined and used at an organisational level, but they are incomplete or inconsistently applied. There is some level of formality, but it is not comprehensive nor fully utilized.
○ There is a comprehensive project management process in place and consistently applied to all projects within the organisation.
○ There is a consistently defined project management process fully integrated into the management capabilities of the organisation and its lifecycle.
○ There is a comprehensive, fully integrated approach to manage projects which exists within an ongoing cycle of continuous improvement.

<<
Organisational Environment Category
People Category
Project Environment Category
Back
Missed Questions

Figure 54 - Questionnaire page - Project Environment (question 1 of 3)



Agile Approach Suitability Assessment

Project Environment Category

Question 2 of 3

For each of the aspects related to the project under evaluation listed below, select the answer that best represents or is closer to the reality of the project.

What is the degree of uncertainty on the system requirements before the project start or in the initial stages of the project?

○ Up to 5%
○ 6% to 15%
○ 16% to 30%
○ 31% to 45%
○ More than 45%

Is there a real necessity for collocating the project team and business customer representatives in order to undertake this project?

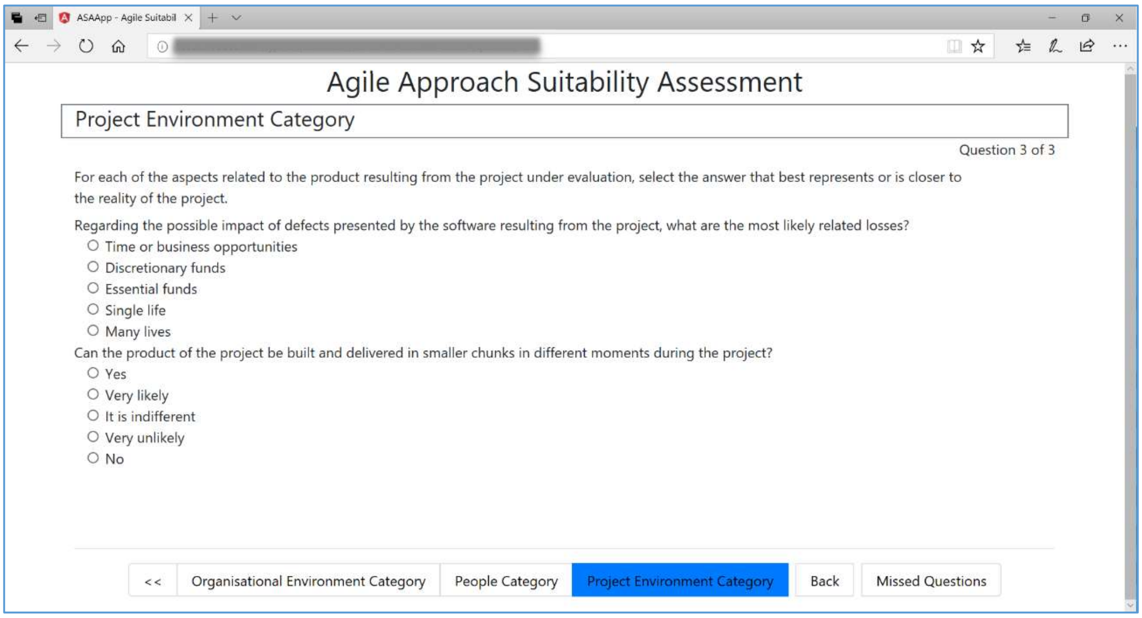
○ Not at all
○ it is indifferent
○ It would be nice
○ It would be important
○ It is mandatory

Is there a need for the project to use tools (digital or not) and techniques in order to enhance the communication and collaboration between the project team and stakeholders?

○ Not at all
○ it is indifferent

<<
Organisational Environment Category
People Category
Project Environment Category
Back
Missed Questions

Figure 55 - Questionnaire page - Project Environment (question 2 of 3)



Agile Approach Suitability Assessment

Project Environment Category

Question 3 of 3

For each of the aspects related to the product resulting from the project under evaluation, select the answer that best represents or is closer to the reality of the project.

Regarding the possible impact of defects presented by the software resulting from the project, what are the most likely related losses?

○ Time or business opportunities
○ Discretionary funds
○ Essential funds
○ Single life
○ Many lives

Can the product of the project be built and delivered in smaller chunks in different moments during the project?

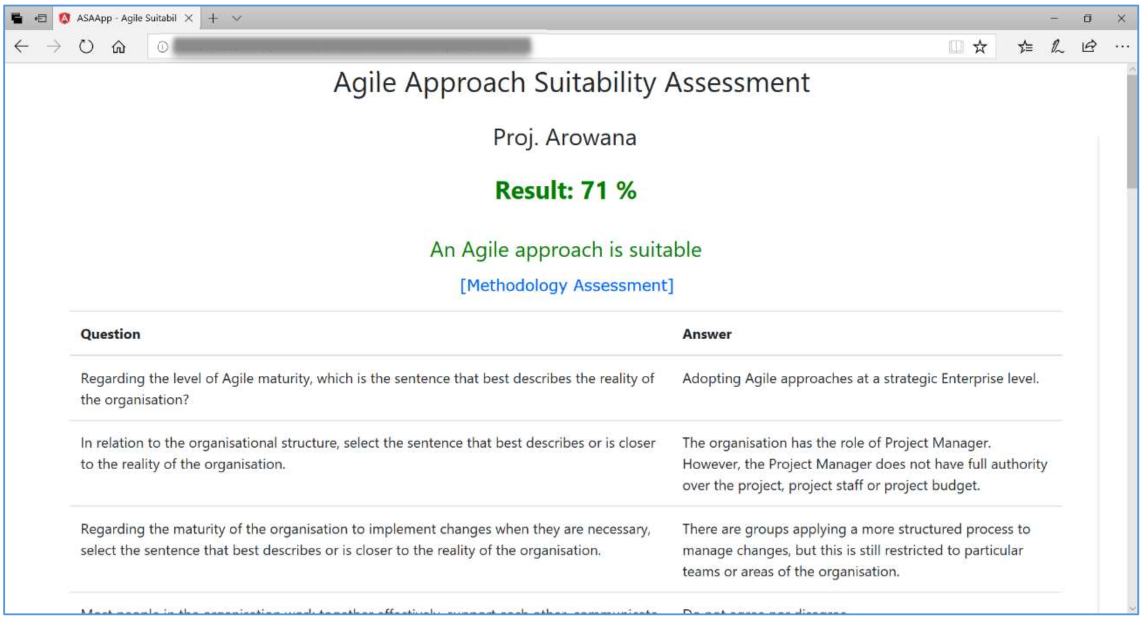
○ Yes
○ Very likely
○ It is indifferent
○ Very unlikely
○ No

<<
Organisational Environment Category
People Category
Project Environment Category
Back
Missed Questions

Figure 56 - Questionnaire page - Project Environment (question 3 of 3)

At any time, the evaluator can interrupt the assessment. The system will automatically save the answers until the point the form has been filled in. The evaluator can return at a later time and continue the assessment through the option "Ongoing Assessment" (on the Main page).

After completing the questionnaire, the system will process and present the result, along with the answers provided (cf. Figure 57).



Agile Approach Suitability Assessment

Proj. Arowana

Result: 71 %

An Agile approach is suitable

[Methodology Assessment]

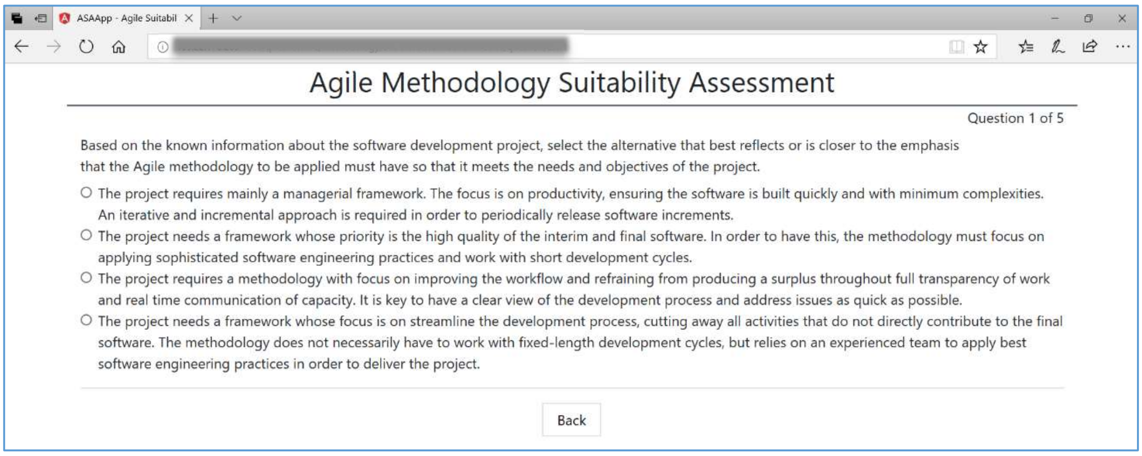
Question	Answer
Regarding the level of Agile maturity, which is the sentence that best describes the reality of the organisation?	Adopting Agile approaches at a strategic Enterprise level.
In relation to the organisational structure, select the sentence that best describes or is closer to the reality of the organisation.	The organisation has the role of Project Manager. However, the Project Manager does not have full authority over the project, project staff or project budget.
Regarding the maturity of the organisation to implement changes when they are necessary, select the sentence that best describes or is closer to the reality of the organisation.	There are groups applying a more structured process to manage changes, but this is still restricted to particular teams or areas of the organisation.
Most people in the organisation work together effectively, support each other, communicate	Do not agree nor disagree

Figure 57 - Agile approach suitability assessment (result page)

If the result is favourable for an Agile approach, the system will give the evaluator the option to proceed to the "Agile Methodology Suitability Assessment Questionnaire", by showing the link "[Methodology

Assessment]". If the result is favourable for a non-Agile approach, the link will not be available, and the evaluator can leave the assessment just closing the window. It will be possible to return to the main page as well by clicking on the button RETURN at the bottom of the page.

By clicking on the link "[Methodology Assessment]", the system will load the page with the first question of the correspondent questionnaire (cf. Figure 58).



Agile Methodology Suitability Assessment

Question 1 of 5

Based on the known information about the software development project, select the alternative that best reflects or is closer to the emphasis that the Agile methodology to be applied must have so that it meets the needs and objectives of the project.

○ The project requires mainly a managerial framework. The focus is on productivity, ensuring the software is built quickly and with minimum complexities. An iterative and incremental approach is required in order to periodically release software increments.

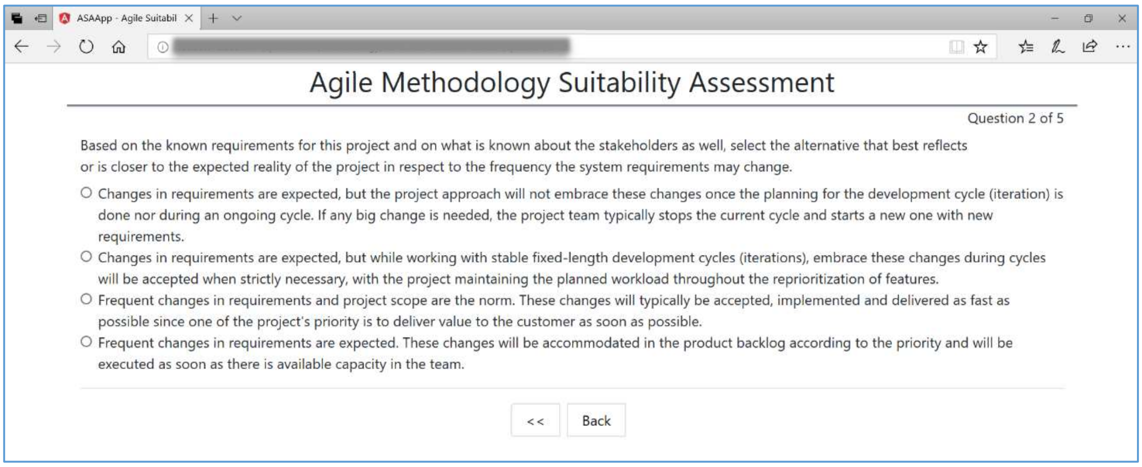
○ The project needs a framework whose priority is the high quality of the interim and final software. In order to have this, the methodology must focus on applying sophisticated software engineering practices and work with short development cycles.

○ The project requires a methodology with focus on improving the workflow and refraining from producing a surplus throughout full transparency of work and real time communication of capacity. It is key to have a clear view of the development process and address issues as quick as possible.

○ The project needs a framework whose focus is on streamline the development process, cutting away all activities that do not directly contribute to the final software. The methodology does not necessarily have to work with fixed-length development cycles, but relies on an experienced team to apply best software engineering practices in order to deliver the project.

Figure 58 - Agile methodology suitability assessment (question 1 of 5)

Like in the previous questionnaire, the questions are presented sequentially (cf. Figures 59 to 62), and at any time the evaluator can interrupt the evaluation and return later to finalize it using the option "Ongoing Assessment".



Agile Methodology Suitability Assessment

Question 2 of 5

Based on the known requirements for this project and on what is known about the stakeholders as well, select the alternative that best reflects or is closer to the expected reality of the project in respect to the frequency the system requirements may change.

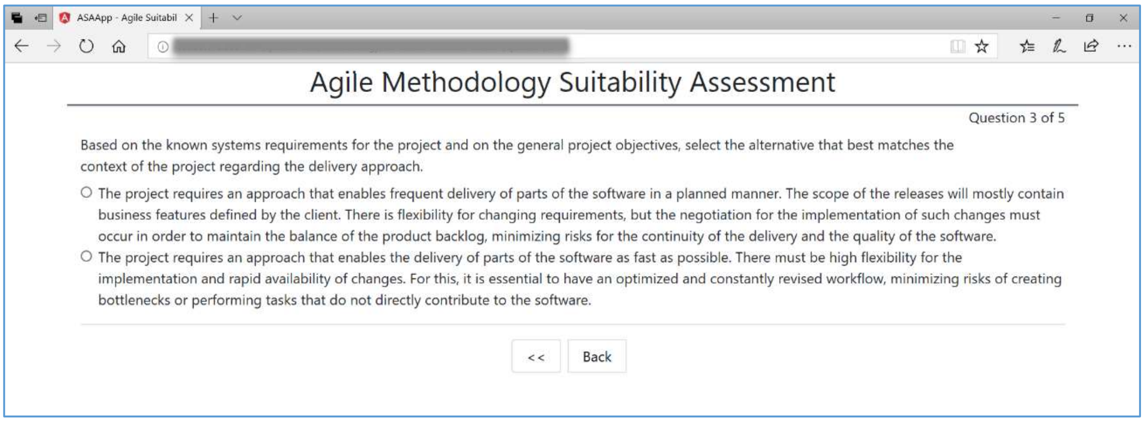
○ Changes in requirements are expected, but the project approach will not embrace these changes once the planning for the development cycle (iteration) is done nor during an ongoing cycle. If any big change is needed, the project team typically stops the current cycle and starts a new one with new requirements.

○ Changes in requirements are expected, but while working with stable fixed-length development cycles (iterations), embrace these changes during cycles will be accepted when strictly necessary, with the project maintaining the planned workload throughout the reprioritization of features.

○ Frequent changes in requirements and project scope are the norm. These changes will typically be accepted, implemented and delivered as fast as possible since one of the project's priority is to deliver value to the customer as soon as possible.

○ Frequent changes in requirements are expected. These changes will be accommodated in the product backlog according to the priority and will be executed as soon as there is available capacity in the team.

Figure 59 - Agile methodology suitability assessment (question 2 of 5)



Agile Methodology Suitability Assessment

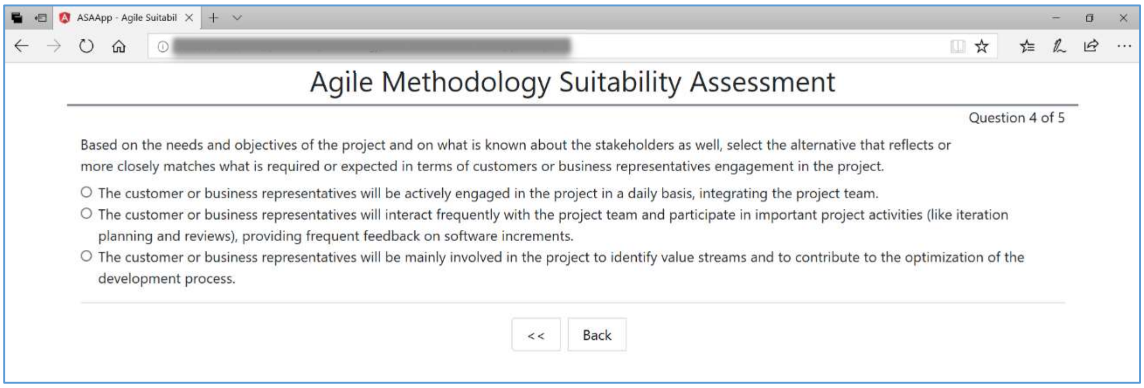
Question 3 of 5

Based on the known systems requirements for the project and on the general project objectives, select the alternative that best matches the context of the project regarding the delivery approach.

○ The project requires an approach that enables frequent delivery of parts of the software in a planned manner. The scope of the releases will mostly contain business features defined by the client. There is flexibility for changing requirements, but the negotiation for the implementation of such changes must occur in order to maintain the balance of the product backlog, minimizing risks for the continuity of the delivery and the quality of the software.

○ The project requires an approach that enables the delivery of parts of the software as fast as possible. There must be high flexibility for the implementation and rapid availability of changes. For this, it is essential to have an optimized and constantly revised workflow, minimizing risks of creating bottlenecks or performing tasks that do not directly contribute to the software.

Figure 60 - Agile methodology suitability assessment (question 3 of 5)



Agile Methodology Suitability Assessment

Question 4 of 5

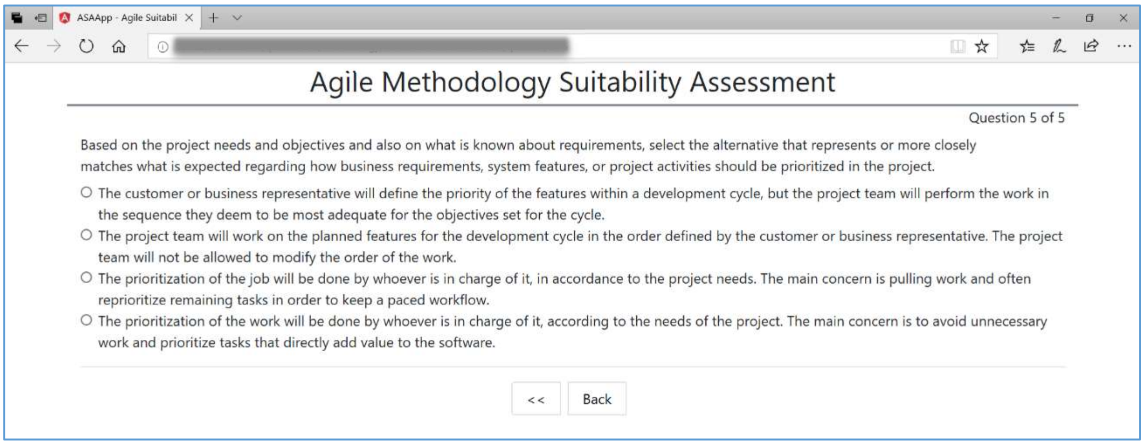
Based on the needs and objectives of the project and on what is known about the stakeholders as well, select the alternative that reflects or more closely matches what is required or expected in terms of customers or business representatives engagement in the project.

○ The customer or business representatives will be actively engaged in the project in a daily basis, integrating the project team.

○ The customer or business representatives will interact frequently with the project team and participate in important project activities (like iteration planning and reviews), providing frequent feedback on software increments.

○ The customer or business representatives will be mainly involved in the project to identify value streams and to contribute to the optimization of the development process.

Figure 61 - Agile methodology suitability assessment (question 4 of 5)



Agile Methodology Suitability Assessment

Question 5 of 5

Based on the project needs and objectives and also on what is known about requirements, select the alternative that represents or more closely matches what is expected regarding how business requirements, system features, or project activities should be prioritized in the project.

○ The customer or business representative will define the priority of the features within a development cycle, but the project team will perform the work in the sequence they deem to be most adequate for the objectives set for the cycle.

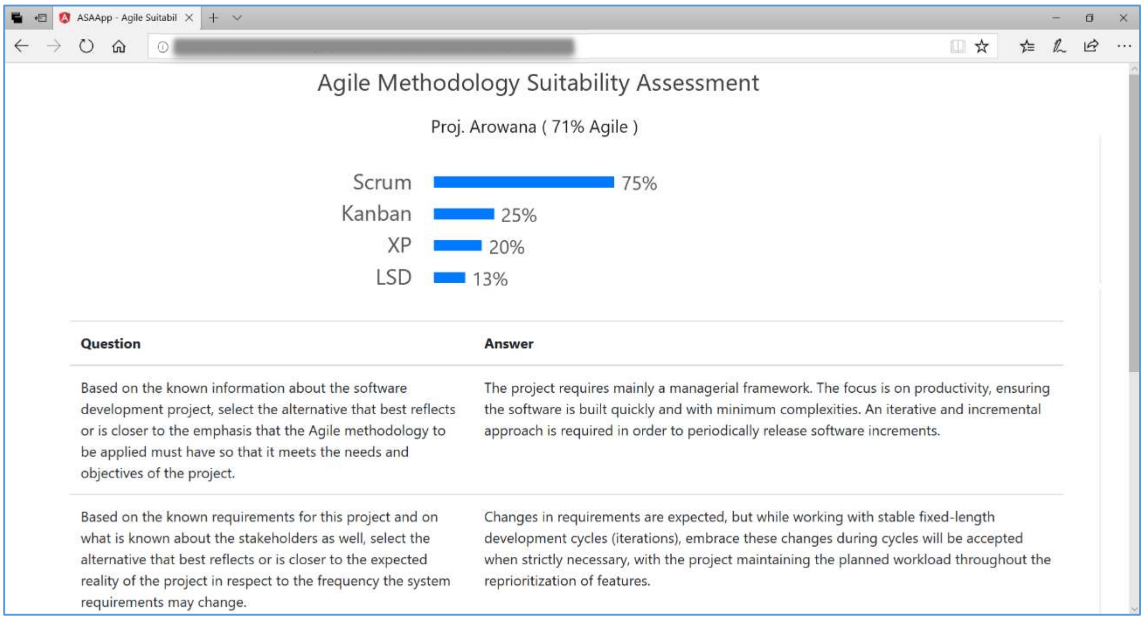
○ The project team will work on the planned features for the development cycle in the order defined by the customer or business representative. The project team will not be allowed to modify the order of the work.

○ The prioritization of the job will be done by whoever is in charge of it, in accordance to the project needs. The main concern is pulling work and often reprioritize remaining tasks in order to keep a paced workflow.

○ The prioritization of the work will be done by whoever is in charge of it, according to the needs of the project. The main concern is to avoid unnecessary work and prioritize tasks that directly add value to the software.

Figure 62 - Agile methodology suitability assessment (question 5 of 5)

After completing the questionnaire, the system will process and present the results along with the answers provided (cf. Figure 63). The evaluator can leave the assessment by closing the window, or return to the main page clicking on the button RETURN at the bottom of the page.



Agile Methodology Suitability Assessment

Proj. Arowana (71% Agile)

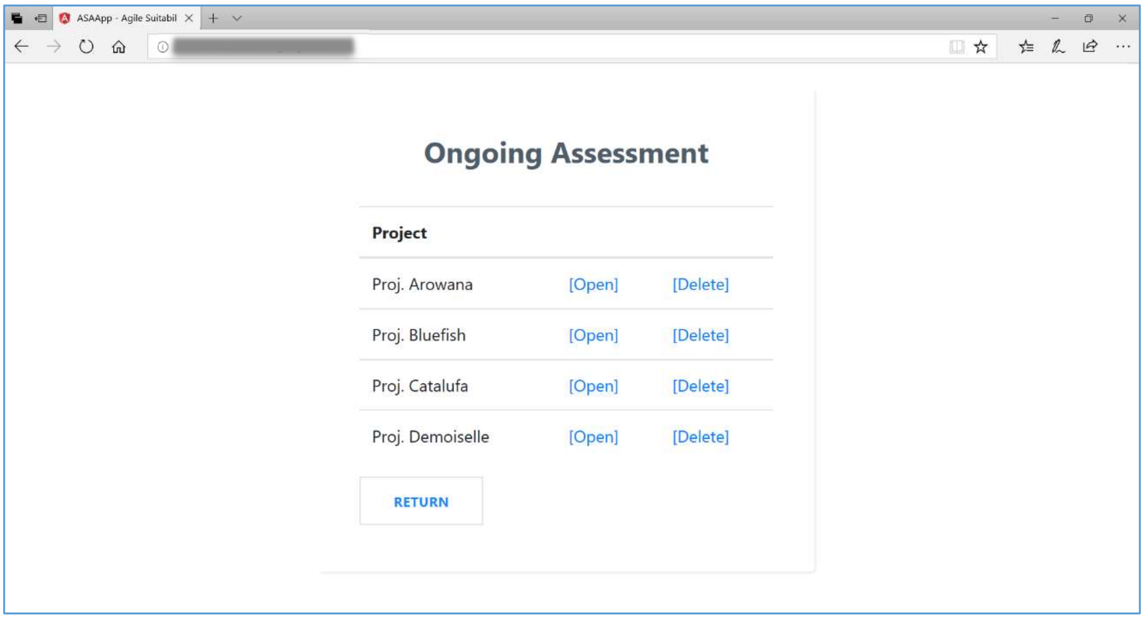
Scrum	75%
Kanban	25%
XP	20%
LSD	13%

Question	Answer
Based on the known information about the software development project, select the alternative that best reflects or is closer to the emphasis that the Agile methodology to be applied must have so that it meets the needs and objectives of the project.	The project requires mainly a managerial framework. The focus is on productivity, ensuring the software is built quickly and with minimum complexities. An iterative and incremental approach is required in order to periodically release software increments.
Based on the known requirements for this project and on what is known about the stakeholders as well, select the alternative that best reflects or is closer to the expected reality of the project in respect to the frequency the system requirements may change.	Changes in requirements are expected, but while working with stable fixed-length development cycles (iterations), embrace these changes during cycles will be accepted when strictly necessary, with the project maintaining the planned workload throughout the reprioritization of features.

Figure 63 - Agile methodology suitability assessment (results page)

Option: Ongoing Assessment

When the evaluator wants to continue an assessment partially filled out, he/she may click on the button "Ongoing Assessment" on the main page of the system. It will load a page displaying all incomplete assessments by the evaluator. He/she then may pick one from the list of projects by clicking on the link "[Open]" (cf. Figure 64).



Ongoing Assessment

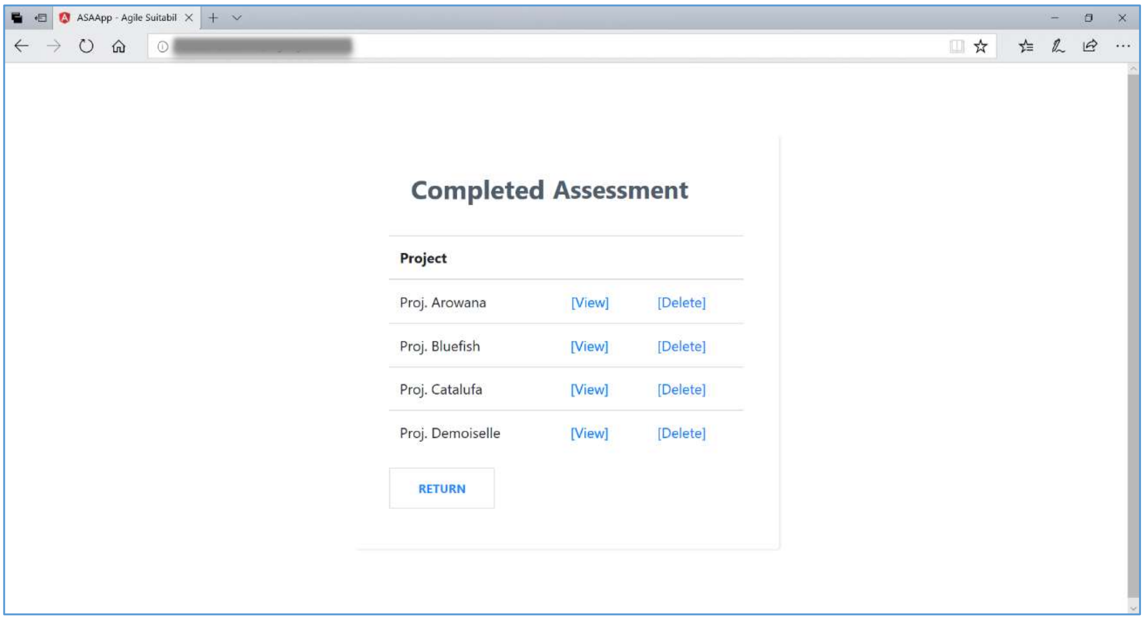
Project		
Proj. Arowana	[Open]	[Delete]
Proj. Bluefish	[Open]	[Delete]
Proj. Catalufa	[Open]	[Delete]
Proj. Demoiselle	[Open]	[Delete]

Figure 64 - Project list of ongoing assessments

After choosing the project, the questionnaire will be opened at the point of the assessment it has been interrupted. To complete the assessment, the evaluator follows the same sequence of steps showed in the option "New Assessment". He/she will also be allowed to delete an ongoing assessment by clicking on the link "[Delete]".

Option: Completed Assessment

If the evaluator wants to consult the score(s) and answers of a completed assessment, he/she clicks on the button "Completed Assessment" on the main page of the system. The system will show a page containing a list with of all assessments completed by the evaluator. He/she then may pick one from the list by clicking on the "[View]" link. (cf. Figure 65).



Completed Assessment

Project		
Proj. Arowana	[View]	[Delete]
Proj. Bluefish	[View]	[Delete]
Proj. Catalufa	[View]	[Delete]
Proj. Demoiselle	[View]	[Delete]

Figure 65 - Project list of completed assessments

The system will display a page with the result of the selected assessment. It may be the result of the approach assessment (cf. Figure 57), or the methodology assessment (cf. Figure 63). The evaluator will also be allowed to delete a completed assessment by clicking on the link "[Delete]".

5.3.5. Prototype validation

The validation of the prototype concentrated in two major areas. The first focused on validating structural aspects of the solution (usability, clarity, and intuitiveness). The second verified whether the way the conceptual model (cf. item 4.2) was prototyped meets the goals of the future system.

For this, we counted on the help of the group of professionals who participated in the (three) focus groups, providing essential insights for the elaboration of the conceptual model and design of the information system proposed. A link to the site where the system prototype was hosted was provided to the participants, and they navigated throughout the prototype, making considerations and registering their opinions. Each group filled out a form (elaborated in Microsoft Excel) answering some questions about the prototype. Appendix P shows the survey forms answered. Tables 24 and 25 summarize the feedback and inputs provided by the contributors.

5.3.5.1. Rating scale questions (from 1 to 7)

Table 24 - Answers of the rating scale questions

Question	Answer of Group 1	Answer of Group 2	Answer of Group 3	Average
1. How easy is the prototype to use?	7	7	7	7
2. How intuitive are the prototype functions?	7	6	7	6,7
3. How clear are the questions and answers of the questionnaire?	6	5	6	5,7

Scales Legend:

Question 1	Question 2	Question 3
1 - Extremely unclear	1 - Extremely counterintuitive	1 - Extremely unclear
2 - Very unclear	2 - Very counterintuitive	2 - Very unclear
3 - Unclear	3 - Counterintuitive	3 - Unclear
4 - Neither unclear nor clear	4 - Neither counterintuitive nor intuitive	4 - Neither unclear nor clear
5 - Clear	5 - Intuitive	5 - Clear
6 - Very clear	6 - Very intuitive	6 - Very clear
7 - Extremely clear	7 - Extremely intuitive	7 - Extremely clear

5.3.5.2. Open-ended questions

Table 25 - Answers of the open-ended questions

Question	Answers by group
4. Do you consider the prototype contains the essential functionalities to achieve the goals of the future system?	• Group 1: Yes. The functionalities for achieving the system goals are clearly and intuitively represented in the prototype. • Group 2: Yes, we do. • Group 3: Yes. All necessary resources are present in the prototype.
5. Is there anything essential missing on the prototype?	• Group 1: No. • Group 2: No. • Group 3: No. We think all the essential features have been implemented.
6. Do you have any thoughts on how to improve the prototype?	• Group 1: In questions where there are multiple statements or sub-questions to answer, the background could have a slightly different colour to make it easier to see the boundary between each one. • Group 2: It could be interesting to show in the prototype where it will be possible to consult the Help, especially for the questions. • Group 3: No.

7. Do you have any other comments?	• Group 1: We really liked the result. We can clearly see our contributions represented in this prototype. We look forward to the final system! • Group 2: No. • Group 3: The prototype well represents the way we work on Agile projects. It is simple, intuitive and objective.

5.3.5.3. Brief conclusions about the prototype usage

Analysing the answers provided by the three different groups, we can say that the groups consider the prototype user-friendly (100%) and intuitive to use (96%). Regarding the clarity of the questions and possible answers formulated, the average of the group responses was 81%, which is a good result, but amenable to improvements. This can be done through a review of the text of questions and answers, as well as by implementing a Help feature. Regarding the utility of the prototype – meaning if it has the necessary functions to achieve the goals of the future system – the groups agreed that all resources are present in the prototype. They appreciated the end-result and also suggested a few improvements.

Based on the impressions that our research contributors expressed about the developed prototype, we conclude that it is adequate and serves as a basis for building an information system to support the process of choosing an Agile software development methodology.

Chapter 6 – Conclusions and Future Research

6.1. Conclusions

In this research, we intended to verify how an information system can support the process of choosing an Agile methodology for software development. Concerning our general objectives, we conclude that:

- Objective 1: Determine the suitability criteria of a software development project for an Agile approach.
  - ✓ We reached this objective by identifying nine relevant aspects (distributed into three categories) that should be considered to determine if an Agile approach is appropriate in the context of a software development project (cf. item 4.1.3.1).
- Objective 2: Identify the main aspects to consider in the determination of the best Agile methodology to use in a given software development context (limited to the Agile frameworks Scrum, Extreme Programming, Kanban, and Lean Software Development).
  - ✓ This goal was achieved by identifying the five aspects that must be considered to perform an Agile methodology suitability assessment in the context of a software development effort (cf. item 4.1.3.2).
- Objective 3: Elaborate a decision model based on the suitability criteria (determined in objective 1) and the main aspects (identified in objective 2).
  - ✓ By defining the conceptual model that suggests the use of an Agile approach and indicates an Agile methodology to use (when an Agile approach fits) we fulfilled this objective (cf. items 4.2.1 and 4.2.2).
- Objective 4: Validate the proposed decision model.
  - ✓ We reached this goal by using data from real software development projects to compare the approach and methodology suggested by the conceptual model with the approach and methodology used in these projects. The number of equal results obtained supports the adequacy of the conceptual model to assess the Agile suitability (cf. item 4.2.3).
- Objective 5: Model and design the information system that will implement the proposed decision model.

  - ✓ This objective was achieved in chapter 5 when we used Agile practices and techniques to model and design this information system (cf. items 5.3.1, 5.3.2, and 5.3.3).

- Objective 6: Develop and validate a prototype of the proposed system.
  - ✓ This goal was reached by developing our prototype using Angular 7 as programming language and Visual Studio Code as IDE (cf. item 5.3.4), and validating the prototype with the help of the participants of the focus groups (cf. item 5.3.5).

With the attainment of these general objectives, we have answered the research question and fulfilled our major objective (and great contribution of the research), proving that an information system can be built to support the decision-maker in the selection of a suitable Agile software development approach. Our next step is to continue this project by developing the system proposed here and using it for its intended purpose.

6.2. Research Limitations

The limited availability of literature regarding the Agile suitability models was a major limitation for this research. The existing Agile suitability filters (cf. item 2.4.4) are relatively obsolete, and it was not possible to find usage history of them, which, on the other hand, reinforces the need of this study.

A purposeful limitation was the decision to restrict the investigation to four (commonly used) Agile methodologies. A question we asked ourselves before starting this study was whether we should include more recently created methodologies, such as the Spotify Squad framework[36], or hybrids, such as SCRUMBAN[37]. However, these are methodologies with less possibility of obtaining reliable research data. Unlike the ones chosen, they have limited literature (of good quality) and also restricted usage history, which would make more difficult to validate our conceptual model.

Obtaining real software development projects data to support the validation of the conceptual model proved to be a challenge. Having access

[36] Spotify Squad framework (or Spotify Model) is an Agile methodology inspired in Scrum, Extreme Programming and Lean Software Development frameworks. It created the concepts of Tribes, Squads, Chapters, and Guilds. The Squad is like a Scrum team. A Tribe is a group of Squads. Chapters are part of a Squad and also a group of team members that work together. The Guild is a group of people who shares interests (Fernandes, 2017).

[37] SCRUMBAN is a hybrid Agile methodology that combines the features of both Scrum and Kanban frameworks. The framework is popular in the service industries, where development and maintenance projects are quite common (Pahuja, 2018).

to such data, containing the necessary characteristics for the validation, and to people able to answer the assessment questionnaires were limitations for this investigation as well. A crucial factor in the success of this process was the author's professional network.

6.3. Proposals for Future Research

As proposals for the continuity of this research, aiming to expand the results obtained here, we have a few suggestions.

Future investigations could involve exploratory studies with groups whose experiences with Agile methodologies resulted in unsatisfactory outcomes. This approach could provide a better understanding of when Agile should not be used. Going a bit deeper, researching when the use of a specific traditional (non-Agile) software development methodology is suitable could also be explored.

As a way of gauging more precisely (and recalibrating if necessary) the weights of the items assessed both in the Agile approach assessment and in the Agile methodology assessment, it might be interesting to test the conceived decision-model with a broader base of software development projects. As mentioned before, obtaining such data is challenging, but it can be a valuable contribution to the continuity of this work as well.

Finally, another idea that can be explored in future studies may be expanding the group of Agile methodologies researched to include more recent frameworks and also hybrid ones, purposely excluded from this investigation.

6.4. Final Remark

In sum, we consider that the present work is a valuable contribution, not only in the scientific domain of information systems and programming and, in particular, of software development methodologies, but also for all the organizations with IT department and all professionals that contribute to the development of software, and need to choose between different options when they need to work in optimized solutions.

Bibliography

Ahmad, M. O., Markkula, J., & Ovio, M. (2013). Kanban in software development: A systematic literature review. *39th Euromicro Conference on Software Engineering and Advanced Applications* (pp. 9-16). Santander, Spain: IEEE. doi:10.1109/SEAA.2013.28

Anderson, J. D. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business.* Sequim, WA: Blue Hole Press.

Apel, S., & Kästner, C. (2009, July–August). An Overview of Feature-Oriented Software Development. *Journal of Object Technology*, pp. pages 49–84.

Asim, M. (2016, February 14). *V-Model and W Model.* Retrieved from LinkedIn: https://www.slideshare.net/asim741/v-model-and-w-model

Association of Modern Technologies Professionals. (2019). *Software Development Methodologies*. Retrieved from IT Knowledge Portal: http://www.itinfo.am/eng/software-development-methodologies/

Balaji, S., & Murugaiyan, M. S. (2012, June 29). Waterfall Vs V-Model Vs Agile: A Comparative Study on SDLC. *International Journal of Information Technology and Business Management*.

Beck, K. (2004). *Extreme Programming Explained: Embrace Change.* Boston, MA: Addison-Wesley Professional.

Beck, K., Grenning, J., Martin, R. C., Beedle, M., Highsmith, J., Mellor, S., . . . Marick, B. (2001). *Manifesto for Agile Software Development*. Retrieved from Manifesto for Agile Software Development: http://agilemanifesto.org/

Boehm, B. W. (1988, May). A Spiral Model of Software Development and Enhancement. *IEEE Computer Society, 21*, pp. 61-72.

Boehm, B., & Turner, R. (2003). *Balancing Agility and Discipline: A Guide for the Perplexed.* Boston, MA: Pearson Education, Inc.

Boer, G. (2017, April 4). *What is Scrum.* Retrieved from Microsoft: https://docs.microsoft.com/en-us/azure/devops/learn/agile/what-is-scrum

Brodzinski, P. (2015, October 26). *Don't Limit Work in Progress.* Retrieved from Software Management Project: http://brodzinski.com/2015/10/dont-limit-wip.html

Brown, D. M. (2011). *Communicating Design: Developing Web Site Documentation for Design and Planning.* Berkeley, CA: New Riders.

Burley, K. (2017, July 19). What Is a Deliverable in Project Management? *Houston Chronicle*.

Calikus, E. (2015, October). *Mobile App Analytics & Sentiment Analysis of Customer Reviews.* Retrieved from https://www.researchgate.net/publication/283048282_Mobile_App_Analytics_Sentiment_Analysis_of_Customer_Reviews

Cockburn, A. (2001). *Writing Effective Use Cases.* Upper Saddle River, NJ: Addison-Wesley.

CollabNet VersionOne. (2019). *The 13th Annual State of Agile Report.* Alpharetta, GA: CollabNet VersionOne. Retrieved from https://www.stateofagile.com/?_ga=2.190828456.268075660.1567500912-491544011.1567500912#ufh-i-521251909-13th-annual-state-of-agile-report/473508

Coutinho, C. (2017). *Tecnologias e Sistemas Cloud - Aula 01.* Retrieved from e-Learning ISCTE-IUL: https://e-learning.iscte-iul.pt/webapps/blackboard/content/listContent.jsp?course_id=_29367_1&content_id=_263118_1

Dąbrowski, M. (2017). *Agile Project Management V2 (AgilePM® V2) study guide mind map.* Retrieved from Mind Meister: https://www.mindmeister.com/zh/287712070/agile-project-management-v2-agilepm-v2-study-guide-mind-map

Dam, R., & Siang, T. (2018, 12 11). *Personas – A Simple Introduction*. Retrieved from Interaction Design Foundation: https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them

East Carolina University. (2015, May 29). *What is a Likert Scale?* Retrieved from East Carolina University: http://core.ecu.edu/psyc/wuenschk/StatHelp/Likert.htm

ECM Lab. (2004). *Prosci's Change Management Maturity Model.* Retrieved from Prosci: http://www.change-management.com/ecmlab/ECM-Lab-Preread-M-Model.pdf

Fernandes, T. (2017, March 6). *Spotify Squad framework - Part I.* Retrieved from Medium: https://medium.com/productmanagement101/spotify-squad-framework-part-i-8f74bcfcd761

Fonseca, J. G. (1997). *Faces da Decisão: as mudanças de paradigmas e o poder da decisão.* São Paulo, SP: Makron Books.

Gonçalves, L. (2019, April 2). *Burndown Chart – O Guia Oficial para Qualquer Scrum Master.* Retrieved from Scaling Your Business: https://luis-goncalves.com/pt-pt/burndown-chart/

Griffiths, M. (2007). *Agile Suitability Filters.* Retrieved from Leading Answers: https://leadinganswers.typepad.com/leading_answers/files/agile_suitability_filters.pdf

Griffiths, M. (2015). *PMI-ACP® Exam Prep: A Course in a Book for Passing the PMI Agile Certified Practitioner (PMI-ACP) Exam.* Minnetonka, MN: RMC Publications.

Hawks, D. (2010, March 7). *4 Advantages of Physical Task Boards.* Retrieved from Agile Velocity: https://agilevelocity.com/scrum/4-advantages-of-physical-task-boards/

Hossain, E., Babar, M. A., & Paik, H.-y. (2009). Using Scrum in Global Software Development: A Systematic Literature Review. *Fourth IEEE International Conference on Global Software Engineering* (pp. 175-184). Limerick, Ireland: IEEE. doi:10.1109/icgse.2009.25

Icasas, P. (2013, November 8). *The Spiral Project Management Method*. Retrieved from Easy Projects: https://explore.easyprojects.net/blog/upward-spiral-spiral-project-management-method/280px-software_development_spiral_svg#agree

IEEE. (1990). *IEEE Standard Glossary of Software Engineering Terminology.* New York, USA: IEEE.

Jadhav, M. A., Sawant, B. R., & Deshmukh, A. (2015). Single Page Application using AngularJS. *International Journal of Computer Science and Information Technologies*.

Jaikrishnan, V. (2019, January 23). *Understanding Cumulative Flow Diagram.* Retrieved from Zepel: https://zepel.io/agile/reports/cumulative-flow-diagram/

Järvinen, P. H. (1997). The New Classification of Research Approaches. *The IFIP Pink Summary - 36 years of IFIP.* Laxenburg, Austria: International Federation of Information Processing Societies.

Järvinen, P. H. (2000). On a variety of research output types. *Proceedings of the IRIS 23.* Trollhättan, SE: University of Trollhättan.

Järvinen, P. H. (2000). Research Questions Guiding Selection of an Appropriate Research Method. *ECIS 2000, 3-5 July* (pp. 124-131). Vienna, Austria: AIS Electronic Library.

Jurevicius, O. (2013, April 25). *Value Chain Analysis.* Retrieved from Strategic Management Insight: https://www.strategicmanagementinsight.com/tools/value-chain-analysis.html

Kahneman, D., Slovic, P., & Tversky, A. (1982). *Judgment under Uncertainty: Heuristics and Biases.* Cambridge, UK: Cambridge University Press.

Konsynski, B. R., Kottemann,, J. E., Nunamaker Jr., J. F., & Stott, J. W. (1984). An Integrated Development Environment for Information Systems. *Journal of Management Information Systems*, pp. 64-104.

Kothari, C. R. (2008). *Research Methodology: Methods and Techniques* (4th Edition ed.). New Delhi, India: New Age International.

Kourounakis, N., Maraslis, A., Velitchkov, I., Sequeira, L., & Beghain, O. (2015). *Agile@EC Guide* (v.1.3.2 ed.). Brussels, Belgium: European Commission Advisory Services.

Larman, C., & Basili, V. R. (2003). Iterative and Incremental Development: A Brief History. *Computer*, 47-56.

Layman, L., Williams, L., & Cunningham, L. (2004). Exploring Extreme Programming in Context: An Industrial Case Study. *Agile Development Conference* (pp. 22-26). Salt Lake City, UT: IEEE. doi:10.1109/ADEVC.2004.15

Leximancer Pty Ltd. (2018, April 12). *Leximancer User Guide, Release 4.5.* Retrieved from Leximancer: https://doc.leximancer.com/doc/LeximancerManual.pdf

Luftman, J. N., Lewis, P. R., & Oldach, S. H. (1993). Transforming the enterprise: The alignment of business and information technology strategies. *IBM Systems Journal, 32*(1), pp. 198-221.

M.C. Partners & Associates. (2012, 12). *Lean / Kanban Case-Study.* Retrieved from M.C. Partners & Associates: http://www.mcpa.biz/wp-content/uploads/2012/12/Kanban-Case-Study-FINAL.pdf

Martin, R. C. (2003). *Agile Software Development, Principles, Patterns, and Practices.* London, UK: Pearson Education.

Mattos, A. M. (2005). *Sistemas de Informação: uma visão executiva.* São Paulo, SP: Saraiva.

Middleton, P., & Joyce, D. (2011, April 05). Lean Software Management: BBC Worldwide Case Study. *IEEE Transactions on Engineering Management*, pp. 20-32. doi:10.1109/TEM.2010.2081675

Morgan, D. L. (1997). *Focus Groups as Qualitative Research* (2nd Edition ed.). Thousand Oaks, CA: Sage Publications, Inc.

Mullaly, M. E., & Thomas, J. (2010). Re-thinking project management maturity: perspectives gained from explorations of fit and value. *PMI® Research Conference: Defining the Future of Project Management.* Washington, DC: Project Management Institute.

Murch, R. (2012). *The Software Development Lifecycle - A Complete Guide.* Seattle, WA: Kindle Direct Publishing.

Nuseibeh, B., & Easterbrook, S. (2000). Requirements Engineering: A Roadmap. *Conference on The Future of Software Engineering.*

Pahuja, S. (2018, June). *What is Scrumban?* Retrieved from Agile Alliance: https://www.agilealliance.org/what-is-scrumban/

Patel, H., Pettitt, M., & Wilson, J. R. (2012). Factors of collaborative working: A framework for a collaboration model. *Applied Ergonomics*, pp. 1-26.

PMI. (2017). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Sixth Edition.* Newton Square, PA: Project Management Institute, Inc.

PMI. (2017). *Agile Practice Guide.* Newtown Square, PA: Project Management Institute.

Pooley, R., & Wilcox, P. (2004). *Applying UML: Advanced Application.* Burlington, MA: Elsevier Butterworth-Heinemann.

Poppendieck, M., & Poppendieck, T. (2003). *Lean Software Development: An Agile Toolkit.* Boston, MA: Addison-Wesley Professional.

Porter, M. E. (2004). *Competitive Advantage: Creating and Sustaining Superior Performance.* New York, NY: Free Press.

Project Management Institute. (2019). *PMI.ORG*. Retrieved from Project Management Institute: https://www.pmi.org/

Ramos, P. N. (2012). *Desenhar Bases de Dados com UML.* Lisboa: Edições Sílabo, Lda.

Ries, E. (2011). *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses.* New York, NY: Crown Publishing Group.

Rising, L., & Janoff, N. S. (2000). The Scrum Software Development Process for Small Teams. *IEEE Software*, 26-32. doi:10.1109/52.854065

Rodrigues, L. F. (2017). *Gestão Estratégica de Sistemas de Informação.* Retrieved from e-Learning. ISCTE-IUL: https://e-learning.iscte-iul.pt/webapps/blackboard/execute/content/file?cmd=view&content_id=_266185_1&course_id=_27194_1

Royce, W. W. (1970). Managing the development of large software systems: concepts and techniques. *IEEE WESTCON*, (pp. 328-338). Los Angeles, CA.

S., A. (2016, August 26). *Extreme Programming.* Retrieved from ANTOSSOFTWARETOPICS: https://antossoftwaretopics.wordpress.com/2016/08/26/extreme-programming/

Santos, D. L. (2018, July 27). *Quando e como utilizar protótipos.* Retrieved from Medium: https://medium.com/cwi-software/quando-e-como-utilizar-prot%C3%B3tipos-ca475657c8d3

Schmidt, R. F. (2013). *Software Enginneering - Architecture-Driven Software Development.* Waltham, MA: Morgan Kaufmann.

SCRUMstudy. (2014, July 24). *Scrum vs. Traditional Project Management.* Retrieved from Medium: https://medium.com/@SCRUMstudy/scrum-vs-traditional-project-management-60f15f1e63f

Sommerville, I. (2016). *Software Engineering* (10th ed.). Harlow, UK: Pearson.

Sutherland, J. (2015). *Scrum: The Art of Doing Twice the Work in Half the Time.* New York, NY: Random House Business Books.

University of Southern California. (2019, Jan 29). *Organizing Your Social Sciences Research Paper: 5. The Literature Review*. Retrieved from USC Libraries: http://libguides.usc.edu/writingguide/literaturereview

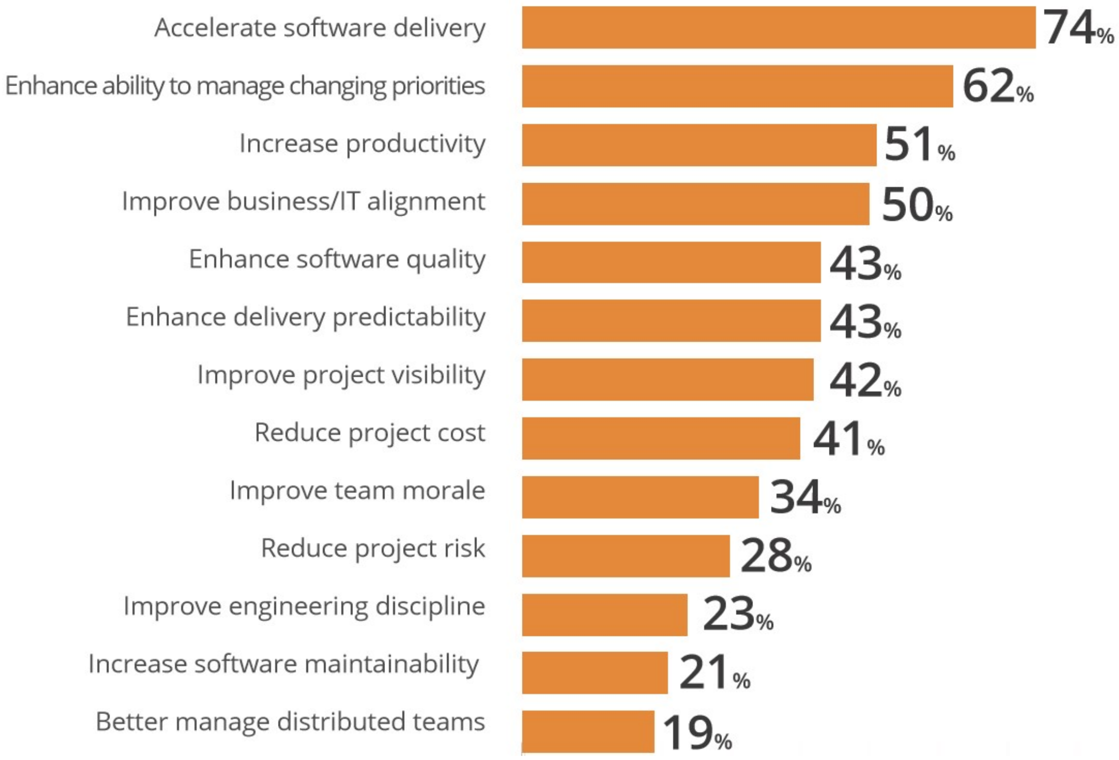
Vanzin, M., Ribeiro, M. B., Prikladnicki, R., Ceccato, I., & Antunes, D. (2005). Global Software Processes Definition in a Distributed Environment. *Proceedings of the 2005 29th Annual IEEE/NASA Software Engineering Workshop.* Greenbelt, MD: IEEE.

Widman, J., Hua, S. Y., & Ross, S. C. (2010). Applying Lean Principles in Software Development Process – A Case Study. *Issues in Information Systems*, pp. 635-639.

Wilson, L. (2009). *How to Implement Lean Manufacturing.* New York, NY: McGraw Hill Professional.

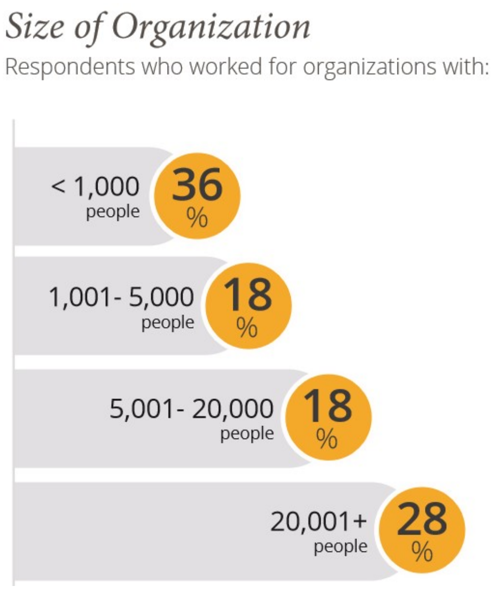
Annexes

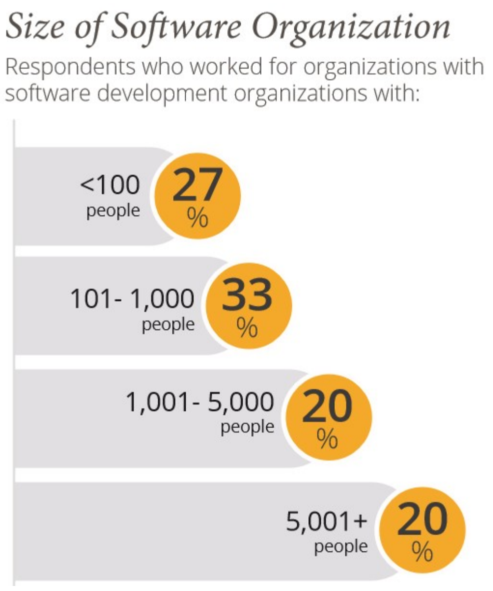
Annex A – 13th Annual State of Agile Report - Reasons for Adopting Agile

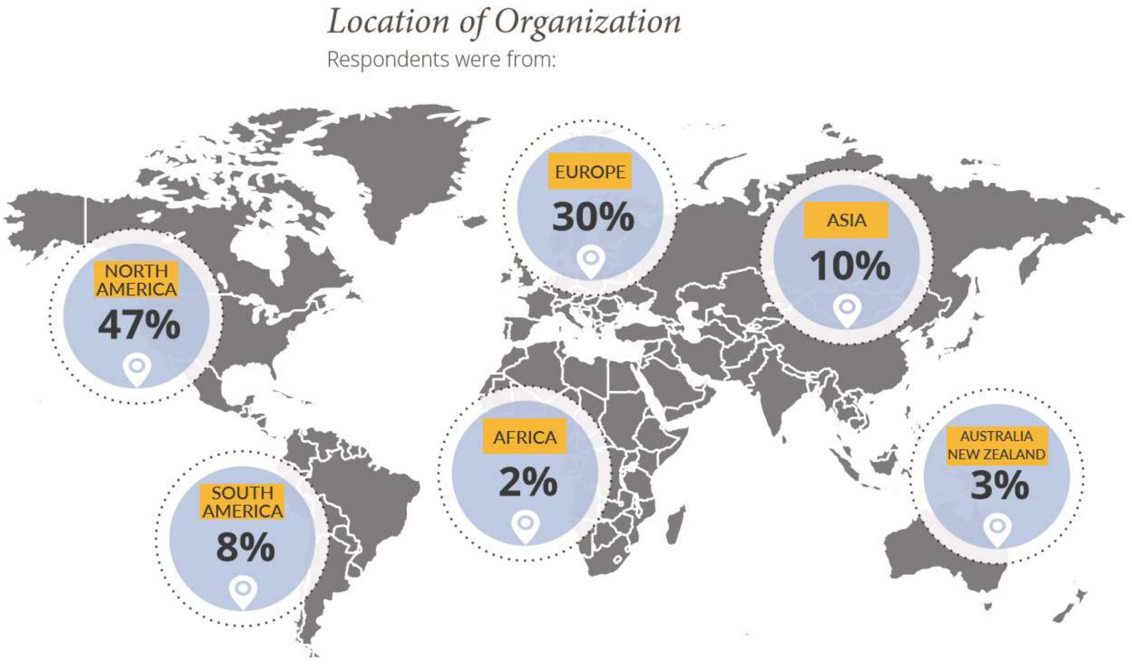


Source: 13th Annual State of Agile Report, page 7 (CollabNet VersionOne, 2019)

Annex B – 13th Annual State of Agile Report - Respondent Demographics

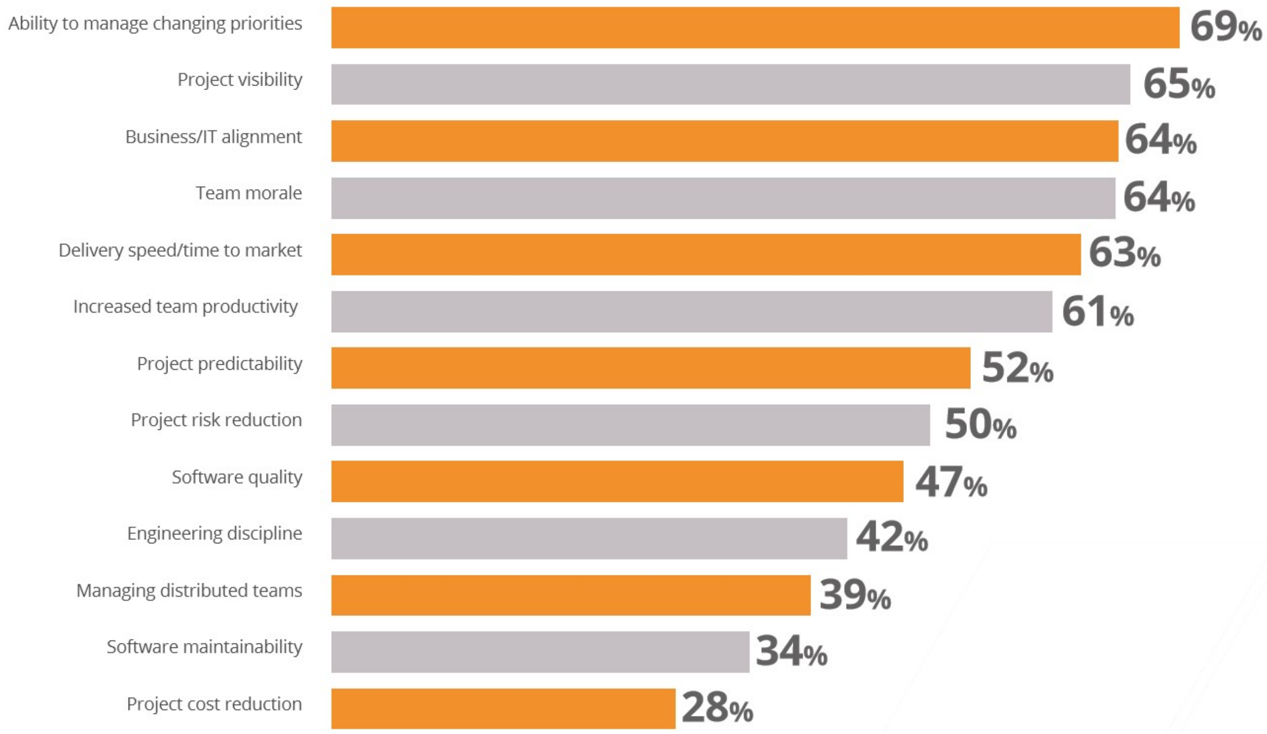






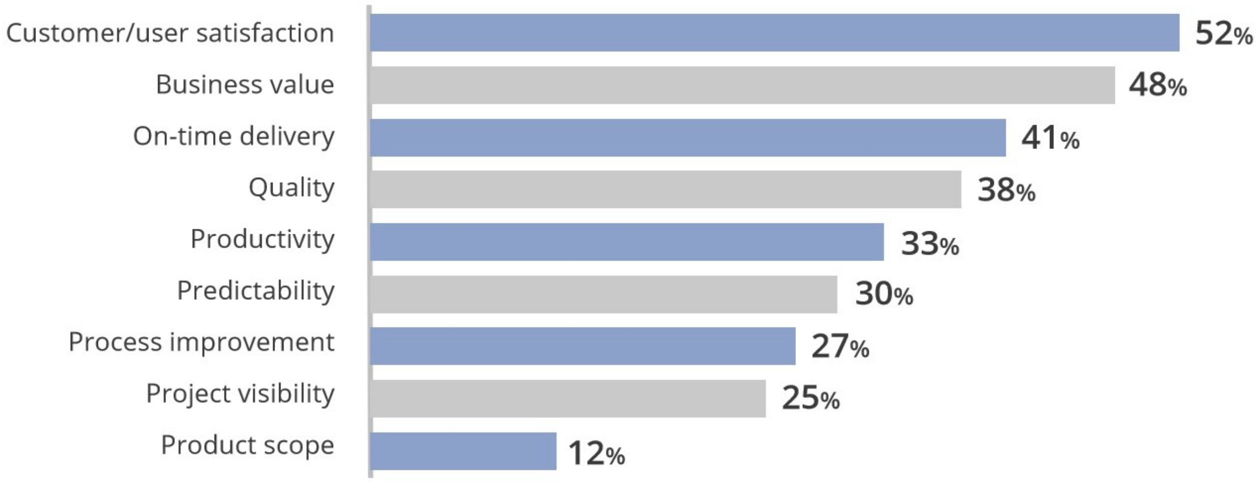
Source: 13th Annual State of Agile Report, page 5 (CollabNet VersionOne, 2019)

Annex C – 13th Annual State of Agile Report - Benefits of Adopting Agile



Source: 13th Annual State of Agile Report, page 8 (CollabNet VersionOne, 2019)

Annex D – 13th Annual State of Agile Report - How Success is Measured in Agile Initiatives



Source: 13th Annual State of Agile Report, page 11 (CollabNet VersionOne, 2019)

Appendices

Appendix A – Focus Group Script

I. Introduction / Study set up

- Receiving participants;
- Set up of technical means for recording;
- Identification of the researcher;
- Presentation of the main objectives of the focus group;
- A brief explanation of the concept and rules of a focus group;
- Reference to the essential areas to be addressed in the discussion;
- Reference to the ethical principles of the research in development;
- Clarification of any questions;
- Brief presentation of participants;
- Beginning of the debate.

II. Questions to be addressed

Part I - It encompasses questions that will explore the aspects relating to choosing an Agile approach as a guide to the management and execution of a software development project. The objective is to collect the impressions and opinions of the participants on the pertinent questions formulated. These questions have been elaborated based on the review and analysis of Agile literature.

- Question 1/5: Under what conditions would you suggest an organisation to opt for an Agile methodology?

- Question 2/5: Regarding the aspects of the ORGANISATION, more specifically within the MANAGEMENT level, what are the characteristics and behaviours that you think the leaders of the organisation should have to work with an Agile methodology?

- Question 3/5: Regarding the aspects of the ORGANISATION, more specifically within the TEAM, what are the characteristics and behaviours that you think the project team should have / present to work with an Agile methodology?

- Question 4/5: Regarding the aspects of the PROJECT, more specifically in the ENVIRONMENT, what are the environmental factors for the project that you consider to be more characteristic of initiatives eligible to use an Agile methodology?
- Question 5/5: Regarding the aspect of the PROJECT, more specifically in the PRODUCT scope, what are the characteristics related to the product of the project you think are most relevant in determining an Agile approach as a development methodology?

Part II - Requirements gathering of the information system that will be designed to implement the decision model derived from the analysis of the aspects discussed in Part I. The objective is to collect the opinion of the participants about the ESSENTIAL features for this information system.

- Question 1/1: An information system will be developed, aiming to implement the decision-making model resulted from the analysis of the aspects discussed in the first part of this focus group. The main objective of the system is to indicate if the project is eligible for an Agile approach or not. In this context, what would be, in your opinion, the ESSENTIAL functionalities of the system?

III. Register of focus group conditions

- Moderator – name;
- Participants – professional experience and type of organisation in which the participants work, degree of experience with Agile; Agile usage (if any) in the organisation the participants currently work;
- Location – city and country;
- Date and time;
- Duration of the activity;
- Comments.

Appendix B – Agile Specialist Interview Script

I. Introduction / Study set up

- Receiving the interviewee;
- Set up of technical means for recording;
- Identification of the researcher;
- Presentation of the main objectives of the interview;
- Reference to the essential areas to be addressed in the interview;
- Reference to the ethical principles of the research in development;
- Clarification of any questions;
- Brief presentation of the interviewee;
- Beginning of the interview.

II. Questions to be addressed

Part I - It encompasses questions that will explore the aspects relating to choosing an Agile approach as a guide to the management and execution of a software development project. The objective is to collect the impressions and opinions of the interviewee on the pertinent questions formulated. These questions have been elaborated based on the review and analysis of Agile literature.

- Question 1/5: Under what conditions would you suggest an organisation to opt for an Agile methodology?

- Question 2/5: Regarding the aspects of the ORGANISATION, more specifically within the MANAGEMENT level, what are the characteristics and behaviours that you think the leaders of the organisation should have to work with an Agile methodology?

- Question 3/5: Regarding the aspects of the ORGANISATION, more specifically within the TEAM, what are the characteristics and behaviours that you think the project team should have / present to work with an Agile methodology?

- Question 4/5: Regarding the aspects of the PROJECT, more specifically in the ENVIRONMENT, what are the environmental factors for the project that you consider to be more characteristic of initiatives eligible to use an Agile methodology?

- Question 5/5: Regarding the aspect of the PROJECT, more specifically in the PRODUCT scope, what are the characteristics related to the product of the project you think are most relevant in determining an Agile approach as a development methodology?

Part II - It encompasses the issues that will explore the aspects related to the choice of an Agile methodology (out of the four studied) aligned to the needs and environment of software development projects. The objective is to collect the impressions and opinions of the interviewee on the pertinent questions formulated. These questions were elaborated based on the review and analysis of Agile literature. The following questions must consider the methodologies: Scrum; Extreme Programming; Lean Software Development; and Kanban.

- Question 1/5: Regarding the changes in system requirements, how do you believe each methodology addresses this aspect? In what way does the positioning of each one on the use of development cycles (iterations) influence this aspect?
- Question 2/5: Regarding the use of software engineering practices, how does each methodology address this aspect? Is there one that gives more emphasis on the use of software engineering practices (in detriment of project management practices)? If so, what are the reasons for this?
- Question 3/5: In relation to the customer engagement in the project, the fourth principle of the Agile Manifesto stands out the need for greater interaction and collaboration between the project team and the business representatives. All Agile methodologies apply this principle at different levels. Based on your knowledge of these methodologies, how do you think these methodologies differ from this principle?
- Question 4/5: Regarding the approach to software deliveries (or parts of the software), the third Agile fundamental principle says, "Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale." In Agile, this principle is widely used but is applied in a specific way by each methodology. In relation to the

following aspects identified, based on your knowledge of these four methodologies, how would you say each one approaches them?

  - Continuous delivery – as a way of obtaining an effective cadence and determining a "standard" speed for deliveries;
  - Deliver as fast as possible – the focus is on the speed of delivering value to the customer;
  - Minimum viable product (MVP) – delivering a simplified but functional initial version of the software.

- Question 5/5: Regarding the prioritisation of system requirements, the first Agile principle says, "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software." Agile methodologies work to prioritise the delivery of product functions that will bring greater value to the business (from the customer's point of view). However, each methodology deals with this prioritisation in different ways. Based on your knowledge of the four Agile methodologies under discussion, how do each of them work on the prioritisation of features and activities within a software development project?

Part III - Requirements gathering of the information system that will be designed to implement the decision model derived from the analysis of the aspects discussed in Part I. The objective is to collect the opinion of the interviewee about the ESSENTIAL features for this information system.

- Question 1/1: An information system will be developed, aiming to implement the decision-making model resulted from the analysis of the aspects discussed in the first part of this interview. The main objective of the system is to indicate if the project is eligible for an Agile approach or not. In this context, what would be, in your opinion, the ESSENTIAL functionalities of the system?

III. Register of interview conditions

- Interviewer – name;
- Interviewee – name, professional experience, and type of organisation in which the interviewee work, degree of experience with Agile;
- Location – city and country;

- Date and time;
- Duration of the interview;
- Comments.

Appendix C – Interview Transcript with Agile Specialist [Portuguese]

Informações da Entrevista	
Entrevistador	Luis Fernando Bordeaux Mattos
Entrevistado	Tiago Palhoto – Agile coach e responsável pela metodologia Agile@EC, utilizada em projetos de desenvolvimento e manutenção de software na Comissão Europeia.
Local	Bruxelas, Bélgica.
Data e hora	24 de janeiro de 2019.
Duração	01h11min
Comentários	A língua utilizada na entrevista foi o Português (Portugal / Brasil).

Parte I

Engloba as questões que irão explorar os aspetos relativos à escolha de uma abordagem Agile como guia para as ações de gestão e execução de um projeto de desenvolvimento de software. O objetivo é coletar as impressões e opiniões do entrevistado às questões pertinentes formuladas. Tais questões foram elaboradas com base na revisão e análise de literatura sobre Agile.

Questão 1/5

Luis Mattos: Em que condições sugeriria a uma organização a opção por uma metodologia Agile?

Tiago Palhoto: Há essencialmente aqui dois pontos. O primeiro é ter atingido aquilo que chamam de *tipping point*, ou seja, aquele momento em que se fala: "Nós temos que mudar, ponto final." Portanto, é algo que organização sente e diz... "Isto como está não pode ser. Nós não estamos a melhorar. Está provado que isto não está a funcionar. Temos mesmo que mudar." Portanto, isso é uma das principais razões para mudar. Isso inclusive é abordado na própria *Scaled Agile Framework* como sendo um dos pontos em que se encoraja a decisão de mudar e usar uma abordagem Agile. O segundo ponto é efetivamente quando tens pessoas, ou seja… o primeiro tem a ver com uma necessidade que é visível para todos, já o segundo tem a ver essencialmente quando tens um conjunto de pessoas que já tem essa perceção, que já tem essa noção do que é o Agile. Portanto eles próprios querem... "Vamos experimentar esta abordagem, pois nós temos aqui um conjunto de pessoas que já conhecem a abordagem." Enfim são pessoas que já tem os princípios básicos, que já estão infetados um bocadinho com alguns fundamentos e princípios do Agile e portanto tens ali um conjunto de pessoas indicadas para dares o primeiro passo. Este segundo obviamente que acaba por ser um bocadinho uma ajuda ao primeiro, mas acima de tudo, creio que o primeiro, pela minha experiência, realmente acaba por ser aquele que traz uma vontade maior no início, mas que não dispensa o segundo, ok? Enquanto que se tiveres o segundo efetivamente vindo dos altos executivos da empresa, pode ser mais fácil de implementar, mas por outro lado pode ser um bocadinho mais difícil quebrar as resistências iniciais. Mas eu considero que estes dois pontos são as condições em que eu sugeriria à uma organização a adoção do Agile.

Questão 2/5

***Luis Mattos*:** Em relação aos aspetos da ORGANIZAÇÃO, mais especificamente no âmbito da LIDERANÇA, quais são as características e comportamentos que julga que os líderes da organização devem ter / apresentar a trabalhar com alguma metodologia Agile?

***Tiago Palhoto*:** Tem um para mim que é fundamental que é a tolerância. A tolerância no sentido de compreender que uma mudança dessas traz obviamente experimentação, traz erros. As pessoas vão errar, esses erros farão parte do seu trabalho diário e isto é bom. É uma mudança de *mindset* das pessoas que não é fácil... "Mas então nós vamos experimentar uma coisa e vamos errar e vamos falar que erramos e etc.?" Faz parte. Tem que acontecer. Tem um segundo ponto que é passar de uma abordagem *command and control* para uma abordagem de confiança nas pessoas que realmente vão fazer do trabalho. Isso é fundamental. Todos sabem, mas poucos aplicam. Ou seja, eu tenho equipas, pessoas, profissionais que eu contrato para fazer uma tarefa, um sistema seja o que for, portanto, são eles as melhores pessoas para executarem essa missão. Eu tenho que confiar neles. O terceiro ponto é criar e garantir que se criam as condições de trabalho para que as pessoas tenham realmente orgulho e gostem daquilo que fazem. Normalmente as primeiras pessoas a sair são aquelas que não gostam daquilo que estão a fazer e que não sentem orgulho e brio naquilo que estão a fazer. Portanto, se nós não criarmos uma atmosfera, se não criamos condições para que não só deixemos as pessoas trabalhar, mas acima de tudo, deixemos usar a sua criatividade e que elas sintam utilidade e sintam brio naquilo que estão a fazer, não vai funcionar. Isto não é só nas abordagens Agile, mas de uma forma geral. Quando falamos precisamente em dar a liberdade às equipas é dar a liberdade para elas usarem a sua criatividade e fazerem coisas que são realmente úteis. O primeiro ponto que eu falei foi precisamente, que dizer, se não tivermos essa eficiência, essa capacidade de tolerância para errar imediatamente, vou estar a limitar a criatividade, e isso depois é uma reação em cadeia que acaba por limitar as iniciativas da equipa. Estes para mim são três pontos fundamentais. Obviamente depois há um outro aspeto que já tem a ver mais um bocado com a orgânica que é perceber que o Agile acaba um bocadinho com essa noção de competências muito específicas. Ou seja, eu até vou um bocadinho mais fundo nisso, pois nós começamos com uma abordagem onde nós criamos equipas *cross-functional* e portanto é uma abordagem que acaba por não ser muito compatível com estruturas orgânicas muito rígidas, com os silos, com os seus departamentos, enfim... até porque temos uma abordagem orientada ao produto, se calhar não tanto ao projeto. Portanto nós queremos ter pessoas de vários departamentos a trabalharem como equipas durante anos a fio. Portanto há também uma necessidade de consciencialização que a própria estrutura orgânica da empresa provavelmente vai mudar, e eu até vou um bocadinho mais fundo que é... eu acredito não só em *cross-functional teams*, mas acredito em *cross-functional people*, ou seja, já não são só as equipas que têm conjunto, que tem que ser capaz de entregar uma solução, mas uma pessoa cada vez mais tem que ser capaz de ter uma multiplicidade de *skills* que lhe permita desenvolver, testar, fazer análise porque isto é o que depois os torna realmente uma equipa. Portanto, eu acho que estes quatro aspetos, enfim, já permitem dar uma ideia dos pontos mais importantes que eu considero. Enfim, há uma lista infindável de mais tópicos, mas enfim, acho que para já, estes quatro são os mais importantes que eu usaria como obrigatórios, sem dúvida.

Questão 3/5

Luis Mattos: Em relação aos aspetos da ORGANIZAÇÃO, mais especificamente no âmbito da EQUIPA, quais são as características e comportamentos que julga que a equipa do projeto deve ter / apresentar para trabalhar com uma metodologia Agile?

Tiago Palhoto: Essa questão engloba um bocado do que eu pontuei na segunda questão. Essa última parte onde eu falei sobre as equipas *cross-functional*, elas próprias têm que ser capazes de como equipa entregarem uma solução, mas acima de tudo também tem que ser capazes, cada pessoa por si só, ter a capacidade de em qualquer momento "socorrer" outra. O espírito de equipa não é só "vamos todos jantar", é no aperto "Olha, eu já acabei as minhas coisas, quem é que precisa de ajuda?" E por muito que eu queira ajudar, se eu não tiver os *skills*, as coisas vão falhar, portanto, isso para mim é um aspeto muito importante. Não é um aspeto técnico, mas eu cada vez mais acho que os membros da equipe... eu acredito na especialização sem dúvida, mas acredito também no alargar os meus horizontes precisamente para conseguir dar uma ajuda em vários pontos e, para mim, os profissionais do futuro, um futuro que já é presente, são aqueles que apresentem por um lado uma especialização, mas que apresentem esta flexibilidade de conseguir em determinados momentos fazerem uma perninha do trabalho aqui, outra ali e conseguirem dar essa indicação. Depois obviamente como elementos de equipas sem dúvida, enfim, não vou citar as normas nem vários estágios no qual uma equipe passa porque isso certamente está na bibliografia, mas aquilo que eu considero importante desde o princípio, e eu procuro sempre fazer isso, é haver um alinhamento das expectativas e dos valores de cada membro da equipe porque para criar uma união é fundamental garantir que todos partilham um conjunto de valores idênticos. Nós que trabalhamos neste ambiente europeu com uma multiplicidade de culturas, temos pelo menos que garantir um alinhamento de valores estável, caso contrário, se isto não for garantido de partida, a coisa não vai funcionar. Para confiares em alguém e para alguém confie em ti tens que... ou seja, tu para confiares em mim tem que ter a noção de que eu partilho um conjunto de valores que são similares aos teus. Caso contrário, se calhar, nunca vais confiar em uma pessoa que não partilha dos mesmos valores básicos que tu e eu diria que este é um ponto fundamental. Afinal, confiança é algo que se conquista, não é? Eu para conquistar a tua confiança, se calhar tenho que partilhar alguns dos teus valores. Imagina que para ti é impensável chegar atrasado à uma reunião, enquanto que eu venho de uma cultura onde chegar 15 minutos atrasado é normal. Isto é apenas um exemplo, mas estás a ver que se não houver esse alinhamento as coisas podem não funcionar? Eu considero isso fundamental. A multiplicidade de *skills*, este alinhamento de valores, eu considero que são dois pontos fundamentais. São dois pontos fulcrais para começares a construir uma boa equipa. Depois obviamente temos o respeito, enfim, são coisas mais básicas não vale a pena estarmos aqui a enumerar, mas agora em nível mais concreto, em nível prático que eu tenho experimentado, esses são pontos fundamentais. Os *working agreements* vem um bocado nessa consequência, os *working agreements* que as equipas estabelecem são precisamente para garantir este alinhamento de valores, ou seja, termos a certeza de que vamos estar todos a falar a mesma língua porque muitas vezes o que esses *working agreements* fazem é ir imediatamente naqueles que saem um bocadinho fora da bitola da equipa, e quando isso começa a acontecer, a própria equipa acaba por começar a segregar aquele elemento. Portanto, eu penso que da experiência prática que eu tenho, seria aquilo que eu diria no fundo como elementos importantes para as equipas.

<u>Questão 4/5</u>

Luis Mattos: Em relação aos aspetos do PROJETO, mais especificamente no âmbito do AMBIENTE, quais são os fatores ambientais para o projeto que julga que são mais característicos das iniciativas candidatas a utilização de uma metodologia Agile?

Tiago Palhoto: Bem nós temos isso aqui na Comissão Europeia. Na metodologia Agile, que nós utilizamos como uma metodologia de gestão de projetos, acima de tudo, o que eu vejo e que pode não ser muito fácil, é que o conceito de projeto em si não é muito compatível precisamente com uma abordagem ágil. E por que digo isso? Porque quando temos uma abordagem ágil nós queremos uma equipa que se torne madura, *high-performance*, que demora para ser formada, mas que depois vai se conhecendo e vai se aperfeiçoando, o que é exatamente o oposto daquilo que é um projeto. Um projeto tem início e fim, e quando ele acaba, toda a gente vai embora e, portanto, aí temos um ligeiro problema, uma discrepância do modelo. Agora obviamente que eu considero que se conseguirmos, por exemplo, uma visão orientada ao produto, podemos claramente criar projetos que no fundo são iniciativas para o desenvolvimento e evolução desse produto que podem ser feitas por uma só equipe, e é um bocado o que se passa aqui. Por exemplo, eu tenho um projeto que tem um princípio, um fim e um conjunto de objetivos focados naquele produto, e temos aqui aquilo que já chamam a equipa de projeto, mas que se reparar, já estamos aqui a caminhar para uma abordagem mais ágil, ou seja, no limite, estamos focados num produto, mas como é dinheiro que tem que sair, um orçamento que tem que ser gasto por um período de tempo, nós criamos aqui novamente um *business case* e dizemos "Olha, agora nosso objetivo é fazer isso, isso, e isso porque agora o *solution environment*, o *solution context* mudou um bocadinho e, portanto temos agora outras prioridades. Vamos lançar um novo projeto no fundo para implementar, para melhorar aquele produto. É um bocadinho daquilo que já se começa a passar aqui. Portanto eu diria, acima de tudo, tentar garantir que se consiga, ainda que num ambiente de projeto onde tens princípio, meio e fim, ter equipas ligadas ao produto que elas sim vão desenvolver esse projeto e não tanto ter no projeto assim "começas um projeto e vais buscar novamente um conjunto de recursos etc." Isso agora é um bocado do modelo de entrega que nós temos aqui, pois nós vamos buscar recursos aqui aos poucos, o que acaba por ser um bocadinho contraditório com aquilo que nós queremos fazer, que é ter equipas que desenvolvam isso. Enfim, eu diria que este seria um aspeto fundamental. O outro claramente, e isso vejo pela experiência, em que tendo um projeto há sempre aquela mentalidade - e isto está um bocado relacionado com o ambiente - que há sempre alguém que é responsável, numa cultura de gestão de projetos, por tudo aquilo que é *output* daquele projeto, quando na realidade quem é responsável é a equipa e, portanto, há ainda muito esta... e a nossa metodologia também diz isto quando integramos o Agile dentro da nossa metodologia de gestão de projetos... o gestor do projeto é o *accountable* por todo o resto, ou seja, a equipa é responsável por fazer o trabalho, mas o gestor do projeto é que acaba por ser o *accountable*, e isto às vezes pode mudar aqui um bocadinho a forma como tudo se faz, como tu te relacionas com o cliente e com o *steering committee*, e obviamente que eu não estou à espera que vá uma equipa de dez pessoas para um *steering committee*, mas também não estou à espera que vá só uma pessoa, que é o gestor do projeto. Então eu acho que muitas vezes há esta questão que no fundo é fruto um bocado do ambiente onde estás, é esta cultura de que é o gestor do projeto que responde por tudo. Portanto, eu acho que isto é algo que também deve ser acautelado e é importante garantir que esta responsabilização, este *accountability* comece a mudar também para a equipa. O gestor do projeto pode estar a gerir um projeto que tenha desenvolvimento de software, mas depois tem uma parte de componente de processo e, portanto, tem que haver alguém que faz o *oversee* de tudo isto, mas tens várias pessoas a trabalhar em várias iniciativas dentro do projeto para implementar e para melhorar o produto. Eu acho que não devemos deixar só na cabeça do gestor do projeto toda a responsabilização. Eu acho que essa abordagem é algo que deve ser acautelada.

Questão 5/5

Luis Mattos**:** Em relação aos aspetos do PROJETO, mais especificamente no âmbito do PRODUTO, quais são as características relativas ao produto do projeto que julga que são mais relevantes na determinação de uma abordagem Agile como metodologia de desenvolvimento?

Tiago Palhoto**:** Um dos aspetos mais importantes que no fundo vem referido num dos princípios do Manifesto Ágil diz que são as equipas que determinam as melhores soluções para os problemas de negócio. Isso significa que deve de facto haver uma perspetiva de solução e manutenção desse produto, e isso está integrado no conceito de qualidade, que é claramente ser a equipa como um todo a definir a estrutura e o desenho de uma solução, porque é essa a equipa que a vai construir, essa equipa que vai desenvolver e essa equipa que vai manter a solução. Aquilo que é fundamental é tu criares uma solução, e para isso tens diversas técnicas de engenharia, nomeadamente o *Refactoring*, que é uma peça chave na técnica de TDD (*test-driven development*), e que serve precisamente para garantir a solução que tu estás a construir. A qualidade para mim não é uma aplicação com poucos erros, a qualidade vê-se na capacidade que essa solução tem para crescer no futuro quando queres adicionar novas funcionalidades. Isso sim é que para mim representa a qualidade de uma solução. É a facilidade com que vais lá e "toma lá mais uma pecinha" e está feito, isso obviamente sem dar escape do resto. Isso sim para mim é que é qualidade. É aquilo que eu considero mais importante. Ou seja, uma equipa que parte desde o princípio com esta preocupação de deixar uma solução devidamente estruturada, modularizada e com a capacidade de, em qualquer momento, poder adicionar coisas. Como eu costumo dizer, para mim o triângulo de ferro ("restrição tripla dos projetos") não é o âmbito, não é o tempo e não é o orçamento, isso para mim são apenas *constraints*, para mim o verdadeiro triângulo de ferro é o valor, é a qualidade, mais as *constraints*. O valor sendo a adequabilidade daquilo que eu entrego, a qualidade sendo a capacidade que a solução tem para crescer de forma indefinida, de forma fácil, quando eu quero adicionar outras coisas, depois vem as três *constraints*. Isso eu diria que é aquilo que eu considero importantes: estes três pontos do triângulo de ferro (âmbito, tempo e orçamento), e no centro a qualidade e o valor, que eu considero fundamentais como o resultado de um produto que tu entregas na sequência de um projeto. Ou seja, tem que ser valorizado pelo seu cliente e tem que ter a qualidade, o resto, são as tais três *constraints*. O princípio de utilizar o chamado *set by design*, isto é, deixar o máximo de opções em aberto, significa que quanto tu fazes esse trabalho inicial, nós temos uma tendência muito grande de achar que se reduzirmos ao máximo o número de opções estamos a ir pela solução correta, quando é exatamente o oposto. No início, o arquiteto de software deve ter uma visão, mas acima de tudo não deve começar logo a fechar todas as hipóteses porque há uma incerteza, com isso tu podes ter um bocadinho mais de trabalho adicional ao início, mas deixas algumas opções em aberto em termos de desenho e vais explorando, e à medida que vai tomando conhecimento, essas opções vão se reduzindo, mas muitas das vezes, o que nós vemos no início de cada projeto, é haver essa tentativa logo no início de reduzir todas as opções e limitar a flexibilidade. Nesse caso o que é que vai acontecer? Vais avançando e começas a ver que pensaste que a solução que tem que ser construída na realidade começa a divergir, e a uma altura vais dizer, "Bom, mas eu tenho que novamente me aproximar cá de cima", e quando estás tão afastado dali, já praticamente não consegues. Enquanto que se mantiveres um conjunto de opções em aberto, à medida que vais avançando, vais conseguindo perceber para onde é que tens que ir. Portanto, é fundamental ter uma equipa com qualidade, recursos com qualidade, nomeadamente neste tipo de abordagens, e creio que ainda mais do que aquilo que tem um projeto tradicional *waterfall*, e isso nós já sabemos e não vale a pena perder tempo com isso aqui, mas sem dúvida ter equipas não só multifuncionais, mas os próprios recursos serem multifuncionais, e recursos multifuncionais para mim são recursos com uma capacidade técnica muito acima da média, porque quem tem esta mentalidade para dizer "bom, eu já sou bom numa coisa eu agora quero divergir um bocadinho, quero expandir um bocadinho mais as minhas as minhas qualificações para poder ser mais flexível", daquilo que eu tenho visto até hoje, só aquelas pessoas com uma maturidade e com uma experiência muito grande naquilo que são muito bons é que depois começam a alargar estes horizontes. Portanto, sem dúvida, as equipas Agile têm que ter ali um core muito bom. Obviamente que há também os elementos mais juniores que são envolvidos, e faz parte também da cultura Agile essa tolerância precisamente ao erro e para receber as pessoas e para trazê-las para dentro da equipa. Enfim, resumidamente, a entrega do valor, a entrega da qualidade e no fundo garantir que tens uma equipa qualificada para fazer esse trabalho, seria mais ou menos o que eu diria sobre essa questão.

Parte II

Engloba as questões que irão explorar os aspetos relativos à escolha de uma metodologia Agile (dentre as quatro em estudo) alinhada às necessidades e ao ambiente do projeto de desenvolvimento de software. Tais questões foram elaboradas com base na revisão e análise de literatura sobre Agile.

Para as questões a seguir, foi considerado que estamos a falar das seguintes metodologias: Scrum; Extreme Programming; Lean Software Development; e Kanban

Questão 1/5

Luis Mattos**:** Em relação às mudanças nos requisitos do sistema, de que forma entende que cada metodologia aborda este aspeto? De que maneira o posicionamento de cada uma sobre a utilização de ciclos de desenvolvimento (iterações) pode influenciar neste aspeto?

Tiago Palhoto**:** Em nível de alterações dos requisitos do sistema, obviamente que o Scrum por ser reconhecido por ter esta cadência, ter esta preocupação que no fundo é um bocado da instanciação dos princípios básicos do Agile Manifesto, uma preocupação com a interatividade, em recolher feedback, em adaptar e etc., e que obviamente é o *core* dessas metodologias para tudo que tem a ver com os requisitos e as alterações de requisitos porque é precisamente isso, quanto a mim, é o (Scrum) que melhor instância esses princípios que estão relacionados com a capacidade de adaptação, com inspecionares e adaptares o teu processo, com a flexibilidade que tens de mudar e tirar e pôr e etc. A utilização destes ciclos curtos permite efetivamente saber se vais na direção certa, e na direção certa precisamente estás a entregar os requisitos que tem mais valor. O Extreme Programming, quanto a mim, eu diria que não está tão relacionado com as mudanças de requisitos de sistema, enfim, eu acho que está mais relacionado com a otimização do trabalho da equipa enquanto desenvolvimento, independentemente de serem alterações de requisitos ou não. Sobre o Kanban eu diria a mesma coisa. Eu vejo o Kanban muito ligado ao Lean Software Development. Um dos cortes que eu gosto de tirar do Lean é toda a parte de redução do desperdício, de evitar saltar tarefas e etc., tudo isso mediado pela ferramenta, pelo próprio Kanban *board*, portanto, eu vejo que no que toca aos requisitos, eu vejo estes três um bocadinho mais generalista. Eu vejo o Scrum, aliás não é por acaso que quando utilizas o Scrum utilizas também umas partes do Extreme Programming, por exemplo, tens o *Pair Programming* onde a aplicação é codificada em conjunto, mas numa equipa que aplica o Scrum, portanto eu diria que o Scrum é aquela que efetivamente, como base, dá o esqueleto, dá a estrutura que me permite lidar com as mudanças nos requisitos do sistema. Quanto ao resto, acho que vai ser mais na parte da utilização das práticas de engenharia de software.

Questão 2/5

Luis Mattos**:** Em relação à utilização de práticas de engenharia de software, de que forma cada metodologia aborda este aspeto? Há alguma que dê mais ênfase à utilização de boas práticas de engenharia de software (em detrimento das práticas de gestão do projeto)? Se sim, que razões julga levar a esta necessidade?

Tiago Palhoto**:** O Extreme Programming nasceu precisamente numa tentativa de otimizar e melhorar toda a parte do código de desenvolvimento do produto e não está tão relacionado com a gestão do projeto e etc. Algumas dessas práticas de engenharia de software começaram a ser utilizadas no Extreme Programming, mas sem dúvida uma das que eu retiro mais efetivamente do Extreme Programming é claramente o *Pair Programming*. Já vi várias equipas a fazerem, algumas mal outras bem, mas mesmo as que fizeram mal começaram mal e depois acabaram bem, mas as que fizeram bem, de facto nota-se uma diferença muito considerável porque leva ao *ownership* da solução. A capacidade que tem duas pessoas de verem o que estão a fazer... a diminuição de erros que possam ser introduzidos na aplicação em termos de lógica e etc. é muito menor, e obviamente possibilita ganharem imenso tempo mais à frente. O próprio TDD, enfim, tem um papel fundamental na qualidade do código. O maior envolvimento dos desenvolvedores na parte de testes junto à área do negócio porque eles também têm que começar a perceber um bocado do negócio, enfim, tudo isso efetivamente torna o Extreme Programming com uma metodologia, quanto a mim, muito importante no que diz respeito às práticas de engenharia de software e que trouxe claramente uma melhoria substancial nessa parte. Relativamente ao Lean e ao Kanban, eu vejo os dois muito juntos. O Lean eu vejo um bocado sempre orientado à otimização do processo. Que normalmente tu consegues ver através da sequência de passos que normalmente eu associo sempre a um Kanban *board* e por isso ando sempre com eles muito ligados. Obviamente que estas são técnicas que no fundo cobrem um bocadinho da gestão, mas também das práticas de engenharia de software. Isso porque se eu estiver numa perspetiva de *DevOps* e se aplicar todos os meus passos, enfim, *continuous integration*, *continuous* *deployment* e *continuous delivery* eu posso utilizar um Kanban e posso utilizar o Lean para fazer no fundo os meus *values streams* e identificar todos os passinhos que tenho em todo este meu ciclo, em todo o meu pipeline, e identificar onde é que eu estou a gastar e onde é que eu estou a perder mais tempo. Ou seja, eu estou a otimizar a utilização das minhas práticas de software. Eu estou a arranjar forma e identificar onde é que eu posso otimizar, ou seja, o Kanban e o LEAN não se focam na utilização das práticas em si, mas focam em mostrar, por exemplo, todo este processo que utiliza as várias práticas. Onde é que eu posso otimizar. Ou seja, eu também considero, isto não só o Lean, mas também outros processos relacionados com a gestão do projeto. Portanto, de uma forma direta, eu diria que o Extreme Programming de facto tem um impacto direto com a gerência e com as práticas de engenharia de software. Eu diria que o Lean e o Kanban por me darem essa visibilidade a vários níveis, ao nível das práticas de software e mesmo a nível de gestão de projeto, me permitem ter esta visão e perceber onde é que eu posso otimizar e eventualmente que outras práticas de engenharia de software eu possa vir a utilizar. Quanto ao Scrum, o Scrum eu vejo como um *placeholder*, uma estrutura onde depois eu possa encaixar tudo isto, portanto, de uma forma genérica, o Scrum acaba por ser o meu *placeholder*, o Extreme Programming acaba por ser aquilo que de facto me entrega as técnicas, como eu referi as práticas importantes de engenharia de software, e o Lean e o Kanban permitem-me olhar para a forma, quer para a estrutura que eu tenho do Scrum, quer para a forma como eu tenho montado as minhas técnicas de engenharia nos vários processos que permitem perceber onde é que eu posso otimizar isso, portanto, tem uma utilização mais indireta nesse sentido. Portanto, acho que essa forma seria mais ou menos o que eu responderia nesse aspeto.

Questão 3/5

***Luis Mattos*:** Em relação ao engajamento do cliente no projeto, o quarto princípio do Manifesto Agile ressalta a necessidade de maior interação e colaboração entre a equipa de projeto e o pessoal de negócio (cliente). Todas as metodologias Agile aplicam, em diferentes níveis, este princípio. Com base no vosso conhecimento destas metodologias, de que forma julga que estas metodologias se diferenciam relativamente a este princípio?

Tiago Palhoto**:** Assim, o Extreme Programming, da experiência que eu vou tendo, per si, apesar de fomentar a utilização de *User Stores* e etc., não é ele que promove esta interação com o cliente. Eu diria mais, eu diria que o Scrum propriamente dito, por ser uma *framework* já estrutural, e que tem um conjunto grande de artefactos, tem um conjunto de cerimônias que promovem precisamente, nos momentos chave, maior interação com o cliente. Portanto, eu não tenho dúvida nenhuma que o Scrum, de uma perspetiva metodológica, de uma perspetiva organizacional estrutural de gestão, falo não numa gestão na parte das expectativas, mas em termos daquilo que nós esperamos em termos da interação com o nosso cliente, faz esse trabalho na perfeição. Depois eu diria que o Lean, o próprio Kanban, enfim, pode efetivamente, quando estás a fazer análises de determinados processos, obviamente que promove pela necessidade de envolvimento do cliente enquanto estás a identificar os *value streams* e queres perceber o processo e como é que podes otimizá-lo e etc. Naturalmente que envolve também o cliente, mas eu diria claramente que o Scrum é aquele que promove de forma visível e formal mais claramente o envolvimento do cliente no projeto. Agora, há uma coisa que eu não vejo, que é um foco na gestão das expectativas e da comunicação das expectativas, que no fundo tem a ver um bocado com a explicação das metodologias que são utilizadas. Só para concluir, o que acontece? Eu até posso dizer ao meu cliente que vou aplicar Scrum e nós vamos ter estas cerimônias. Eu até posso dizer, "Olha, tu aqui tens que fazer isto, isto e isto", mas se eu não te explicar efetivamente que tens que fazer isto e como é que se faz, eu não estou a gerir as tuas expectativas de forma adequada. Portanto, quando se diz que "*Business people and developers must work together daily throughout the project"*, isso é completamente verdade. Agora, se eles (utilizadores) não tiverem a noção do como e do porquê que fazer as coisas, isso vai cair em saco roto. E o Scrum, como eu referi formalmente, identifica todos os momentos, mas não me diz exatamente quem (deve) nem como gerir essas expectativas. Não creio também que as outras restantes metodologias que nós identificamos aí se foquem muito nesse aspeto. Portanto, eu diria que o Scrum é aquele que torna claramente mais evidente e mais formal e mais acessível esta necessidade que traz mesmo o cliente para perto. Ele (Scrum) formaliza os *Sprint Reviews*, que é onde tu vais mostrar aquilo que estás a fazer, e isso também é uma forma de trazer o cliente para mais próximo de ti. Obviamente que são as técnicas de desenvolvimento que tu usas, eventualmente as *User Stories* que são promovidas, e eu penso que as *User Stories* são também promovidas no Extreme Programming e foi um dos pontos iniciais de utilização até mais do que no Scrum, mesmo porque o Scrum Guide sequer falas em *User Stories*, enquanto que no Extreme Programming, por outro lado, lá está escrito por todo lado. As *User Stories* são também de facto uma forma simples de comunicares e de conversares com o cliente e também são uma forma de aproximação. Se calhar eu diria que o Scrum, em primeiro lugar, mostra de forma clara e evidente os momentos onde isso deve ocorrer, e depois eventualmente o Extreme Programming, que na sequência da utilização usa as *User Stories* como uma forma muito fácil para conversar com o cliente e de ao mesmo tempo manter o âmbito do projeto devidamente organizado. Eu diria que estas duas são aquelas onde eu vejo mais impacto esse aspeto. O Lean e o Kanban, per si, não creio que sejam metodologias com foco em promover o envolvimento do cliente nos projetos.

Questão 4/5

***Luis Mattos*:** Em relação à abordagem para entregas do software (ou de partes do software), o terceiro princípio Agile fundamental diz "*Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.*" Em Agile, este princípio é utilizado amplamente, porém, tratado de forma específica por cada metodologia. Em relação aos aspetos identificados a seguir, com base no seu conhecimento destas quatro metodologias, como diria que cada uma os aborda?

- *Continuous delivery* – como forma de obter uma cadência efetiva e determinar uma velocidade "padrão" para as entregas;
- *Deliver as fast as possible* – o foco é a velocidade da entrega de valor ao cliente;
- *Minimum viable product (MVP)* – a conceção de uma versão inicial simplificada e funcional do software.

Tiago Palhoto**:** Então, começando pelo Scrum, obviamente o *continuous delivery*, e se aí estivermos a falar numa perspetiva de "não efetivamente entregue para produção", o Scrum obviamente promove a continuidade ao fazer uma entrega e me permite ver o meu produto crescer, o valor a crescer e obviamente que ele permite ver isso claramente e isso faz parte da própria natureza do Scrum. *Delivery* não *releasable*. O Lean não, pois ele lá está mais voltado para otimizar esses processos. O Extreme Programming, da forma também como faz a abordagem aos testes, a forma como também promove a integração constante do código, que é um fator fundamental para *continuous delivery*, eu diria que estes dois são aspetos fundamentais nestas, enfim, nestas duas metodologias que abordam de uma forma muito mais pragmática este elemento. Agora, por exemplo, "*Delivery as fast as possible*", eu aqui considero que quer o Lean, quer o Kanban, são duas ferramentas fundamentais para te ajudar a perceber se efetivamente tu estás a fazer da melhor maneira. Tal como referimos há pouco, o Lean dá-me um conjunto de recomendações e práticas que me permitem tentar otimizar, e no fundo ser mais eficaz, o meu processo de entrega e, ao mesmo tempo, usando ferramentas *value streams* que me permitem monitorizar o meu processo e perceber onde é que eu posso otimizar precisamente numa perspetiva de entrega o mais rápido possível. E eu posso dizer que o entregar o mais rápido possível não quer dizer apenas entregar mais cedo. Eu posso ter a mesma data de entrega, mas entregar mais coisas e portanto isso para mim também é entregar mais rapidamente, e isso também é nessa perspetiva do que eu estou a falar. O Scrum obviamente permite-me controlar isso, ou seja, eu se reduzir as minhas iterações vou estar, a partida obviamente, a entregar mais frequentemente, mais rápido, mas obviamente se calhar entrego em blocos menores. O Scrum também formaliza isso. É uma ferramenta excelente para formalizar e dizer que em vez de a cada duas semanas, que todas as semanas vou entregar, portanto estou a dizer, vou entregar o mais rápido possível, mas é apenas uma formalização, o Scrum per si não controla se efetivamente vais entregar mais ou não, mas está ali, está a te dizer "vamos trabalhar nesta cadência e portanto pessoal orientem-se, mas tem que entregar qualquer coisinha." Agora, considero que as técnicas que tens efetivamente no Extreme Programming podem te contribuir para isso, apesar de que eu considero que quer o Lean, quer o Kanban, são aqueles que podem ajudar efetivamente a melhorar todos os seus processos, inclusive em um processo de *DevOps* que tenhas montado. Precisamente isso permite ajudar a entregar o mais rápido possível. Todo esse *delivery pipeline* eu considero que de fato o Kanban, por ter uma visualização adequada nomeadamente da limitação do teu WIP, e no fundo evitares precisamente a existência de gargalos, quer o Kanban, quer o Lean, eu acho que tem um papel fundamental para te ajudar nessa matéria porque ajudam claramente a perceber onde é que podes melhorar, sem dúvida. MVP, a conceção de uma versão inicial simplificada e funcional do software, quanto a mim sem dúvida que o Scrum é uma perspetiva exatamente disso, aliás, uma das comparações que tens com o *waterfall* é que ele só no final do produto desenvolvido é que tu vais entregar, enquanto o Scrum, com esta abordagem interativa, claramente permite começares desde cedo a entregar valor e permite imediatamente "Bom, eu ao fim dessas iterações já vou ter este bocadinho aqui, epa, isto é um MVP e eu vou entregar." Portanto, o Scrum sem dúvida é que me dá, que me formaliza essa forma de pensar. Quanto ao resto, eu vejo que já são ferramentas mais práticas que podem ajudar a trabalhar na implementação de um MVP. Eu acho que aqui estamos a falar mais de uma questão conceptual e processual e dizer olha, qual dessas mitologias de fato está mais focada na construção, ou seja, na entrega e na criação de MVP’s e trabalhar com MVP’s. Eu penso que o Scrum, pelo fato de ser uma *framework* que abrange tudo isto, vai nesse sentido. Claro o Kanban, se tu estiveres a trabalhar em modo suporte, por exemplo, ou seja, que não tens interações, também te pode estar a dizer "Olha, temos aqui um lote da próxima inovação, o próximo MVP vai ser a correção de todos estes elementos que temos aqui. Não interessa. Não estamos a falar de sprints. O próximo MVP queremos lançar com isto tudo corrigido." Daí eu vou ter um Kanban board que vai avançando, onde cada uma delas (correção) vai chegando e eu vou fazendo, vou fazendo, sempre respeitando obviamente os meus WIP’s. Obviamente que nesse caso o próprio Kanban também funciona dessa maneira, e isso é uma das razões da diferença entre o Scrum e o Kanban: enquanto o Scrum está a trabalhar incrementalmente, enfim, por períodos para mostrar progresso e etc., normalmente o Kanban é usado numa perspetiva contínua e não está preocupado com interações, está sempre a entregar valor e daí, enfim, o "*delivery as fast as possible*". O próprio Kanban também acaba por promover isso, pois o time está preocupado em que venha outro, venha outro, venha outro, e nesse sentido eu diria sim, no MVP, eu diria que quer o Scrum, quer o Kanban, são duas formas de organizar a forma como entregas a trabalho, acho que seriam os dois mais relevantes e que tem impacto nessa matéria.

Questão 5/5

***Luis Mattos*:** Em relação à priorização de requisitos do sistema, o primeiro princípio Agile diz "*Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.*" As metodologias Agile trabalham no sentido de priorizar a entrega das funções do produto que trarão maior valor para o negócio (sob o ponto de vista do cliente). Entretanto, cada metodologia trabalha com esta priorização de formas diferentes. Com base no vosso conhecimento das quatro metodologias Agile em discussão, de que maneira cada uma delas trabalha a priorização de *features* e atividades no âmbito de um projeto de desenvolvimento de software?

***Tiago Palhoto*:** Quanto ao Extreme Programming, eu não tenho experiência na utilização do Extreme Programming só nessa vertente. Eu sempre utilizei o Extreme Programming acoplado a outra *framework*, nomeadamente o Scrum, ou seja, nunca utilizei só o Extreme Programming, portanto, não estou a ver o Extreme Programming por si só na parte da utilização dos requisitos do sistema. Aqui o que vejo claramente é o Scrum, obviamente pela forma como lida e como gere todo um *backlog*, que já ali é nativo. O próprio Scrum fala nisso, o papel do *Product Owner* deixa isso bem claro e, portanto, não tenho dúvida de que o Scrum instância muito bem esse princípio. O próprio Kanban também funciona dessa forma, ou seja, eu tenho aqui um ótimo trabalho apenas eu não vou é trabalhar em iterações, mas também tenho alguém que me diz "Olha estas coisas estão aqui a chegar e agora qual é que vem? Vem esta, vem esta e vem esta." Podes ter priorização ou não, mas efetivamente a preocupação é que vais puxando sempre trabalho e muitas vezes com alguém que vai pondo as coisas em ordem e tu vais apenas puxando esse trabalho. Aliás, não é por acaso que o próprio Scrum dentro de cada iteração usa um Kanban *board*, enfim eles estão intimamente ligados. Quanto ao Lean, ele é mais utilizado numa perspetiva de otimização e redução de desperdício, então aqui, o que eu posso dizer é, a priorização de requisitos do sistema é uma coisa e a forma como nós depois lidamos com isso e como estamos constantemente a alterar atividades é outro. O *context switching* é uma das grandes formas de desperdício e, portanto, dá aqui há algumas indicações. As prioridades são giras, mas se estás constantemente a mudar as coisas enquanto nós estamos a trabalhar em algo vamos ter problemas, pois vamos estar aqui a perder um bocadinho da eficiência, ou seja, eu diria que o Lean pode te dar aqui algumas indicações relativamente não a priorização dos requisitos do sistema, mas eventualmente na forma como tu tratas a mudança de requisitos, de uma forma constante e etc. Pode te dar algumas indicações do quão prejudicial isso pode ser. No entanto, eu considero sem dúvida, que o Scrum, e o próprio Kanban, desde que obviamente haja alguém que faça essa priorização e limitação dos WIP's, ou seja, garantindo que não estás esgotar a tua capacidade em cada fase do processo, eu diria que são aquelas que, pelo menos da minha experiência, em termos de utilização são aquelas que satisfazem mais no que diz respeito à priorização dos requisitos do sistema.

Parte III

Levantamento dos requisitos funcionais do sistema de informação que será projetado a fim de implementar o modelo de decisão derivado da análise dos aspetos discutidos nas Partes I e II. O objetivo é coletar a opinião do entrevistado sobre quais seriam as funcionalidades ESSENCIAIS para este sistema de informação.

Questão 1/1

Luis Mattos: Um sistema de informação será desenvolvido de forma a implementar o modelo de decisão resultante da análise dos aspetos abordados nas Partes I e II desta entrevista. O objetivo principal do sistema é indicar se um dado projeto de desenvolvimento de software é candidato a uma abordagem Agile e, em caso positivo, indicar qual entre as quatro metodologias Agile mencionadas na Parte II desta entrevista é a mais indicada. Neste contexto, quais seriam os requisitos funcionais que julga ESSENCIAIS para este sistema?

***Tiago Palhoto*:** Eu faria um sistema desses baseado um bocadinho num *assessment*, num questionário. Agora, em termos de funcionalidades obviamente eu procuraria focar um bocadinho, como falamos, nos aspetos fundamentais que eu dividiria, se calhar, em primeiro lugar, ajudar a decidir que metodologias usar, eu pessoalmente não usaria isso numa abordagem exclusiva, mas numa abordagem onde podes usar uma em combinação com outra, não exclusiva, porque a realidade mostra isso mesmo. Portanto, esse seria um dos meus primeiros pontos. Eu jamais limitaria ao A, ao B ou C, mas com base no resultado poderias eventualmente dizer, vais utilizar isso e vais utilizar também aquilo. Eu veria essencialmente funcionalidades com foco pelo menos em alguns aspetos... um claramente em perceber a maturidade do contexto organizacional onde tu estás inserido. Isso por si só já toca em alguns aspetos que o consideraria fundamental abordar. Agora, eu destinaria sempre a nível do aspeto de três ou quatro secções, mas uma delas relacionada claramente com o contexto organizacional, nomeadamente ao nível das pessoas, a nível de competências ou características de liderança, mas também a nível da própria estrutura orgânica, a nível da própria estratégia da organização. Isso seria um ponto. O outro ponto seria a nível da própria realidade atual da empresa. No fundo é buscar saber onde estão, qual é a situação atual, ou seja, se os projetos estão atrasados ou não, se tem perdas ou não tem perdas, enfim se há novas equipes e etc. Tudo isto são aspetos. Na segunda parte eu procuraria saber qual é a realidade prática atual em termos de equipas, ou seja, o que usam como ferramentas e o que não usam, se esta questão de pensar em mudar vem das equipas, se vem daí a liderança e etc. Portanto, tentaria perceber também aquilo que tentaste perceber ao nível da organização e alto nível estratégico. Ter esta visão ao nível das equipas, enfim, de todos aqueles que estão envolvidos no desenvolvimento dos produtos, mas ainda de uma perspetiva comportamental ambiental. Depois tentaria eventualmente perceber quais é que são atualmente as dores deles em termos de entrega, ou seja, quais são as dificuldades, porque no fundo o que eles querem é desenvolver software e entregar, portanto, é importante ter aqui um conjunto de campos onde tentas perceber qual é o vosso problema: Vocês entregam em atraso? Entregam aquilo que é realmente pretendido? Entrega o é pretendido dentro do prazo, mas com muitos erros, enfim, não tanto focar em coisas objetivas de "estar a usar isso ou estás a usar aquilo", "usas estes gráficos ou não usas" porque isso significa que estás a usar algo, mas não significa que estás a usar bem. Portanto, em termos de requisitos funcionais, a minha questão será sempre com base em questionário, em inquéritos, não vejo outra forma. Agora, em termos de orientação de perguntas, principalmente nesta parte agora, na parte mais prática de perceber quais são os reais problemas porque é isto que vai começar a ajudar a perceber quais dessas metodologias podem ajudar a resolver, não é perguntar "vocês fazem interações ou querem fazer interações?", "Bem, queremos fazer iterações", mas isso não te diz porque é que eles querem fazer ou porque que eles acham que querem fazer iterações. No fundo tu podes estar a dar uma sugestão com base naquilo que eles acham que querem quando na realidade, se calhar, é interessante termos a certeza que sabemos o que é que eles querem para que o diagnóstico venha correto. Se eu apenas perguntar, "vocês têm um *product backlog* atualizado?" Eles podem dizer que sim sem saber o que significa estar atualizado. Enquanto eu diria, "olha, o teu *product backlog* reflete exatamente aquilo que o teu cliente quer? Foi ele que disse exatamente que é isto que quer? Ou seja, não estou a dizer que são essas as perguntas, mas a abordagem tem que ser numa perspetiva de questionar da forma mais natural... no sentido de questionar o problema e a necessidade e não tanto os usos dos *information radiators* ou os próprios recursos que nós disponibilizamos para resolver as coisas, ou seja, é deixar as perguntas de um forma a levantar o problema e não propriamente dizer, "olha, usas isso como fonte de soluções para os problemas" e tentar perceber "se eu não uso isto é porque tenho este problema, se eu não uso aquilo é porque tenho aquele problema, ou se uso isto é porque não tenho aquele problema." Isso não é verdade porque ao usarmos podemos estar a usar erradamente, ou seja, é entendermos a necessidade, focarmos no problema a ser resolvido e não nas ferramentas que existem para solucioná-lo. Muitas vezes eu vejo questionários muito focados em avaliar se se está a seguir o processo, tipo um *checklist*, agora, isso não indica se o processo está a funcionar ou não. Isso é válido numa primeira abordagem onde só queres saber se estão a seguir um processo. No caso desse sistema, as questões devem ser mais voltadas para identificar claramente se se faz isso ou não, e se faço isto significa que tenho mesmo um problema e não estar a "Olha, tens esta solução, não tens e se tens essa solução é porque já resolveste aquele problema." Agora, o que que isso significaria em termos de funcionalidades, eu vejo algo seriamente orientado a um questionário, claramente, a nível de ponderação também, se calhar, podes ter um conjunto de respostas dadas que podem ser de tal forma ponderadas que podem logo dizer que com isto, independentemente do que o resto das pessoas disserem, a metodologia mais adequada é o Scrum. Ou se eles disseram, "Nós temos constantemente processos que nunca estão otimizados, nós aqui vamos ter que aplicar claramente o Lean, apesar de as equipas não verem que estão em desperdício, nós cá em cima temos essa noção e portanto vamos tentar propagar isso. Portanto, aqui não sei te dizer se faz sentido ou não haver uma ponderação no peso das respostas. Portanto, não é uma pergunta fácil, mas neste campo é isto que eu te consigo responder.

--- // ---

Appendix D – Interview Transcript with Agile Specialist [English]

Interview data	
Interviewer	Luis Fernando Bordeaux Mattos
Interviewee	Tiago Palhoto – Agile coach and responsible for the Agile@EC methodology, applied in software development and maintenance projects in the European Commission.
Location	Brussels, Belgium.
Date and time	24th of January 2019.
Duration	01h11min
Comments	The language used in the interview was Portuguese (Portugal / Brazil). This is a version translated to English.

Part I

It encompasses questions that will explore the aspects relating to choosing an Agile approach as a guide to the management and execution of a software development project. The objective is to collect the impressions and opinions of the interviewee on the pertinent questions formulated. These questions have been elaborated based on the review and analysis of Agile literature.

<u>Question 1/5</u>

Luis Mattos**:** Under what conditions would you suggest an organisation to opt for an Agile methodology?

Tiago Palhoto**:** There are essentially two points here. The first is to have reached what it's called the tipping point, that is, the moment when you say, "We have to change, period." So, it's something that the organisation feels and says... "This cannot go on like this. We are not improving. It is proven this is not working. We really have to change." So, that's one of the main reasons to change. This is even addressed in the *Scaled Agile Framework* itself as one of the points in which the decision to change and use an Agile approach is encouraged. The second point is effectively when you have people, I mean... the first has to do with a need that is visible to everyone, the second has to do essentially when you have a group of people who already have this perception, who already have this notion of what Agile is. So, they themselves want to... "Let's try this approach because we have here a group of people who already know the approach." So, they are people who already have the basic principles, already “infected” a bit with some fundamentals and principles of Agile and, therefore, you have a set of people there to take the first step. This second obviously ends up being a little help to the first, but above all, I believe the first one, from my experience, really ends up being the one that brings a greater will at the beginning, but does not dispense the second, okay? While if you have the second effectively coming from the top executives of the company, it may be easier to implement, but on the other hand it may be a bit harder to break the initial resistances. But I consider that these two points are the conditions under which I would suggest to an organisation the adoption of Agile.

Question 2/5

***Luis Mattos*:** Regarding the aspects of the ORGANISATION, more specifically within the MANAGEMENT level, what are the characteristics and behaviours that you think the leaders of the organisation should have to work with an Agile methodology?

***Tiago Palhoto*:** There is one that, in my opinion, is essential, which is tolerance. Tolerance in the sense of understanding that such a change obviously brings experimentation, it leads to errors. People will make mistakes, and these mistakes will be part of your daily work, and this is good. It's a mindset change for people that is not easy... "But then we're going to try one thing, and we're going to make mistakes, and we're going to say that we're wrong and so on?" Yes, it has to happen. There is a second point that is to move from a command and control approach to trusting the people who are really going to get the job done. This is key. Everyone knows, but few apply. That is, I have teams, people, professionals which I assign to perform a task, to build a system, whatever, so they are the best people to perform this mission. I have to trust them. The third point is to create and ensure that working conditions are created so that people are really proud and enjoy what they do. Usually, the first people to leave the company are those who do not like what they are doing and who do not feel pride in what they do. Therefore, if we do not create an atmosphere, if we do not create conditions for not only letting people work but above all, letting them use their creativity and feel usefulness and sense of what they are doing, it will not work. Of course, this is not only in Agile approaches but in a general way. When we talk about giving teams freedom, it's giving them the freedom to use their creativity and do things that are really useful. The first point I mentioned was precisely about that, if we do not have that efficiency, that capacity of tolerance, to err immediately, we'll be limiting the creativity, and then it's a chain reaction that ends up limiting the team's initiatives. These, in my opinion, are three fundamental points. Obviously, there is another aspect that has to do with something more organic, that is to realize that Agile ends up a little with this notion of very specific skills. That is, I can even go a little deeper into this, because we started with an approach where we create cross-functional teams and, therefore, It's an approach that ends up not being very compatible with very rigid organic structures, with silos, with their departments, and that is because we have a product-oriented approach, maybe not that much a project-oriented one. So, we want to have people from various departments working as teams for years, and there is also a need for awareness that the organisational structure of the company itself is likely to change. I can even go a little deeper on that... I believe not only in cross-functional teams, but I believe in cross-functional people, that is, it is no longer just the teams that have to be able to deliver a solution, but the members of the team by themselves must have a multiplicity of skills that allows them to develop, to test, and to analyse, because this is what makes high-performance teams. Therefore, I believe these four aspects allow us to have an idea of the most important points to consider. There is an endless list of more topics, but anyway, I think by now, these four are the most important that I would use as mandatory, no doubt.

Question 3/5

Luis Mattos: Regarding the aspects of the ORGANISATION, more specifically within the TEAM, what are the characteristics and behaviours that you think the project team should have / present to work with an Agile methodology?

***Tiago Palhoto*:** This question covers a bit what I said in the previous question. That last part where I mention cross-functional teams, they have to be able to, as a team, deliver a solution, but most of all, they also have to be able, each person by itself, to have the ability to, at any moment, help a colleague. The team spirit is not just "having dinner together", it's in the grip "Look, I've already finished my stuff, who needs help?" And as much as I want to help, if I do not have the skills, things will fail, so, this is, in my opinion, a very important aspect. It is not a technical aspect, but I really believe that team members... I believe in specialization of course, but I also believe in widening horizons precisely to be able to give an aid in several points and, for me, the professionals of the future, a future that is already present, are those that present on one hand a specialization, but also present the flexibility of being able to, at certain moments, make a piece of work here, another piece there and provide that indication. Obviously as team elements, I will not mention the norms or the various stages in which a team passes because that is certainly in the bibliography, but what I consider important from the beginning, and I always try to do, is to have an alignment of the expectations and values of each team member, since to create identity, it's essential to ensure that everyone shares a set of identical values. We, who work in this European environment with a multiplicity of cultures, we must at least ensure a stable alignment of values, otherwise, if this is not guaranteed in the beginning, it won't work. To trust someone and someone to trust you, I mean… you to trust me, have to have the notion that I share a set of values that are similar to yours. Otherwise, maybe you will never trust a person who does not share the same basic values as you and I would say that this is a key point. After all, trust is something you gain, isn't? To win your trust, maybe I have to share some of your values. Imagine, for instance, that for you it is unthinkable to arrive late to a meeting, whereas I may come from a culture that arriving 15 minutes late is normal. This is just an example, but can you see that if there is no such alignment, things may not work? I consider this essential. The multiplicity of skills, this alignment of values, I consider them fundamental points. They are two key points to start building a good team. Then obviously we have the respect, anyway, are more basic things are not worthy to be listed here, but now, on a more concrete level, on the practical level that I have experienced, these are fundamental points. The working agreements come a lot in this consequence. The working agreements that the teams establish are precisely to guarantee this alignment of values. That is, to be sure that we will all be speaking the same language, because often what these working agreements do is to go immediately to those who leave a little outside the gauge of the team, and when this begins to happen, the team itself begins segregating that element. Therefore, I think that, from the practical experience I have, these would be the important elements regarding the teams.

Question 4/5

Luis Mattos**:** Regarding the aspects of the PROJECT, more specifically in the ENVIRONMENT, what are the environmental factors for the project that you consider to be more characteristic of initiatives eligible to use an Agile methodology?

Tiago Palhoto**:** Well, we have a bit of this here in the European Commission. In Agile methodology, which we use as a project management methodology, above all, what I see, and which may not be very easy, is that the project concept itself is not precisely compatible with an Agile approach. And why do I say that? Because when we have an Agile approach, we want a team that becomes mature, high-performance, which takes time to be formed, but that improves over time, which is exactly the opposite of what a project is. A project has a beginning and an end, and when it is over, everyone leaves and, therefore, we have a slight problem, a discrepancy in the model. Now, of course, I think that if we can achieve, for example, a product-oriented vision, we can clearly create projects that are basically initiatives for the development and evolution of this product that can be done by a single team, and this is a bit what happens here. For example, you have a project that has a beginning, a defined end, and a set of objectives focused on that product, and you have what we call "the project team", but if you notice, you already have moved towards a more agile approach, that is, in the limit, we are focused on a product, but since there is a budget that has to be spent for a period of time, we have created a business case again, and we say, "Look, now our goal is to do this, this, and this because the solution environment, the solution context has changed a bit, so we have other priorities." It is a little bit of which we already got here. So I would say, above all, trying to ensure that this is achieved, even in a project environment where you have a beginning, a middle and an end, to have teams linked to the product that they are going to develop in the project, and not that much the case when "you start a project, and you are going to get back a set of resources, etc." This is a bit of the delivery model that we have here because we get resources here gradually, which turns out to be a bit contradictory with what we want to do, that is to have teams that develop this. Anyway, I would say that this would be a key aspect. The other clearly, and this I say from experience, that having a project there is always that mentality – and this is a bit related to the environment – that there is always someone responsible, in a project management culture, for all outputs of that project, when in reality the responsible is the team and, therefore, there is still a lot of this... and our methodology also says this when we integrate Agile into our project management methodology... "the project manager is accountable for everything else", that is, the team is responsible for doing the work, but the project manager is accountable, and this can sometimes change here a little bit the way everything is done, the wat you interact with the client and with the steering committee, and obviously, I'm not expecting a team of ten to go to a steering committee, but I'm not also expecting only the project manager to go. So, I think there is often this issue that is basically due to the environment where you are; it is this culture where is the project manager that accounts for everything. Therefore, I think this is something that should also be taken care of, and it's important to ensure this accountability begins to change also for the team. The project manager may be running a project that includes software development, but also has a process component part, so, it's important someone overseeing all of it, but you have several people working on several initiatives within the project to improve the product. I believe we should not just leave it for the project manager to hold all accountability. I think this approach is something that needs to be taken care of.

Question 5/5

Luis Mattos**:** Regarding the aspect of the PROJECT, more specifically in the PRODUCT scope, what are the characteristics related to the product of the project you think are most relevant in determining an Agile approach as a development methodology?

Tiago Palhoto: One of the most important aspects mentioned in one of the principles of the Agile Manifesto says that the team determines the best solutions to the business problems. This means that there must be, indeed, a perspective of solution and maintenance of this product, and this is integrated into the concept of quality, which is clearly the team as a whole to define the structure and design of a solution, because the is the team who will design, develop, and maintain the solution. What is essential is that you create a solution, and for this, you have many engineering techniques, namely the Refactoring, which is a key part in TDD (test-driven development) technique, and that serves precisely to guarantee the solution that you are ramping up. The quality for me is not an application with few errors, the quality, in my opinion, is the ability that this solution has to grow in the future when you want to add new features. That is what represents the quality of a solution. It's how ease you go there and add one more piece, and it's done, this obviously without letting the rest escape. That is what quality means. It is what I consider to be important. That is, a team that starts from the beginning with this concern to create a solution properly structured, modularized, and with the ability to, at any moment, be able to add things. As I usually say, the iron triangle ("the triple constraint of projects") is not the scope, neither the time and nor the budget, this are only constraints, in my opinion the true iron triangle is the value and is the quality, plus the constraints. The value being the suitability of what I deliver, the quality being the ability that the solution has to grow indefinitely, easily, when I want to add other things, then comes the three constraints. This I would say is what I consider important: these three points of the iron triangle (scope, time and budget), and at the centre the quality and value, which I consider fundamental as the result of a product that you deliver in sequence of a project. That means, it has to be valued by the client and must have good quality, the rest, are the three constraints. The principle of using the so-called set by design, that is, leaving the maximum of options open, means that when you do this initial work, you have a very strong tendency to think that reducing the number of options to the maximum you are creating the right solution when it is exactly the opposite. In the beginning, the software architect must have a vision, but most of all, he/she should not start closing all the options because there is uncertainty. Due to this, the project may have a little more work in the beginning, but you leave some room in terms of design, and you can explore it. As you become aware, these options are narrowing. The problem is that many times, what we see at the beginning of the project is that there is an early attempt to reduce all options and flexibility. In that case, what is going to happen? You move forward, and you begin to see that you have thought that the solution that has to be constructed in reality begins to diverge, and at a certain point you will say, "Well, I have to approach again from the plan", but you are so far from there that you probably can’t do that anymore. As long as you keep a set of options open, as you move forward, you'll be able to figure out where you need to go. Therefore, it is key to have a team with quality, resources with quality, especially in this type of approaches. I believe that what a traditional Waterfall project has… and we already know, and it is not worth wasting time with it here… but, no doubt, I believe in to have not only multifunctional teams, but each team member itself to be multifunctional, and multifunctional people for me are people with technical skills far above the average, because people who have this mentality… "Well, I'm already good at something that I know, now I want to diverge a little from this, I want to expand a more my qualifications to be more flexible"… from what I have seen until today, only people with maturity and experienced in what they are really good are able to broaden these horizons. So, no doubt Agile teams have to have a really good core there. Obviously, there are also junior elements involved, and this is also part of Agile culture… that tolerance to errors, to receive people, and to bring them into the team. In short, the delivery of value, the delivery of quality, and to ensure that you have a qualified team to do the work… that would be more or less what I would say about this question.

Part II

It encompasses the issues that will explore the aspects related to the choice of an Agile methodology (out of the four under study) aligned to the needs and environment of software development projects. These questions were elaborated based on the review and analysis of Agile literature. For the following questions, it has been considered the following methodologies: Scrum; Extreme Programming; Lean Software Development; and Kanban.

Question 1/5

Luis Mattos**:** Regarding the changes in system requirements, how do you believe each methodology addresses this aspect? In what way does the positioning of each one on the use of development cycles (iterations) influence this aspect?

Tiago Palhoto**:** Regarding changes in system requirements, obviously that Scrum for being recognized for having this cadence, having this concern which is a bit of the instantiation of the basic principles of the Agile Manifesto, a concern with interactivity, collecting feedback, adapting etc, and which obviously is the core of these methodologies of everything that has to do with requirements, and changes in requirements, because it is precisely this. In my opinion, Scrum is the one that best enables these principles related to adaptability, inspections, adaptations to the process, with flexibility. Using these short cycles effectively lets you know if you are going in the right direction, and in the right direction you are precisely delivering the most valuable requirements. Extreme Programming, I would say it is not that related to changes in system requirements. I think it is more related to the optimization of the team's work as development, regardless of whether they are requirements changes or not. About Kanban, I would say the same thing. I see Kanban very connected to Lean Software Development. One of the ideas that I like to take out of Lean is the part of reducing waste, avoiding jumping tasks, etc., all mediated by the tool, by the Kanban board itself, so I see that as far as the requirements are concerned. I see these three a little more generalist. I see Scrum, it is not by chance that when you use Scrum you also use parts of Extreme Programming, for instance, you have Pair Programming when the application is coded together, but in a team that applies Scrum, so, I would say Scrum is one that effectively, as a base, gives the skeleton, gives the structure that allows to deal with changes in system requirements. As for the rest, I think it will be more related to the use of software engineering practices.

Question 2/5

Luis Mattos**:** Regarding the use of software engineering practices, how does each methodology address this aspect? Is there one that gives more emphasis on the use of software engineering practices (in detriment of project management practices)? If so, what are the reasons for this?

Tiago Palhoto**:** The Extreme Programming was born precisely in an attempt to optimize and improve software code and is not that closely related to project management and so on. Some of these software engineering practices started to be used in Extreme Programming, but with no doubt, one of the most effective withdrawals from Extreme Programming is clearly the Pair Programming. I've seen many teams do it, some badly, others well, but even those that did bad started badly and then finished well, but those that did well, in fact, it's possible to notice considerable difference since it leads to the ownership of the solution. The ability of two people to see what they are doing… the reduction of errors that can be introduced in the application in terms of logic, etc is much smaller and obviously makes possible to gain a great deal of time later. TDD itself, in short, plays a key role in the quality of code. The greater involvement of the developers in the testing part next to the business area since they also have to know a lot of the business, in fact, all this effectively makes Extreme Programming a methodology, in my opinion, very important with regard to software engineering practices, and that has clearly brought about a substantial improvement in that part. Regarding Lean and Kanban, I see these two very close. Lean I see a lot focus on the optimization of the process. That usually you can see through the sequence of steps that usually I always associate with a Kanban board, and so I always walk with them very connected. Obviously, these are techniques that in the background cover a bit of management, but also a lot of software engineering practices. This is because if I'm in a DevOps perspective and if I apply all my steps, continuous integration, continuous deployment, and continuous delivery, I can use a Kanban and I can use Lean to do my values streams in the background to identify all the passages that I have throughout my cycle, throughout my pipeline, and identify where I'm wasting effort and time. That is, I am optimizing the use of my software practices. I'm working out the form and identifying where I can optimize, that is, Kanban and Lean do not focus on using the practices themselves, but focus on exposing, for example, the whole process that uses the various practices. They show what I can optimize. That is, I also consider not only Lean but also other processes related to project management. So, in a direct way, I would say that Extreme Programming actually has a direct impact on management and software engineering practices. I would say that Lean and Kanban, for giving this visibility on a variety of levels, both in terms of software practices and even in project management, allow me to take this vision and realize where I can optimize my workflow and, eventually, what software engineering practices I can use. As for Scrum, I see it as a place holder. A framework which I can fit all this, so, in a generic way, Scrum turns out to be my place holder, Extreme Programming turns out to be what in fact gives me the techniques, as I mentioned, the important software engineering practices, and Lean and Kanban allows me to look at the form, either for the structure I have of Scrum, or for the way I've set up my engineering techniques in the various processes that allow me to perceive where I can optimize this, therefore, having a more indirect use in this sense. Thus, I think this would be more or less what I would answer in that aspect.

Question 3/5

***Luis Mattos*: In relation to the customer engagement in the project, the fourth principle of the Agile Manifesto stands out the need for greater interaction and collaboration between the project team and the business representatives. All Agile methodologies apply this principle at different levels. Based on your knowledge of these methodologies, how do you think these methodologies differ from this principle?

***Tiago Palhoto*: The Extreme Programming per se, from my experience, in spite of fomenting the use of user stores, etc., it is not the one that promotes this interaction with the client. I would say that Scrum is since it is a structural framework, that has a large set of artefacts and a set of ceremonies that precisely promote, in key moments, greater interaction with the client. Therefore, I have no doubt that Scrum, from a methodological perspective, from a structural organisational perspective of management, I speak not on a management part of expectations, but in terms of what we expect in terms of interaction with our client, does this work perfectly. Then I would say that Lean, the Kanban itself, can effectively when you are analysing certain processes, obviously promotes the need for customer involvement while you are identifying the value streams and you want to understand the process and how you can optimize it and so on. Of course, it also involves the client, but I would clearly say that Scrum is one that more clearly and visibly promotes the client's involvement in the project. However, one thing I do not see is the focus on managing expectations and communicating expectations, which basically has to do a bit with explaining the methodologies that are used. Just to conclude, what happens? I can tell my client that I'm going to apply Scrum and we're going to have these ceremonies etc. I can even say, "Look, you have to do this, this and this," but if I do not really explain to you what you have to do and how to do it, I'm not managing your expectations properly. Therefore, when it is said that "Business people and developers must work together daily throughout the project", this is completely true. Now, if they (the users) do not have the notion of how and why they do things, it's not going to work. And Scrum, as I mentioned formally, identifies every moment, but it does not tell me exactly who (should) or how to manage those expectations. I do not believe that the other methodologies that we identify are also focused on that aspect. So, I would say that Scrum is one that clearly makes it more obvious, and more formal, and more accessible this need that brings the customer close. It (Scrum) formalizes sprint reviews, which is the moment you show what you're doing, and this is also a way to bring the customer closer to you. Obviously it's the development techniques you use, eventually the user stores that are promoted, and I think that user stories are also promoted in Extreme Programming and it was one of the starting points of using it even more than Scrum since the Scrum Guide doesn’t even mention user stories, while in Extreme Programming, on the other hand, it is written all over. Users stories are also a simple way of communicating and talking to with the customer and are also a way of getting closer. Perhaps I would say that Scrum, in the first place, shows clearly the moments when this should occur, and eventually the Extreme Programming uses the user stories as a very easy way to talk to the client and at the same time maintain the scope of the project properly organised. I would say these two are the ones I see more impact on this aspect. Lean and Kanban, per se, I do not believe are methodologies focused on promoting customer engagement in projects.

Question 4/5

Luis Mattos**:** Regarding the approach to software deliveries (or parts of the software), the third Agile fundamental principle says, "Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale." In Agile, this principle is widely used but is applied in a specific way by each methodology. In relation to the following aspects identified, based on your knowledge of these four methodologies, how would you say each one approaches them?

- Continuous delivery – as a way of obtaining an effective cadence and determining a "standard" speed for deliveries;
- Deliver as fast as possible – the focus is on the speed of delivering value to the customer;
- Minimum viable product (MVP) – delivering a simplified but functional initial version of the software.

Tiago Palhoto**:** Starting with Scrum, obviously the continuous delivery, and if we're talking from a "non-releasable delivery" perspective, Scrum obviously promotes continuity by delivering and allows me to see my product grow, value to grow, and obviously it allows you to see this clearly and this is part of its very nature. The Lean does not, because it is more focused on optimizing the processes. The Extreme Programming, as the way it approaches the tests, and how it also promotes constant integration of code, which are fundamental factors for continuous delivery, I would say that these two are fundamental aspects. Scrum and Extreme programming approach continuous delivery in a more pragmatic way. Now, as for "Delivery as fast as possible", here I consider that both Lean and Kanban are two fundamental tools to help you implement it if you are effectively doing the right thing. As I mentioned earlier, Lean gives me a set of recommendations and practices that allow me to try to optimize, and ultimately to be more efficient in my delivery process, and, at the same time, using value streams tools that allow me to monitor my process and realize where I can optimize it precisely at a delivery perspective, as fast as possible. And I can say that delivering as fast as possible does not mean just delivering sooner. I can have the same delivery date but deliver more things. In my opinion, this is also fast delivery, and that's also the perspective I'm talking about. Scrum obviously allows me to control this, that is, if I reduce my iterations I will be, obviously, starting to deliver more often, faster, but obviously I might deliver in smaller blocks. Scrum also formalizes this. It is an excellent tool to formalize and to say that instead of every two weeks, that every week I will deliver. So I am saying, I will deliver as fast as possible, but it is only a formalization, Scrum per se does not control itself effectively if you're going to give it more or not, but it's there, it's telling you "let's work on this cadence and so be on your guard, but you don’t have to hand over anything." Now, I think that the techniques you have effectively in Extreme Programming can contribute to this, although I believe that both Lean and Kanban are the ones that can effectively help improve all of your processes, including in a process of DevOps that you have mounted. Precisely, this allows you to help deliver as fast as possible. All of this pipeline delivery I consider that Kanban, in fact, having a good visualization in particular of the limitation of your WIP, and in the background to avoid precisely the existence of bottlenecks, both Kanban and Lean, I think it has a fundamental role for help you in this matter because they clearly help you realize where you can improve, no doubt. MVP, the design of a simplified and functional initial version of the software, as for me, no doubt that Scrum is a perspective of exactly that. In fact, one of the comparisons that you have with Waterfall is that it only at the end of the product developed is that you'll deliver, while Scrum, with this interactive approach, clearly allows you to start early on delivering value and allows you to immediately "Well, at the end of these iterations I'm going to have this little bit here, now, this is an MVP, and I'll deliver." Therefore, Scrum certainly gives me, which formalizes this way of thinking. As for the rest, I see that they are already more practical tools that can help work on implementing an MVP. I think that here we are talking about a conceptual and procedural issue and say which of these mythologies is actually more focused on the construction, that is, the delivery and creation of MVPs and working with MVPs. I think Scrum, because it's a framework that covers all of this, goes in that direction. Of course, Kanban, if you have to work in support mode, for example, that is, that you have no interactions, you can also be saying "Look, we have here a lot of the next innovation, the next MVP will be the correction of all these elements we have here. No matter. We're not talking about sprints. The next MVP we want to launch with this all corrected." Then I will have a Kanban board that is advancing, where each one of them (the tasks) is coming, and I do it, I do it, always respecting my WIPs obviously. Obviously, in this case, Kanban itself also works this way, and this is one of the reasons for the difference between Scrum and Kanban. While Scrum is working incrementally, for periods to show progress, etc., Kanban is usually used in a continuous perspective and is not worried about interactions; it is always delivering value and hence, "delivery as fast as possible". Kanban itself also ends up promoting this, because the team is worried that there will be another, come another, come another, and in that sense, I would say, yes, in MVP, I would say that both Scrum and Kanban are two forms of organise the way we deliver to work. I think they would be the two most relevant that have an impact on this.

Question 5/5

Luis Mattos**:** Regarding the prioritisation of system requirements, the first Agile principle says, "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software." Agile methodologies work to prioritise the delivery of product functions that will bring greater value to the business (from the customer's point of view). However, each methodology deals with this prioritisation in different ways. Based on your knowledge of the four Agile methodologies under discussion, how do each of them work on the prioritisation of features and activities within a software development project?

Tiago Palhoto**:** As for Extreme Programming, I have no experience in using it in this area alone. I have always used Extreme Programming coupled with another framework, namely Scrum, that is, I have never used only Extreme Programming, so I cannot see Extreme Programming alone regarding managing system requirements. What I see clearly is Scrum, obviously by the way it handles and manages a backlog, which is already there native. Scrum itself talks about it, the role of the Product Owner makes this very clear and, therefore, I have no doubt that Scrum instantiates this principle very well. Kanban also works this way, that is, I have a great job here, but

I'm not going to work on iterations, but I also have someone who says, "Look, these things are coming and now what's next? And then it comes this and comes this…" You may have prioritisation or not, but actually the concern is that you are always pulling work and often with someone putting things in order and you are just pulling that job. By the way, it is not by coincidence that Scrum itself within each iteration uses a Kanban board; in short, they are closely linked. As far as Lean is concerned, it is more used in an optimization and waste reduction perspective, so, here what I can say is, the prioritisation of system requirements is one thing, and the way we deal with it and how we are constantly changing activities is another. Context switching is one of the great ways of wasting, and so here are some pointers. The priorities are great, but if you are constantly changing things while we are working on something, we will have problems since we will lose a bit of efficiency, that is, I would say that Lean can give you here some indications regarding not the prioritisation of system requirements, but eventually in the way you treat the requirements change, in a constant, etc. It may give you some indication of how prejudicial it can be. However, I certainly consider that Scrum, and Kanban itself, as long as there is obviously someone who prioritises and limits WIP's (i.e. ensuring that you are not exhausting your capacity at every stage of the process), I would say that are those which, at least from my experience, in terms of use are the ones that most satisfy the prioritisation of system requirements.

Part III

Requirements gathering of the information system that will be designed to implement the decision model derived from the analysis of the aspects discussed in Parts I and II. The objective is to collect the opinion of the interviewee about the ESSENTIAL features for this information system.

<u>Question 1/1</u>

Luis Mattos**:** An information system will be developed to implement the decision model resulting from the analysis of the aspects covered in Parts I and II of this interview. The main objective of this system is to point out if a given software development project is a candidate for an Agile approach and if so, to indicate which of the four Agile methodologies mentioned in Part II is the most appropriate one. In this context, what would be the functional requirements that you consider ESSENTIAL for this system?

***Tiago Palhoto*:** I would create a system like this based on an assessment, a questionnaire. Now, in terms of functionality obviously I would try to focus a little, as we say, on the fundamental aspects that I would divide, maybe, in the first place, helping to decide which methodologies to use, I personally would not use it in an exclusive approach, but in an approach where you can use one in combination with another, not exclusive, because reality shows it. So, that would be one of my first points. I would never limit it to A, B or C, but based on the result you could possibly say, you will use this, and you will use that as well. I would essentially see features with a focus on at least some aspects… one clearly in realizing the maturity of the organisational context where you are inserted. This itself already touches on some aspects that I would consider essential to address. Now, I would always aim at three or four sections, but one of them clearly related to the organisational context, namely at the level of people, at the level of competencies or leadership characteristics, but also at the level of the organisational structure itself, the level of the organisation's own strategy. That would be a point. The other point would be at the level of the actual reality of the company. The bottom line is, to find out where they are, what is the current situation, that is, if the projects are late or not, if there are losses or no losses, then if there are new teams and so on. These are all aspects. In the second part I would try to find out what the actual practical reality is in terms of teams, that is, what they use as tools and what they do not use if this question of thinking about change comes from teams, if that comes from leadership and so on. So, I would try to understand what they have tried, to perceive at the organisational level and at a high strategic level, to have this vision at the level of the teams, of all those who are involved in product development, but still from an environmental behavioural perspective. Then try to eventually understand their "pains" in terms of delivery, that is, what difficulties they have, because what they really want is to develop software and deliver, so it's important to have a set of fields here trying to capture what your problem is… Do you deliver late? Do you deliver what is really intended? Delivery is done within the time limit, but with many errors, in short, not so much focus on objective things as "to be using this or you are using that", "if you use these graphics or not" because that means that you are using something, but it doesn’t mean that you're using it well. So, in terms of functional requirements, my question will always be based on questionnaires, in surveys, I see no other way. But, in terms of guiding questions, especially in this part, in a more practical way of realizing what the real problems are because is it that will help to figure out which of these methodologies can help and not simply ask, "Well, do we want to work with time-boxed iterations?”. I may want to work with it, but that does not tell you why you want to do it. In fact, you can suggest it based on what you think they want, but in reality, it may be better to be sure why they think they need it in order to have a precise diagnosis. If I just ask, "Do you guys have an updated product backlog?" They can say “yes” without knowing what means “to be up to date”. Instead, I would say, "Look, does your product backlog reflect exactly what your client wants? Was your client who said exactly what this is?" Do you know what I mean? I’m not saying that these are the questions, but the approach has to be in a perspective of questioning in the most natural way… in the sense of questioning the problem and the necessity, and not so much the uses of the information radiators or the own resources that we offer to solve the things. That is, leaving the questions in a way to raise the problem and not properly say, "Look, you use that as a source of solutions to problems" and try to realize "If I do not use it, it's because I have this problem, if I do not use it, it's because I have that problem, or this is because I do not have that problem." This is not true because when we use it, we may be using it wrong. It’s understanding the need, focusing on the problem to be solved and not on the tools that exist to solve it. I often see assessments that are focused on evaluating if you are following the process, like a checklist, but it does not indicate whether the process is working or not. This may be valid in a first approach when you just want to know if you are following a process. In the case of this system, the questions should be more focused on clearly identifying whether this is done or not, and if I do, this means that I have a problem… not like "Look, you have this solution, you do not have it, and if you have that solution it is because you've solved that problem already." What this would mean in terms of functionalities, I see something seriously oriented to a questionnaire. This is clear. At the level of weighting, you may have a set of given answers that can be weighted that can soon indicate, regardless of the rest of answers, the most appropriate methodology is Scrum, for example. Or if the answers point out that, "We constantly have processes that are never optimized”, clearly suggests the use of Lean, even though the teams do not see that they are in waste, up here we have this notion, and therefore we will try to propagate this. So, here I can’t tell you whether it makes sense or not to weigh the answers. It's not an easy question, but this is what I can suggest you as requirements for this solution.

--- // ---

Appendix E – Focus Group Transcript - 1st Group

Focus Group data	
Moderator	Luis Fernando Bordeaux Mattos
Participants	Four middle management professionals from a humanitarian organisation (under non-disclosure agreement) based in Belgium. Low level of experience with Agile methodologies. The organisation has been working on the implementation of Agile practices in its internal projects.
Location	Brussels, Belgium.
Date and time	25th of January 2019.
Duration	54min
Comments	The language used in the activity was English.

Part I

It encompasses questions that will explore the aspects relating to choosing an Agile approach as a guide to the management and execution of a software development project. The objective is to collect the impressions and opinions of the participants on the pertinent questions formulated. These questions have been elaborated based on the review and analysis of Agile literature.

<u>Question 1/5</u>

Luis Mattos**:** Under what conditions would you suggest an organisation to opt for an Agile methodology?

Participant 1**:** The pre-requisite needs to be clever to implement an Agile methodology. I see your work. I see what you're going to say. Governance. I can read her mind. She wants to say governance. [laughs]

Participant 2**:** Clear accountability and readiness to really take decisions which so many times here is not so strong. I do not practice Agile, but I have the impression that Agile methodology requires even more stringent stakeholder engagement. Which we don't have in place even today because the project is not always based on stuff correctly, with the right people. All decisions are not really been discussed or taken.

Participant 3**:** I would say good knowledge of the methodology because it's very simple when you see that an Agile project charter with like ten pages or something like that, but when it comes to applying, it's very complex. So, I think the precondition is to have someone who knows about the best practices.

Participant 1**:** If you don´t know Agile, seams that Agile means fast, but Agile doesn´t mean fast. If you don´t know, the temptation of the people is to make this assumption.

Participant 2**:** And it’s true that the reining is such a charter I think they apply in Agile, but they don't apply the principles actually.

Participant 3**:** In the same direction, to have someone who uses the methodology, but also who is empowered to play this role as a grandmaster and so on and that people actually respect the rules. That is the problem that we face when we experience some of those methodologies that people don't like, they just drop the project. So, we need to have this part of the governance.

Participant 1**:** And also you have to know in which project you can use Agile and which project you cannot use Agile because it doesn't work. In a construction project, there's nothing Agile, in my opinion. I don't see anything Agile with construction.

Participant 2**:** Maybe part of the roof if they [the client] don’t tell you what they want.

Participant 1**:** Exactly! And it's not because the roof is less important or more important. You can’t do the typical things of Agile, like scoring every milestone or use MoSCoW. These things don't apply to any project. So, you need to know where you can apply this methodology. Typically, when you talk about this, you talk about IT projects, and there it fits almost everywhere.

Participant 3**:** That's where it [Agile] comes from, but now it is extended to all the topics.

***Participant 4*:** This is really about like saying there are the principles that are very very important and also the mindset of people like to be engaged with the big immensity, and because it's really easy to drop something like that, and as we spoke about it before, it made it easier these years and people just tend to abandon it. So, the levels of commitment, it's important to have in this case.

***Participant 1*:** There is also one point that is when you move non-Agile, people that are not used to Agile methodologies there's always this feeling of... a kind of fear of the blank paper. Normally that people are used to deliver something that works, and this is not the context of Agile. You deliver prototypes, and you refine, refine, refine prototypes. Now when you, for example, here when you deliver an IT product, we don't expect a prototype, even if they want to work with Agile. We expect the final product, every time. So, this is the typical mindset that you need to have before starting an Agile project.

***Participant 2*:** If you combine this with getting an approval for the budget, especially in this case, if someone is working on a prototype in order to demonstrate "Look, something to prove this is something we need, and we can benefit." Maybe this doesn't get budgeted to go that far. And also because there is a misunderstanding here. Let's allow this to happen. Let's evaluate then first; then we see if we should give more budget for it. We can assign a limited budget for this and agree to see if it works if we are not clear about requirements in the beginning.

***Participant 4*:** Yeah. There is one more thing I think about it. Is the evaluation process about this... OK, good, that's great! Let's say we are going to use the methodology, but we need someone that commits to evaluating. Okay. How is it going? And how we can use it to make the best of out of it because it's okay to implement, but then we need to know if it was really good. Was it efficient for our day-to-day basis? So, later, we might evaluate that.

***Participant 1*:** In my experience, you can’t evaluate a methodology transversely. You just have to evaluate the group, because normally the methodology, when you implement Agile, you adapt the methodology and the retrospective with the group. So, is there when you adjust, but you don't adjust the methodology in general for the organisation, you adjust it for your project, for your group, and you have to tailor every time Agile to fit the group, not the entire organisation.

Question 2/5

***Luis Mattos*:** Regarding the aspects of the ORGANISATION, more specifically within the MANAGEMENT level, what are the characteristics and behaviours that you think the leaders of the organisation should have to work with an Agile methodology?

***Participant 2*:** Leaders who are part of the Agile team or who are overseeing it?

***Luis Mattos*:** Who are overseeing it.

***Participant 2*:** Actually, I would say they have learned to listen. It turns out they probably don't know anything about Agile, and they need to hear who knows. You need to trust the team. The team is more than just the Scrum Master. Who is the one that is leading the team and making sure that providing quality assurance that everything is followed? According to the principals etc. So, the leaders need to trust them; they need to put their time in the project as well. Make them available for proving guidance and support. I think there’s a lot to prove in order to gain that confidence.

***Participant 3*:** I would say that depends on the maturity of the organisation on Agile methods. If it's at the start, they will have a key role to communicate on it... In change management... if it's already embedded into the role is quite different. It's more about governance, but here I would say that the dual would be this one, if we go for Agile, they need to be able to talk about it in an inaccurate manner. Knowing what to talk about.

Participant 1: Probably a leader should have a kind of helicopter view on the business that they want to improve, that they want to achieve with the Agile project and on the architecture that they want to put in place. They need to steer the project, and since an Agile project can diverge from the business very quickly, if you don't put a huge effort on having guidelines and vision, this might be a risk for any Agile project. So, the leader should be there with a helicopter view on vision methodology, be clear on that. Being this what we want to achieve, how do we achieve it?

Participant 2: If you ask me this question reverently an organisation like ours, which has so many projects, I´d say that would be normal to have such a mindset of starting to talk a common language on project methodology and governance. This needs to be driven and supported by the leaders, and not just being delegated to someone that doesn't care about it. This needs to be strongly supported and understood.

Participant 4: Exactly. This is what I missed here. This promoting characteristic, because it's exactly what happens most of the time. Someone just places the idea on the table and then ask someone else, usually us, to implement it. That's what happens, and then no one talks about it anymore. This is the promoting part that is necessary, just like refreshing and updating on the subject to keep the initiative alive. This is very important.

Participant 3: In the same idea, you need to know also the impact that it may have in terms of the follow up you need to build. In our case, projects are still quite based on the waterfall structure, but then we think we could start using Agile in our projects, but in Agile, the follow-up approach is not the same. So, how do you do this in this new framework? For the managing layer, it is better to think about it also.

Question 3/5

Luis Mattos: Regarding the aspects of the ORGANISATION, more specifically within the TEAM, what are the characteristics and behaviours that you think the project team should have / present to work with an Agile methodology?

Participant 3**:** Obviously, interest and understanding of what Agile is about. Why would they use this method instead of another one not having this imposed on them? I think it's very important if you work with this method to understand why you use it.

Participant 2**:** Maybe in terms of the team, there is a very high dependency. For the one next to me, everything needs to be Agile, for the other one next to me it doesn't look at this which I'm dependent on. There needs to be a mindset really focused on delivery, delivery oriented, let's say. Maybe Participant 1 can tell us the truth [laughs].

Participant 1**:** I don't have any truth. I'm just thinking of… Yeah... probably one of the things that the team at same point should... the first thing they should focus on is to be organised... they should be self-organised in a very good way because when they deliver, the product is no longer there. So, they need to deliver it to someone else, and if it's not well organised from the beginning, they can mess up clearly on documentation. This is one of the first things that disappear. The functionalities… and that's the tricky part, sometimes the client asks for functionalities that if you don't score them an appropriate way, they disappear from the product, from the tool, they shadow, and they become kind of like Easter eggs... and so the team should be very organised and should also be very coherently on the job that they're doing. So, when you do... often you do the shuffle within the team, so you need to be ready to not step into your previous work when you shuffle the position from the quality of the architecture committed to be developed. So, they need to stick with their current position and not to roll back unless it's acquired from the deal.

Participant 2**:** But you want to hear more than just characteristic, you want a team more dedicated, efficient, a team work-minded.

Participant 3**:** It's not compulsory to have a team dedicated, but it's better. It´s easier when you have a team dedicated to the project and not to something else. And in our organisation, in general, you work on a project in addition to your current job. There are not many project managers doing this 100 percent. I was thinking also it's again not compulsory, but of having people physically present in the same place. Here in the organisation, by definition, anywhere in the world doesn't mean that you can't work.

Participant 1**:** Another thing that fits with that in the context of building the team... the team should be fully independent of the outside. So, the team should be able to carry on the old project from the beginning to the end without going external to the team, because one of the problems that may cause is if you have to replace someone in the team in the middle of a project. That's going to be very problematic for the health of the project, so... Or if you have to act in person because you're alive at least some skills that you forecasted in the beginning, so the team should be the same from the beginning to pay on the project.

Participant 4**:** It's really related to what was already spoken, but I think it's great for the instant engagements. The self-organisation, because as you're saying, the people work in different workstreams. So, the sense of this teamwork is very important and also to make sure that the person is really engaged in the methodology actually because this is one of the challenges that we are facing now. Because it's something, well, we are doing that, we have all those to do’s and deliverables and so on, but then, well, it's not important actually because I have other priorities. So, it´s more like this, to get the results like it takes more time, at least here.

Participant 1**:** But this is not the team responsibility; this is the leader’s responsibility... to set the priorities. And normally in the ideal world, the team is fully committed to a product and doesn't shuffle from product A to product B to product C.

Participant 4**:** This would be the idea. Yes, I understand. Of course. I believe that what would be the challenge here... OK, yeah, the reality is that we shuffle from one product to another because this is really what happens. Then, how to settle this? The idea, in this case, is that we customize things to get there, isn't it?

Question 4/5

***Luis Mattos*:** Regarding the aspects of the PROJECT, more specifically in the ENVIRONMENT, what are the environmental factors for the project that you consider to be more characteristic of initiatives eligible to use an Agile methodology?
***Participant 2*:** In Agile, I thought it would be good for the Agile team to seat somewhere closer together.

***Participant 1*:** I had to say that if you bring people into that share some kind of similar culture would be probably easier to succeed in a project. So, you want to have this shared knowledge and values among people that work in a multicultural environment because that would be easier for the team to work together, to build trust. Because if you have to step into the first line between European, Asian, North American… So, if you have people that are there for the first time and if you mix up people... I don't know… you might have some misunderstanding. When you face another culture, you need to know what you can say and what you cannot say.

***Participant 4*:** For me, it´s not great saying that because there is ah...

***Participant 1*:** No, no, for me it´s not that what is necessary, but it's necessary that everyone is conscious that work, especially if you are working in a multicultural project when you get up a team that is very close, people need to be aware of the cultural differences between each other in order to avoid misunderstandings.

***Participant 3*:** I think it must be a quite flexible environment in a sense to be quite horizontal, with people used to work not having a very big hierarchy. Used to, like our organisation, but this is quite adapted because you get to work with people, you know, I think this... not having a boss. It´s a horizontal system. This is the mindset that you must have. Also being used to work with people from different backgrounds. That being strict with your department because in this kind of project you would get to work with IT people and HR people depending on the project. It is quite multi-functional. I don't think that everybody is at ease in this kind of subject.

Participant 2**:** This would be possible in a very solid organisation.

Participant 4**:** Yeah. That's true. In a space where you can do the self-management as well. First, you need to have the freedom and your nature be empowered to go through transversal subjects and different departments, but through all that... and I understand when you say about the culture, it´s because you need to see some similarities. And then it goes more fluid.

Participant 1**:** Not necessarily to see the similarities, but to understand the differences.

Participant 2**:** Yeah, but this also applies to our job here since we work in an international organisation. This is in all the work we do. I see that as natural, this is kind of expected, but in an Agile project, because you are that much closer, probably in a day-to-day work, those differences just come up more often.

Participant 4**:** People need to feel safe. In the sense of, okay, if there is any blockage, then you need to feel safe to express yourself and break this blockage. Because when I see a lot of hidden bottlenecks in terms of interpersonal relationships, where it's really on a personal level, it's on an emotional level. And then it blocks the project; it blocks the process; it takes time to reorient things. And I see a lot of people that don't feel safe here. Yes. So, I see a lot of different sort of change management that I've seen here with people saying about people advisor and so on. So, I think the safety even though it's like, okay, that sounds quite obvious, but for me it's not.

Participant 1**:** The same points to me, because especially here, you know when you're talking to people that you are pretty sure that something that you say might be used against you sooner or later. And this, when you have a closed project team like you have when working Agile, should never happen.

Participant 4**:** So, having this mindset, yes, for us, is really essential.

Participant 2**:** I also think about the professionalism that you mentioned. There is a lot of emotional things that shouldn't be here, and these emotional things are simply driven. In some cases, this affects people trying to hide something that you know about it, but they wouldn't say that they're going to behave in a way that they rather block than just saying, "Okay, you know I don't know, onboard me." We could be more beyond than just accept that we all do. We can all learn from each other from different areas having this openness to say what o need to say and to really help each other.

Question 5/5

Luis Mattos**:** Regarding the aspect of the PROJECT, more specifically in the PRODUCT scope, what are the characteristics related to the product of the project you think are most relevant in determining an Agile approach as a development methodology?

Participant 3**:** If you go to the definition of the Agile method, one thing about the expectations regarding what must be changed... the basic definition they give you applies where you are changing expectations, especially in IT, where it's evolving very fast. That's why it was ever there. That would be the first thing. If it's a very stable thing, very stable product, no changing environment, then an Agile method is not very interesting. There's this idea of adapting to changes.

Participant 1**:** When I hear the product, for me, it certainly means that probably the project phases are going down, and we are going into the operational mode. So, to me, it's like, it's mainly over the Agile part of the work because the project is going, so because before you talk about prototype, prototype, prototypes a product, and then you deliver a product, and you deliver not only to the unit, you deliver to the operations and the support line, and through all the follow up as opposed. So, probably the project should be well documented, be well pleased, and be well on hand over to the support to have a sustainable product afterward. Because you don´t expect after one month that support come back to you for some clarification on everything. Maybe this will happen, but will happen in a period of six months when you collect all these little things and you call for a version two.

Participant 3**:** You must be able to find iterations also to decompose and have deliverables which are usable, which is not actually always very easy.

I've had quite a few projects that promote the Agile methodology and then they were working so hard with iterations. So, how is that? Maybe they should not use Agile.

Participant 2**:** I think it's also the expectation. Any product you build in an Agile manner hardly brings out something completely different that was expected. After all these iterations, because you always test and demo stuff during the iterations, it’s easier to spot things. “No, that's not what I wanted at all.”

Participant 1**:** One of the other characteristics that should have is that the product, in the end, is the 80 percent threshold. So, 80 percent of the milestones or stories, you can do it with the 20 percent of the effort. The less 20 percent of the use cases that you have to build, represent the 80 percent of the effort because you need to explore there, there are very nice clever things. So, you have really to find your basic 80 percent of your milestones delivered and then you might have a sustainable product.

Participant 4**:** It's hard to say that because the area we apply these practices, the deliverables are quite either like super small chunks deliverables, like too detailed small ones or it took like one month, and it changed everything. And it´s quite hard to follow. So, for me, it's hard to answer this question, but in theory I know the answer, Okay, that should be like... you need to have complete results, but then when I apply into one work, what we are doing is quite difficult to say because we are on the opposite side. We are just customizing to place a cue which we can deliver because today we are not delivering something. So, in the end, Agile was helping us just to see if there are some results and some evolution.

Participant 2**:** And it was coming back to the team in order this to work like you just described. And maybe in the team that needs to be a few experts and the rest, you know, different levels of expertise, but there needs to be at least one expert in order to make sure things are evolving well, because sometimes... I'm saying this because here in the organisation sometimes people are just put into positions they don't have any expertise. And this is dangerous. And then the expectations are high, and it fails, and then the methodology is blamed.

Participant 1**:** Another thing that I was thinking about the product is about expectation management. When you start a project, you start with not only a nightmare expectation of a project... Who do these and that, etc. If you're doing an Agile project, you might end up with a score that doesn't affect the expectation of the management. So, this should there be some kind of alignment with the change management or the communication of the project. Should there be alignment on what is, in reality, you aren't going to deliver, because you end up with a product that doesn't make sense if you make a huge commercial for the most important functions, and then you don´t deliver it because it's coming at the end of your stories.

Participant 2**:** Or in this manner with a different perspective, or leaderships have a great idea, they have a vision of what they want and what will happen, and then if the team works in an Agile manner and it takes into account what it is. So, while the solution is any useful or not, at the end of this Agile project, there is an output that doesn't fit to what the leadership had envisaged. But maybe this mindset needs to be given all the leadership because leadership needs to be all open to change its vision or reluctance to trust the team.

Part II

Requirements gathering of the information system that will be designed to implement the decision model derived from the analysis of the aspects discussed in Part I. The objective is to collect the opinion of the participants about the ESSENTIAL features for this information system.

Question 1/1

Luis Mattos**:** An information system will be developed, aiming to implement the decision-making model resulted from the analysis of the aspects discussed in the first part of this focus group. The main objective of the system is to indicate if the project is eligible for an Agile approach or not. In this context, what would be, in your opinion, the ESSENTIAL functionalities of the system?

***Participant 3*:** I think quite simply on this. This is an objective that I have set for myself this year. In which cases, we go for an Agile method. I see it as a checklist, actually. I don’t see anything more complicated than this. I think of this as a decision tree.

***Participant 1*:** Probably, you will need to have some kind of hot points or group of answers that you can score. I cannot discuss submitting to see all the questions, but you probably can make a group of questions related to the IT feasibility, another group of questions related to their business-related, and you give to each question a kind of score and how much of it is the percentage of Agile.

***Participant 2*:** I would start with the requirements. What are the requirements? Are they well known and understood? And this would score from 0 to 2, for instance. Based on this, it can be maybe a criterion, of course, in the right direction.

***Participant 1*:** And probably to make it effective you need to have a different type of users with different, let's say, angle to see the idea in order to have this scoring more effective. So, you need an IT user with the view from the IT angle, a business user with the view from the business angle, or you can split it into... actually you can put heavy users...

***Participant 2*:** Maybe you can attribute categories to resources.

***Participant 4*:** Different people and budgets.

***Participant 1*:** Each resource is like when you do the scoring with Scrum. Each person should say what they think even if they don’t think that they really understand it completely, but they might have some knowledge, or they might have some personal experience. Well, it would simplify it a lot.

***Participant 3*:** But it depends on how broad you want to see it because it can be just at the project level. Granted the characteristics of a project, but it could also have transversal questions of the conditions there in the organisation. Could also be this.

Participant 1**:** Another thing you should match is all this input that you catch from different people. It should be tailored to the drivers of the relationship. So, you should find some key transversal drivers for the organisation, more departmental drivers, or both. And you should match all this with each other. How to do that I don’t know... with a weighted or some kind of master system to score, in order to have a more contextual result, because we share all the users you have another view of all the people in the organisation, but you view the noise they match their drivers. So, they need to match their drivers.

Participant 3**:** Maybe something that would be interesting would be not pointing to a unique way. Those questions, those characteristics in the end, like you said, this is a project that is 60 percent Agile, well, it's only 60 percent. So, watch out for this, this and this. Well, if it's a hundred percent Agile, then just go for it. You don't have many questions to ask yourself but in some cases...

Participant 2**:** This is after assessing the whole Agile point of view. I think you need to know what is eligible to be Agile.

Participant 3**:** Yes, but that would be the next step. You can go for Agile, but here are some recommendations, given the characteristics of the job.

Participant 1**:** Another thing related to the business the project can spit out; you can find outputs of previous projects. So, if a previous project has the same 60 percent of the other project, you can give you already, because this might be interesting to the final point of the project here with all the lessons learned, so you have a similar project, and after a year that starting, you have got all the lessons learned of a similar project.

Participant 3**:** And link it to the documents of previous projects.

Participant 1**:** Maybe it might not be in the same context, but it might end up being with the same challenges, and with the same lesson learned, and you get exactly the appropriate lesson learned from the project which resembles the one that you are trying to start.

Participant 3**:** Especially for the ones "60 percent Agile".

Participant 1**:** Yeah. But also if it's like 99 percent Agile, you can get another 99 percent of the same Agile project with the same lessons learned, so you can already have some feedback even before starting the project. Something relevant because you can have lessons learned now, but where to take it, where they are, how you collect it, are they relevant to my project or not? Which is the most important question?

Participant 4**:** And just to make a connection with what you're saying, you're also saying about not internal projects, but all the industry projects. So, it would be like a sort of comparison like, okay, what is the average in the industry for this project and then how does it stand from my project here? So, it's a sort of database as well.

Participant 1**:** Yeah. I'm not sure if you can get enough information from an industry project. So, it depends on the type of question that you want to weigh, and you want to score it. I'm sure if we can get some insight of an industry project, but maybe you can also judge a product from an external point of view as a tip, as a user, a project that using the Bible. Maybe is use in your organisation and you have some knowledge, and you can score it again. Why not?

--- // ---

Appendix F – Focus Group Transcript - 2nd Group [Portuguese]

Informações da atividade	
Moderador	Luis Fernando Bordeaux Mattos
Participantes	Seis gestores de projeto de TI de uma organização portuguesa do sector da Banca (sob acordo de não divulgação). Nível intermediário de experiência com metodologias Agile (média de dois anos e meio). Atualmente a organização utiliza amplamente Agile em seus projetos de TI.
Local	Lisboa, Portugal.
Data e hora	29 de janeiro de 2019.
Duração	49min
Comentários	A língua utilizada na atividade foi o Português (Portugal / Brasil).

Parte I

Engloba as questões que irão explorar os aspetos relativos à escolha de uma abordagem Agile como guia para as ações de gestão e execução de um projeto de desenvolvimento de software. O objetivo é coletar as impressões e opiniões dos participantes às questões pertinentes formuladas. Tais questões foram elaboradas com base na revisão e análise de literatura sobre Agile.

Questão 1/5

***Luis Mattos*:** Em que condições sugeriria a uma organização a opção por uma metodologia Agile?

***Participante 1*:** Acho que podemos usar o Agile um bocadinho na montagem do que se espera do projeto. A medida que vamos avançando, vamos detalhando o nosso âmbito, os nossos requisitos, e a própria metodologia em si vai se adaptando à medida que fazemos entregas. Portanto, eu acho que se tivermos um âmbito bem detalhado, podemos optar por uma metodologia mais tradicional como o *Waterfall*, mas havendo essa dúvida, eu diria que o Agile seria uma abordagem mais apropriada.

Participante 2: Eu posso complementar, ou melhor, concordar com o que o Participante 1 diz, mas ao invés de falar de âmbito eu falaria de requisitos. Se os meus requisitos são bem conhecidos, se eu sei exatamente quais são os requisitos do meu projeto, eu acho que não há porque utilizar Agile, a não ser que eu esteja em uma *software house* em que entras em um portfólio de atividades a serem realizadas. Se nós estamos em uma abordagem onde eu conheço bem quais são os meus requisitos, onde eles são muito bem definidos, como por exemplo num projeto de *Legal*, eu sei exatamente o que é aquilo, então para que que eu vou para o Agile? Faço a identificação dos meus requisitos, analiso quais são os impactos que eu tenho na minha solução e planeio a implementação da mesma. Acho que a noção dos requisitos, o grau de certeza dos requisitos, são as condições que me permitem optar por uma metodologia ou por outra.

Participante 3: Eu acho que os fatores principais seriam verificares se a equipa, ou melhor, se a empresa é projetizada ou não, se os requisitos para o projeto estão bem definidos ou não, e como funciona a gestão de configuração na organização. São fatores que eu acho que seriam básicos para decidires por uma metodologia Agile, partir para uma abordagem híbrida, ou mesmo manter-se na linha de um *Waterfall*... algo mais tradicional.

Participante 4: Eu queria concordar e complementar. Penso que a questão dos requisitos seria provavelmente a característica mais importante entre escolher o Agile ou outra metodologia não Agile. Mas também a estrutura da própria organização. O Participante 3 até falou se é projetizada ou se não é. Se há a possibilidade das equipas trabalharem em conjunto ou se há muitas barreiras burocráticas, e também a questão de tempo. Penso que conceitualmente o Agile também surgiu da necessidade de algumas empresas contratarem uma bolsa de horas em vez de um projeto, e ao contratarem uma bolsa de horas conseguirem um compromisso com uma funcionalidade, por exemplo. E a partir daí também há as empresas que trabalham com iterações, a definir esses limites temporais e construir um entregável a partir daí.

Participante 1**:** Eu gostaria de adicionar mais um ponto que tem a ver com o conhecimento dos vários recursos. Acho que o Agile leva a uma proximidade que facilita a mitigação de *gaps*, sejam funcionais, sejam no desenvolvimento, ou até mesmo do negócio. Acho que havendo um bom conhecimento repartido entre todas as áreas envolvidas no projeto, acho que o Agile também se justifica. Não havendo tanto conhecimento à partida e sendo necessário maior interação, acho que o Agile pode ser útil na medida em que ajuda a elucidar e complementar os requisitos.

Questão 2/5

Luis Mattos**:** Em relação aos aspetos da ORGANIZAÇÃO, mais especificamente no âmbito da LIDERANÇA, quais são as características e comportamentos que julga que os líderes da organização devem ter / apresentar a trabalhar com alguma metodologia Agile?

Participante 3**:** Primeiramente eu acho que o líder dever saber delegar e ter conhecimento da equipa que está a gerir, porque se você conhece a sua equipa, você sabe até onde ela pode chegar, e você confia, e tendo confiança, você acaba tendo mais tranquilidade para delegar. Além disso, quando você delega, fornece à equipa todo o ambiente para a, digamos, responsabilidade. Então a delegação tem a ver com a responsabilidade da própria equipa.

Participante 4**:** Eu acho que é importante também saber enquadrar-se na equipa e não estar a ver tudo de cima, digamos assim. Muitas vezes é necessário estar em muito contato com a equipa para poder estar por dentro dos temas, ou seja, quase que ao mesmo nível. Isso é importante para que possas tratar ou agilizar alguma coisa e não estar, digamos assim, em um patamar mais acima onde podes estar a ver os temas muito de longe. É importante essa possibilidade de estar próximo da equipa e conseguir fazer essa ligação. Criar uma relação com a equipa para poderes desbloquear as coisas o mais rapidamente possível.

Participante 1**:** Você precisa ser versátil ao nível da tomada de decisão. Nós podemos ter uma iteração bem definida, mas nada me garante que as iterações à frente vão seguir na mesma linha. Poderemos ter que nos adaptar, seja por uma mudança no mercado, seja por uma mudança interna. E aí a decisão tem que partir de quem de direito, assim como a orientação também. Portanto, tens que ser versátil para ser capaz de justificar o porquê das mudanças e das transições, assumindo as responsabilidades e sendo também assertivo naquilo que propões.

Participante 6**:** Partindo do princípio de que estás comprometido com os objetivos da organização em relação a metodologia Agile, é importante também conseguir medir os benefícios que a metodologia está a trazer ao longo do tempo e pensar como é que vais conseguir disseminar o processo entre as várias equipas Agile. Penso que esse seja o principal *driver* para o processo de mudança e garantir que estão todos a trabalhar no mesmo sentido.

Participante 2**:** Eu acho que a equipa tem muito a ver com a questão do valor que é a entrega para a organização. Enquanto numa metodologia como *Waterfall* eu recebo todo o valor através dos requisitos que foram definidos de uma única vez, enquanto que eu aqui [no Agile] irei recebê-los notadamente ao longo do tempo, ou seja, ao longo do tempo eu vou percebendo o valor, e acho que por isso a exigência tem que ser bastante grande no início. Por isso é importante calibrar a equipa, e principalmente a pessoa que define a prioridade das entregas. Calibrar bem o que é prioritário, o que é valor para a organização ou não. Eu penso que será a peça chave. No *Waterfall* eu posso estar sentado a esperar que o valor seja entregue... o produto final, mas aqui [no Agile] não, aqui, como sabes perfeitamente, o projeto pode ser um pouco mais longo, ou não, mas que vai haver entregas periódicas, e as primeiras entregas deverão ter mais valor do que as posteriores, ou seja, deverão ter mais urgência para a organização do que as seguintes, e para isso vais precisar ter um maior nível de exigência e mais proximidade. Eu acho que já foi dito isso aqui antes, que mais presença, mais proximidade é urgente.

Participante 4**:** Respondendo mais objetivamente sobre quais características, acho que passa pela confiança, compromisso, responsabilização e, também, reconhecimento das equipas. Isso tudo, assim como maior transparência dentro da equipa e do próprio líder.

Questão 3/5

Luis Mattos: Em relação aos aspetos da ORGANIZAÇÃO, mais especificamente no âmbito da EQUIPA, quais são as características e comportamentos que julga que a equipa do projeto deve ter / apresentar para trabalhar com uma metodologia Agile?

Participante 4: Eu acima de tudo diria diretamente responsabilidade, pelo fato da equipa garantir que as coisas sejam feitas e que conseguem cumprir com os planeamentos que se propõe. O espírito de equipa, porque uma equipa tem que estar sempre coesa e muitas vezes a falta desse espírito leva com que os entregáveis fiquem incompletos devido a competições dentro da equipa. O conhecimento de um encaixa na necessidade do outro e vice-versa. Portanto, ser responsável e ter uma boa "política" de camaradagem são duas coisas essenciais.

Participante 1: E ter humildade. Humildade para reconhecer que eu sozinho domino o meu domínio, mas que sim, sou humilde o suficiente para dizer quando não domino algum tema, pedir ajuda e receber *input* da equipa do projeto.

Participante 2: Eu acho que a experiência também é um aspeto importante. Está um pouco relacionado com a humildade. E o comportamento é um pouco mesmo ter espírito de equipa. Há também a parte que sou eu que vou lá buscar o meu trabalho e não esperar que o tragam para mim. Sou eu que vou lá buscar, sou eu que crio o meu ritmo, mas o meu ritmo é controlado pela equipa, porque a equipa também observa qual é o ritmo que eu estou a colocar. Então se eu tiver com um mal ritmo a equipa observa e eu vou me sentir envergonhado, se eu tiver um bom ritmo eu vou ficar entusiasmado. E acho que esse é um dos aspetos que também acelera um pouco as entregas se a equipa tiver um bom comportamento.

Participante 3: Eu acho que de uma forma geral é o sentido de unidade. Porque são complementares, pois a tarefa não é de um, a tarefa é da equipa. Então a ideia é que você saia muito mais do seu mundo. Você não apenas dizer, olha, eu fiz a minha parte e pronto, mas sim buscar que minha parte tenha encaixe na parte do outro para que o trabalho da equipa seja entregue.

***Participante 6*:** Há uma ideia multidisciplinar também do que se faz, portanto, vestir a camisola não só, apesar de que eu acho isso muito justo, mas também garantir que haja um *tester* a ajudar o *developer*, assim como um *business analyst* para ajudá-lo na sua função, uma vez que o trabalho dele [developer] não acaba na programação da funcionalidade que tem que entregar, mas só quando o ciclo se fecha é que a equipe ganha o valor com a entrega do que foi comprometido.

Questão 4/5

***Luis Mattos*:** Em relação aos aspetos do PROJETO, mais especificamente no âmbito do AMBIENTE, quais são os fatores ambientais para o projeto que julga que são mais característicos das iniciativas candidatas a utilização de uma metodologia Agile?

***Participante 1*:** Eu vou falar mais do *mindset* da equipa e não tanto do ambiente em si. E quando digo *mindset*, faço uma comparação com as pessoas que estão há muito tempo no mercado de trabalho, em organizações mais enraizadas, digamos assim. Há mercados de trabalho que já vem formatados com diversas metodologias, onde as pessoas que lá estão já estão há muitos anos na casa e comparo-as com as que... não digo quem tenha menos experiência de trabalho, mas mais jovens, com uma mente mais aberta no que toca a adoção de novas metodologias. O problema é que nas organizações que estão muito enraizadas há um bloqueio a novas ideias e novas formas de trabalho. Portanto, acho que sim, o ambiente também tem que promover essa abertura, tem que permitir às pessoas inovarem com as ideias que trazem, nas metodologias que querem aplicar... e por isso eu diria que nos ambientes mais jovens, nas empresas criadas mais recentemente, o ambiente é mais propício para inovações ao invés de em empresas que já estão aí enraizadas no mercado.

***Participante 4*:** Continuando aqui com o tema da reação à mudança e da inovação, eu creio que a criação de *open workspaces*, como se vê agora, foi uma resposta a isto. Eu estou lá no meu gabinete e ao me deparar com um problema envio um e-mail e fico a esperar por uma resposta. Hoje em dia as pessoas querem uma resposta imediatamente e os *open workspaces* proporcionam essa maior proximidade entre as pessoas para falar e resolver os problemas de imediato.

***Participante 3*:** Eu acho que tem a ver mais com a questão da cultura da empresa, da direção, da organização como um todo, de querer mostrar bem qual é o papel daquela equipa, o que ela entrega e como se encaixam as peças do quebra-cabeças. Até porque normalmente hoje em dia no mercado vês muitas empresas que trabalham no modelo de *home-office* devido a limitação cada vez maior do ambiente físico causada por diversos fatores. E quando escutas falarem que o Agile só funciona bem no físico, se isso for verdade, há uma tendência de no futuro o Agile desaparecer, porque é uma tendência do mercado a descentralização das operações das empresas, principalmente nas provedoras de serviços. Há muitas empresas no mercado que hoje já atuam quase que 100 porcento em modo *home-office*. Empresas grandes. Eu acho que empresas como essas podem trabalhar com Agile desde que elas fomentem nas equipas determinar claramente qual o papel esperado de cada um dentro daquele determinado projeto e contexto organizacional.

***Participante 4*:** Eu queria só complementar o ponto que o Participante 4 falou sobre os *open workspaces*. Para mim, não há dúvida que eu concordo com o uso deles, mas eu iria um pouquinho além, porque há *open workspaces* que são muito cheios, então é importante que os grupos de trabalho estejam próximos. Não adianta que os grupos estejam no mesmo *workspace*, mas um grupo esteja muito afastado do outro, pois você perde a sinergia proporcionada por esse ambiente.

***Participante 2*:** Eu concordo com a questão do *open workspace* porque a parte da confiança se transmite muito por aí com a questão presencial, mas também é possível à distância. Acho que é possível, mas a empresa tem que estar bem consciente daquilo que está a fazer. Outro ponto que considero que tem a ver com essa variável do ambiente é o facto de eu ter produtos propriamente ditos. Por exemplo, a Microsoft que tem o Word, o PowerPoint consegue perfeitamente separar a gestão de ambos. Se estamos a falar de uma empresa em que é difícil uma gestão por produto, acho que é um ambiente onde o Agile poderá não funcionar da melhor forma. Ou então, se for algo que não seja viável obter uma abordagem por produtos, deve ser algo onde o desenvolvimento do produto possa ocorrer de forma incremental. Em que não se perde nada em avançar um pouco mais com algo em vez de entregares outra coisa qualquer. Eu acho que é mais por aí.

***Participante 4*:** Acho que um fator ambiental é a cultura do próprio país. Tanto Portugal quanto o Brasil são tradicionalmente países avessos ao conflito. No norte da Europa as pessoas estão mais acostumadas a dizer diretamente quando algo está errado, e nós culturalmente levamos isso para o lado pessoal, mas não significa que nós estamos errados, significa sim que o trabalho que nós produzimos não foi o que era o esperado. Isso ao sul da Europa nota-se muito e é um fator que muitas vezes nos impede de avançar mais rápido na solução de alguns problemas. Porque temos o receio de ofender alguém ou passar por cima de alguém quando já temos as respostas todas na mesa, mas falta-nos o aval de alguém que está ausente e que só volta ao passar uma semana, e nós esperamos uma semana para a resolução desse problema.

Questão 5/5

***Luis Mattos*:** Em relação aos aspetos do PROJETO, mais especificamente no âmbito do PRODUTO, quais são as características relativas ao produto do projeto que julga que são mais relevantes na determinação de uma abordagem Agile como metodologia de desenvolvimento?

***Participante 4*:** Eu diria para começar se é um produto fechado ou se é um produto que se planeia e que sofre alterações ao longo do tempo. Acho que se for um produto fechado, podes conseguir aplicar uma metodologia *Waterfall* conforme o Participante 2 já referiu, pois os requisitos estão fechados. Mas se for um software onde você vai lançar uma primeira versão, seguida de uma segunda versão e etc. há uma condição qua favorece mais o uso de uma metodologia Agile.

Participante 1**:** Eu acho que não depende do produto, sinceramente. Acho que independentemente do produto se consegue aplicar tanto uma metodologia quanto a outra. Depende mais do ambiente da organização, depende mais do que conhecemos ou não conhecemos do produto, e são essas as características que vão nos dizer se podemos aplicar ou não uma metodologia Agile.

Participante 6**:** Aqui também, se tu estás a desenvolver um produto e estás com uma gana doida para investir em uma metodologia, se o próprio produto permitir você crescê-lo incremental e iterativo, sem dúvida que esta [Agile] é a melhor abordagem, pois lá está que no *Waterfall* normalmente estás a querer ter o produto já todo acabado até chegar a pôr ele no mercado, então uma atenção que tens que ter é o *time-to-market*. A importância de lançares algo com o mínimo *set* de *features* importantes, e nesse caso a abordagem Agile é mais adequada. Então eu diria que as características do produto vão também nesse sentido, nomeadamente o mercado e se o produto em si pode ser realmente trabalhado de uma maneira incremental.

Participante 1**:** Concordo contigo. Se é um produto que nós conseguimos repartir em entregáveis, o Agile sim se aplica.

Participante 3**:** Eu acho que a questão do produto está muito mais associada à necessidade do cliente. Se para o cliente faz sentido receber algo antes, facilita o uso de uma abordagem Agile. No *Waterfall* você poderia sim fazer um planeamento de forma a fazer entregas faseadas, só que você ficaria, digamos assim, mais engessado, preso aos *milestones* estabelecidos lá no início. O Agile é mais livre nesse sentido, pois eu não preciso dizer exatamente o que eu vou entregar no início, eu vou entregando em partes e essas partes podem ser móveis de acordo com a necessidade do cliente ou do mercado. Já no *Waterfall*, embora eu possa quebrar as entregas, eu tenho que ter o âmbito fechado na partida e dizer qual a parte do produto que eu vou entregar em cada *release*. Se em algum momento o cliente mudar a necessidade, o controle de mudanças no *Waterfall* é muito mais complexo do que fazer isso num ambiente Agile.

Participante 4: Eu não concordo com parte do que foi dito na medida em que se o produto for divisível em incrementos preferiríamos utilizar o Agile. Acho que as duas principais características são: o conhecimento do requisitos e o tempo para fazer, porque dividir um produto em incrementos pode até ficar mais caro. Dependendo da fase em que estamos, conseguimos desmobilizar as equipas, e se já tivermos o conhecimento dos requisitos à partida, eu diria que o *Waterfall* provavelmente irá ficar mais barato. Acho que isso se prende muito com o conhecimento e com o grau de incerteza, como o Participante 1 falou, sobre a questão do valor das entregas.

Parte II

Levantamento dos requisitos funcionais do sistema de informação que será projetado a fim de implementar o modelo de decisão derivado da análise dos aspetos discutidos na Partes I. O objetivo é coletar a opinião dos participantes sobre quais seriam as funcionalidades ESSENCIAIS para este sistema de informação.

<u>Questão 1/1</u>

Luis Mattos: Um sistema de informação será desenvolvido de forma a implementar o modelo de decisão resultante da análise dos aspetos discutidos na parte I. O objetivo principal do sistema é indicar se um dado projeto de desenvolvimento de software é candidato a uma abordagem Agile ou não. Neste contexto, quais seriam os requisitos funcionais que julgam ESSENCIAIS para este sistema?

Participante 3: Acho que um dos requisitos seria o conhecimento que determinada equipa tem sobre aquele produto, aquele sistema, sobre o software a ser desenvolvido. Quanto mais conhecimento eu tenho do âmbito e da tecnologia, é mais fácil, entre aspas porque isso não é matemática, aplicar um *Waterfall*. Se eu tiver um ambiente onde eu não consigo ver muito longe, onde eu tenha uma visão mais limitada, o Agile já faz muito mais sentido porque eu faço, verifico, testo e tenho como, digamos assim, "consertar" o caminho do projeto. Eu acho que essa questão...

***Luis Mattos*:** Me deixa tentar esclarecer melhor essa questão. O contexto aqui seria o seguinte, a primeira parte desse estudo envolve toda uma parte de revisão da literatura, mais a análise dessas informações que nós estamos aqui discutindo, mais os dados da entrevista com especialista em Agile que eu já realizei, e a ideia é: essas informações vão ser todas analisadas e vão gerar um modelo de decisão indicando quais são as características e/ou aspetos que eu tenho que analisar, que vão variar entre as características da organização, as características em relação ao projeto e essas outras subdivisões que estão aqui nessas perguntas, e vão indicar o caminho. De acordo com as informações coletadas, o resultado pode indicar que o Agile é um bom caminho, ou não. Como é que eu transformo isso num sistema informático? É isso. Como é que eu transformo isso numa ferramenta?

***Participante 1*:** Eu começo pelo projeto. Ou seja, em primeiro lugar, se isso será o projeto de uma casa. Se algo nos é exigido, como o Participante 2 estava a falar, se se tratar de um projeto legal, os requisitos precisam estar muito bem documentados, muito bem explícitos. Pode ser o ponto de partida. Eu também diria a tecnologia que se vai utilizar e a experiência que a equipa tem nessa tecnologia. E também o número de elementos envolvidos no projeto.

***Luis Mattos*:** Ok Participante 1, mas como é que eu questiono isso numa ferramenta? Que tipo de recurso que eu uso numa ferramenta para isso?

***Participante 2*:** Não sei se estou a perceber a pergunta.

***Participante 1*:** Basicamente tu tens um software em que tu vais inserir características que vão te dizer se tu vais de um projeto Agile ou não.

***Participante 2*:** Ah ok, estou a perceber. Eu acho que é isso, é ao nível de incerteza dos requisitos que nós temos... nível de incerteza não só... o nível de incerteza do número de requisitos que eu tenho e de tipos de requisitos. Como o Participante 1 estava a dizer sobre os requisitos legais, lá a encomenda está totalmente fechada. Pode até ser algo construído de forma é incremental, mas neste caso não é algo que eu vou percebendo e tendo mais certeza daquilo que eu preciso à medida que vou avançando. Isso já é um bocadinho de uma variável um pouco subjetiva, mas que dá para transformar em variáveis objetivas. O número de áreas envolvidas também parece que poderá ajudar, está relacionado, ou seja, quanto mais áreas envolvidas, mais áreas a interpretar algo, menos precisos serão os meus requisitos à partida. O número de sistemas envolvidos poderá também ser uma variável.

***Luis Mattos*:** Tudo isso que vocês estão falando é rico, muito rico, só que ainda tem a ver com a primeira parte. A segunda parte na verdade é... eu vou tentar perguntar de outra forma. Eu preciso de um sistema que vai me ajudar a tomar uma decisão sobre algo. Neste caso, que tipo de funcionalidades essenciais esse sistema tem que me oferecer?

***Participante 4*:** Acho que já percebi. Tu queres saber algo do gênero... tem que ser um sistema que possa receber parâmetros de decisão, que queres poder priorizar e dizer se este parâmetro é mais ou menos importante que outro, consoante isso existirá um motor de regras que vai levar a uma decisão.

***Luis Mattos*:** Exatamente isso!

***Participante 4*:** Ok. Eu diria então que vais ter que ter um limite de introdução de parâmetros, portanto podes ter até dez critérios, por exemplo. Vais ter que poder atribuir um peso para cada critério, ou seja, esse critério vai ter uma preponderância de dez, vai ser muito importante para decidir se vais de Agile ou não. Tens que poder recalibrar esses pesos ao longo do tempo, poder ir ajustando, e tens também além disso, tens que poder depois programar os possíveis auto campos, ou seja, o parâmetro X tem uma importância de 10 e caso a resposta seja à esquerda então são 10 pontos a favor do Agile e 1 ponto a favor do *Waterfall*. O segundo parâmetro já se a pontuação for 5, então à esquerda são 5 a favor do *Waterfall* e à direita são 5 a favor do Agile. Tudo isso no final, com um algoritmo que vai ter em conta todos esses pontos, vai te dar uma decisão. Se quiseres uma coisa mais absoluta, o sistema vai te recomendar o Agile ou recomendar o *Waterfall*, mas se quiseres algo mais subjetivo para poderes analisar depois, o resultado seria algo que te daria um intervalo de confiança, por exemplo, de 70 porcento para um projeto Agile e 30 porcento para um projeto *Waterfall*, e daí tu podes ver mais claramente se aplicas uma, se elas são mais ou menos parecidas ou se há clara preponderância de uma. Daí é elaborar esses algoritmos e as fórmulas para implementar essas regras.

***Participante 3*:** Eu diria que além disso, se eu tivesse uma forma de além de determinar qual o modelo, se a ferramenta pudesse me indicar qual o tipo de ferramenta de mercado seria melhor para o controlo do projeto, seria interessante também. Por exemplo, eu poderia ter aqui determinadas características da minha empresa, eu entro com os dados e o sistema diz que o JIRA é uma boa ferramenta para o meu projeto. O JIRA no modo Kanban, por exemplo. Ou o JIRA no modo Scrum. Ou sugerir o ETM como a melhor escolha. Talvez não indicar a ferramenta propriamente dita, mas me orientar a buscar no mercado as características que a ferramenta deveria oferecer. Isso seria interessante porque hoje em dia, você falar só em metodologia sem dizer qual ferramenta pode suportar aquela metodologia acaba ficando algo muito solto, porque as pessoas precisam de um fluxo a ser seguido, e as ferramentas são feitas em cima de fluxos. Então dizer se é Agile, mas sem um fluxo, fica faltando algo.

***Luis Mattos*:** Eu queria explicar o porquê desse *focus group* estar limitado a discussão dos aspetos que podem ajudar na opção por uma abordagem Agile sem especificar a metodologia ou ferramentas a serem utilizadas. Porque neste caso, exigiria um conhecimento nesse grupo que só as pessoas com profundo expertise nas quatro metodologias que são foco do estudo tem, como o especialista que eu entrevistei. No fundo a ferramenta vai te dizer: vou de Agile ou não vou e, dentro das quatro metodologias mais utilizadas, qual é a mais indicada e o devido grau de aderência. Tipo assim, o resultado pode apontar com 70 porcento de força o uso do Scrum, mas existem outras características combinadas que podem apontar também o Kanban com 50 porcento de força. A ferramenta não vai tomar a decisão por você, ela vai te dar subsídios para que você tome a decisão, vai descer ao nível da indicação de uma metodologia ou até mais de uma, mas não tanto ao nível de ferramentas. Isso pode ser uma sugestão de trabalhos futuros.

Participante 6**:** Uma coisa que tu tinhas que agregar de alguma maneira é uma forma de parametrizar a própria organização. Tens que perceber como é que a organização está estruturada. Se ela é matricial ou se é por projetos, porque isso influencia na opção por um Agile ou um *Waterfall*. Depois, quando já desces lá ao outro nível, aí já é onde há mais uma cultura de projetos e de pessoas, de equipas tal como estão a decorrer, como é que se pretende comunicar… conhecimentos mais aprofundados que as próprias equipas podem informar.

Participante 2**:** Um requisito não essencial, mas que seria bom ter e penso que seja complementar é a análise dos recursos que constituem a equipa. Se não tivermos equipas definidas, qual o histórico de um determinado profissional relativamente aos projetos com tais características? As variáveis de valor que não estamos a utilizar para determinar se o caminho é Agile, mas qual a metodologia Agile iremos utilizar, podermos usar também para escolher a equipa. Não é um requisito essencial, mas pensei nisso aqui.

Participante 4**:** Eu penso que o funcionamento desse sistema parece um pouco com uma rede neuronal em que todas as características que falamos antes seriam os nós de entrada e a decisão seria o nó de saída. Incorporando também o *feedback* de *lessons learned* para projetos semelhantes, mas que ocorreram mal no passado, não darem a mesma resposta.

--- // ---

Appendix G – Focus Group Transcript - 2nd Group [English]

Focus Group data	
Moderator	Luis Fernando Bordeaux Mattos
Participants	Six IT project managers from a Portuguese organisation in the Banking sector (under non-disclosure agreement). Intermediate level of experience with Agile methodologies (average of two and a half years). The organisation currently uses Agile extensively in its IT projects.
Location	Lisbon, Portugal.
Date and time	29th of January 2019.
Duration	49min
Comments	The language used in the activity was Portuguese (Portugal / Brazil). This record presents the translated version to English.

Part I

It encompasses questions that will explore the aspects relating to choosing an Agile approach as a guide to the management and execution of a software development project. The objective is to collect the impressions and opinions of the participants on the pertinent questions formulated. These questions have been elaborated based on the review and analysis of Agile literature.

Question 1/5

Luis Mattos: Under what conditions would you suggest an organisation to opt for an Agile methodology?

Participant 1: I think we can use Agile a bit in the composition of what is expected of the project. As we move forward, we detail the scope, the requirements, and the methodology itself is adapting as we deliver. Therefore, I think if we have a very detailed scope, we can opt for a more traditional methodology, like Waterfall, but if there is uncertainty, I would say that Agile would be a more appropriate approach.

Participant 2: I agree and would like to add on what Andre said, but instead of talking about the scope, I would talk about requirements. If my requirements are well known, if I know exactly what the requirements of my project are, I think there is no reason to use Agile, unless I'm in a software house, where you have a portfolio to be carried out. If we are in an environment where we know well what the requirements are, where they are very well defined, such as a Legal project, I know exactly what it is, so why would I go for Agile? I clear my requirements, analyse the impacts that I have in my solution, and plan the implementation of it. I think the notion of the requirements, the degree of uncertainty on the requirements, are the conditions that allow me to choose a methodology or another.

***Participant 3*:** I think the key factors would be to check whether the team or the company is projectized or not, whether the requirements for the project are well defined or not, and how the configuration management in the organisation works. These are factors that I think would be basic for deciding on an Agile methodology, starting with a hybrid approach, or even staying in the line of a Waterfall... something more traditional.

***Participant 4*:** I would like to agree and complement. I think the question of requirements would probably be the most important characteristic in choosing Agile or another non-Agile methodology. But the structure of the organisation itself as well. Participant 3 mentioned about whether it's projectized or not. Whether there is the possibility of teams working together or there are many bureaucratic barriers, and also the question of the time. I think, conceptually, Agile also arose from the need for some companies to have a slot of hours to be consumed rather than a project, and by having this slot of hours they can compromise on functionality. There are also companies that work with iterations, defining these time constraints and building the deliverable from there.

***Participant 1*:** I would like to add one more point that has to do with the knowledge over the various features. I think Agile leads to the proximity that facilitates the mitigation of gaps, whether functional, whether in development or even business. Having a good knowledge spread among all areas involved in the project, I think Agile is also justified. Since there is less knowledge at the outset, and more interaction is required, I think Agile can be useful in helping to elucidate and complement requirements.

Question 2/5

Luis Mattos**:** Regarding the aspects of the ORGANISATION, more specifically within the MANAGEMENT level, what are the characteristics and behaviours that you think the leaders of the organisation should have to work with an Agile methodology?

Participant 3**:** In the first place, I believe the leader should know how to delegate and know your project team s/he is managing because if you know your team, you know how far you can go, and you trust, and having confidence, you end up having more tranquillity to delegate. In addition, when you delegate, it gives the team the whole environment for, let's say, create a sense of responsibility. So, the delegation has to do with the responsibility of the team itself.

Participant 4**:** I think it's important to know how to fit into the team and not only see everything from above, so to speak. Often, it's necessary to be in close contact with the team to be within the themes, that is, almost at the same level. This is important so that you can treat or expedite something and not be, let's say, at a higher level where you can be seeing the themes from very far. It's important to be close to the team and able to respond to that call. Create a relationship with the team to unlock things as quickly as possible.

Participant 1**:** You need to be versatile at the decision-making level. We can have a well-defined iteration, but nothing guarantees that the iterations ahead will follow along the same way. We may have to adapt, either due to changes in the market or due to internal changes. And there the decision has to come from who has the authority, as well as the guidelines to act. Therefore, you have to be versatile in order to be able to justify the reasons for the changes and the transitions, taking the responsibilities, and being also assertive in what you propose.

Participant 6**:** Assuming that you are committed to the organisation's goals concerning the implementing an Agile approach, it is also important to be able to measure the benefits that the methodology brings over time and to think about how you will disseminate the process among the various Agile teams. I think this is the main driver for the change and also ensures that people are all working together towards the same objective.

***Participant 2*:** I think the team has a lot to do with the value that is delivered to the organisation. While in a methodology like Waterfall I get all value through the requirements that have been defined at a single time, here [in Agile] I will receive them [the requirements] over time, that is, over time I will perceive value, and that's why I believe the exigencies must be higher at the beginning of the project. That's why it is also important to calibrate the team, and especially the person who defines the priority of deliveries. Calibrate well what is priority, what is value for the organisation or not. I think this will be key. In Waterfall, I may be sitting waiting for the value to be delivered... the final product, but here [in Agile] no, here, as you know very well, the project can be a little longer or not, but there will be periodic deliveries, and the first deliveries should have more value than the later ones, that is, they should have more urgency for the organisation than the following, and because of this, you will need to have a higher level of exigencies and more closeness. I think it has been said here before, that more presence, more closeness is urgent.

***Participant 4*:** Talking more objectively about the characteristics, I'd say confidence, commitment, responsibility, and also team's recognition. All of this, as well as greater transparency within the team and the leader himself.

Question 3/5

***Luis Mattos*:** Regarding the aspects of the ORGANISATION, more specifically within the TEAM, what are the characteristics and behaviours that you think the project team should have / present to work with an Agile methodology?

***Participant 4*:** I would say responsibility mainly because the team is the one that ensures things are done and that complies with the plan that was agreed. I also believe in team spirit, because a team has to be cohesive, and often the lack of this team spirit leads to incomplete deliverables due to disagreements within the team. The knowledge of one person fits into the needs of another one, and vice versa. So, I would say that responsibility and collaboration are two essential things.

Participant 1**:** And being humble. Being humble to recognize that alone I may know very well my domain, but yes, I'm humble enough to say when I don't master a specific theme, ask for help, and receive input from the project team.

Participant 2**:** I believe that experience is also an important aspect. It's a bit related to being humble. And good behaviour, which is also connected to team spirit. It's expected that I go after my job and not that I wait for it [the job] to be brought to me. It's up to me to go and get it [the job]. I must determine my pace, but my pace is also "controlled" by the team because the team observes my pace. So, if I have a bad pace, I will feel embarrassed if I have a good pace, I will feel enthusiastic. And I think that is one of the aspects that also speeds up deliveries when the team has good behaviour.

Participant 3**:** I believe in sense of unity. The task does not belong to one person; the task belongs to the whole team. So, the idea is that you get out of your "own world". You do not just say, look, I did my part and that's it, but rather, I work in the sense that my part fits in the part of another person so that the whole work is delivered.

Participant 6**:** There is also this multidisciplinary concept, therefore, not only giving your best for the company, though I think this is fair, but also ensuring, for instance, that there will be a tester helping the developer, as well as a business analyst supporting him [the developer], since his job does not end with programming a feature that will be delivered next, but only when the whole development cycle ends. And this is when the team earns its value, with the delivery of what was agreed.

Question 4/5

Luis Mattos**:** Regarding the aspects of the PROJECT, more specifically in the ENVIRONMENT, what are the environmental factors for the project that you consider to be more characteristic of initiatives eligible to use an Agile methodology?

Participant 1**:** I prefer to focus on the team's mindset and not so much on the environment itself. And when I say mindset, I compare it to people who have been in the job market for a long time, in more old-fashioned organisations, so to speak. There are organisations pre-formatted by several methodologies, whose people have been working the same way for many years, and I compare them with those... I wouldn't say not less experienced people, but younger people, more open to the adoption of new methodologies. The problem is that in deeply old-fashioned organisations there is a kind of a blockage to new ideas and new ways of working. So, I think the environment also has to promote this openness, it should allow people to innovate in the ideas they bring, in the methodologies they want to experiment... and I would say that in younger environments, in companies created more recently, the environment is more conducive to innovation than in old-fashioned companies.

Participant 4**:** Talking about the reaction to change and innovation, I believe the creation of open workspaces, as we can see nowadays, was a response to this. I'm there in my office, and then I come across an issue, then I send an email, and I wait for an answer… Nowadays, people want this answer right away, and open workspaces promote this proximity among people to talk and solve problems immediately.

Participant 3**:** I think this has more to do with the culture of the company, the directive layer, the organisation as a whole that wants to show what the role of the team is, what it delivers, and how the pieces of the puzzle fit together. Nowadays, you usually see many companies that already work in a home-office model due to the increasing limitation of using physical workspaces, motivated by several factors. And when we listen that Agile only works well if people share the same workspace, if this is true, Agile might disappear in the future, because there is a clear trend for companies to decentralize their business operations, especially in the context of service providers. There are many companies in the job market that already work almost 100 percent in a home-office mode. Top companies. I think organisations like these can work well with Agile as long as they foster your teams to clearly define the role of each one and align the expectations within a particular project and in the organisational context itself.

Participant 4**:** I just want to add on the point that Participant 4 mentioned about the open workspaces. For me, no doubt I agree with the use of them, but I would go a bit further because there are open workspaces that are very crowded; thus, it is important to make sure that the workgroups are close. It doesn't help if groups are in the same workspace, but scattered, because you will lose the synergy provided by this environment.

Participant 2**:** I agree with the use of open workspaces because the trust part relates to face-to-face interactions, although I think this is also possible at a distance. I think it's possible, but the company has to be aware of what it's doing. Another point that I think has to do with this environment variable is if I work with branded products. For example, Microsoft has Word and PowerPoint and knows how to manage both separately. If we are talking about a company with no experience in managing by product, let's say, I think it will be an environment where Agile may not work in the best way. Or, if it's not viable to have a product approach, it must be something where product development could occur incrementally, where you don't lose anything moving forward a bit with something instead of delivering something else. I would say this is the way.

Participant 4**:** I believe that an environmental factor is the culture of the country itself. Both Portugal and Brazil are traditionally countries that are conflict-averse. In northern Europe, people are more used to openly point out if something is wrong, and we [Portuguese people] culturally take it personally, but it doesn't mean that we are wrong, it only means that the work we produced was not what was expected. This is very noticeable in south Europe and is a factor that often prevents us from advancing faster in solving problems. Because we are afraid to offend someone or to go over someone when we already have all the answers on the table, but we lack the endorsement of someone who is absent and who only will come back within a week, and we wait a week for the resolution of the problem.

Question 5/5

Luis Mattos**:** Regarding the aspects of the PROJECT, more specifically in the PRODUCT scope, what are the characteristics related to the product of the project you think are most relevant in determining an Agile approach as a development methodology?

***Participant 4*:** First, I would verify if it's a closed product or if it is a product that is planned and that undergoes changes over time. I think if it's a closed product, you can apply a Waterfall methodology as Participant 2 has already mentioned because the requirements are closed. But if it's a software where you will release a first version, followed by a second version and so on, there is a condition that favours the use of an Agile methodology.

***Participant 1*:** I think it doesn't depend on the product, honestly. I think that, regardless of the product, you can apply both methods. It depends more on the organisation's environment, depends more on what we know and what we don't know about the product. These are the characteristics that will tell us whether or not we can apply an Agile methodology.

***Participant 6*:** Here I think if you are developing a product and you have that crazy desire to invest in a new methodology, if the product itself allows you to grow it incrementally and iteratively, Agile is undoubtedly the best approach since in Waterfall you usually want to have the end product all the way up to putting it on the market. So, one thing that you must pay attention to is the time-to-market. The importance of launching something with the minimum set of important features, in which case the Agile approach is more appropriate. So, I would say that the characteristics of the product also point to that direction, namely the market, and if the product itself can actually be worked in an incremental way.

***Participant 1*:** I agree with you. If it's a product that we can deliver in small chunks, Agile definitely suits best.

***Participant 3*:** I think the product issue is much more related to the customer's need. If it makes sense for the customer to receive something in advance, it facilitates the use of an Agile approach. In Waterfall, you could do some planning so that you make partial deliveries, but in that case, you'd be, let's say, more restricted, stuck to the milestones set at the beginning. Agile is freer in this sense because I don't need to say exactly what I'm going to deliver upfront. I'll deliver in small chunks, and these chunks can be moved according to the customer's needs or the market changes. But in Waterfall, although I can break the deliveries, I need a closed scope up front and define each part of the product I'll deliver in each release. If at some point the customer changes his mind, change control in Waterfall is much more complex than doing it in an Agile environment.

***Participant 4*:** I don't fully agree with the belief that Agile is the only option when the delivery of the product is divisible. I believe the two main contributors here are: knowledge of the requirements for building the product and the time to implement it. Sometimes, splitting a product in increments can even get more expensive. Depending on where we are, we may be able to demobilize the teams, and if we already know the requirements upfront, I would say that Waterfall probably would be a cheaper option. I think this is very much related to the knowledge and the degree of uncertainty, like Participant 1 said about the value of the deliveries.

Part II

Requirements gathering of the information system that will be designed to implement the decision model derived from the analysis of the aspects discussed in Part I. The objective is to collect the opinion of the participants about the ESSENTIAL features for this information system.

Question 1/1

***Luis Mattos*:** An information system will be developed, aiming to implement the decision-making model resulted from the analysis of the aspects discussed in the first part of this focus group. The main objective of the system is to indicate if the project is eligible for an Agile approach or not. In this context, what would be, in your opinion, the ESSENTIAL functionalities of the system?

***Participant 3*:** I think one of the requirements would be the knowledge that the team has on that product, on that system, on the software being developed. The more knowledge I have on scope and technology, the easier it is, between quotes of course because this is not math, to apply a more traditional method, like Waterfall. If I have an environment where I can't see ahead, where I have a more limited view, Agile makes a lot more sense because I develop, I verify, I test, and can adjust the project path. I think this question...

Luis Mattos**:** Allow me to try to clarify a bit this question. The context here would be as follows, the first part of this study involves a comprehensive literature revision, plus the analysis of all the information we are discussing here, plus data from the Agile expert interview I have already performed, and the idea is: all this information will be analysed and will lead to a decision model indicating what are the characteristics and / or aspects that I should evaluate, which will encompass the characteristics of the organisation, the characteristics in relation to the project, and these other subdivisions that are here in those questions, and will point the way. According to the information collected, the result may indicate that Agile is a good path or not. How do I transform this into a decision support system? How do I turn this into that tool?

Participant 1**:** I would start with the project. That is, first, whether it will be the design of a home, for instance. If something is required from us, as Participant 2 said, if it's a Legal project, the requirements must be very well documented, very explicit. It may be the starting point. I would also say the technology that will be used and the experience of the team in this technology. And the number of elements involved in the project as well.

Luis Mattos**:** Ok, Participant 1, but how do I question all of this using an information system? What kind of features does a tool like that need?

Participant 2**:** I'm not sure I understood this question.

Participant 1**:** Basically, you have a software in which you will insert characteristics that will tell you whether you go for an Agile approach or not.
Participant 2**:** Ah, okay, I get it. I think that's it… it's at the level of uncertainty on the requirements that I have... not only on the level of uncertainty... also the uncertainty level on the number of requirements and the type of requirements. Like Participant 1 said about Legal requirements,

there the order is completely closed upfront. It may even be something built incrementally, but in this case, there's nothing to uncover as I move forward. This is a bit subjective variable, but I think it's possible to turn into objective variables. The number of areas involved also seems to help, it's related. That is, the more areas involved, the more areas to interpret something, the less accurate my requirements will be at the outset. The number of systems involved may also be a variable.

***Luis Mattos*:** All of these points you're raising are rich, really rich, but they still relate to the first part of the focus group. This second part actually is... I'll try to question it in another way. I need a system that will help me to decide on something. In this case, what kind of essential functionalities does this system need to offer?

***Participant 4*:** I think I've figured it out. You want to know something like… it has to be a system that can receive as input decision parameters, which you want to be able to prioritise and define if these parameters are more or less important than others, depending on that, there will be a rule-engine that will lead to a decision.

***Luis Mattos*:** Exactly!

***Participant 4*:** Okay. In that case, I would say that you will need a parameter entry limit so that you can have up to ten criteria, for example. You will have to be able to assign a weight to each criterion, that is, a criterion with a preponderance of ten, for instance, it will be very important to decide whether you go for Agile or not. You have to be able to recalibrate these weights over time, adjust them, and, in addition, you must then be able to program possible auto-fields, that is, the parameter X has an importance of 10 and if the answer points to the left, then are ten points in favour of Agile and one point in favour of Waterfall. Is in the second parameter the score is five, then to the left are five in favour of Waterfall and to the right are five in favour of Agile. In the end, all of these parameters will be calculated by an algorithm that will consider all these points and will give you a result. If you want something more absolute, the system will recommend Agile or recommend Waterfall, but if you want something more subjective to evaluate later, the result could be something that would give you a confidence interval, for example, 70 percent point to an Agile project and 30 percent pointing to a Waterfall project, and from there you can see more clearly if you apply one if they are more or less alike or if there is a clear preponderance of one. Hence it is to elaborate these algorithms and the formulas to implement these rules.

***Participant 3*:** I would say in addition, if I had a way beyond determining which model, if the tool could also tell me what kind of tool would be best for project controlling, it would be interesting as well. For example, I could have certain characteristics of my company here, I enter the data, and the system points out JIRA is a good tool for my project. JIRA in Kanban mode, for example, or JIRA in Scrum mode, or even suggest ETM as the best choice. Maybe not indicate the tool itself but help me to search the market using the characteristics that this tool should offer. That would be interesting since nowadays if you talk about methodology without saying which tool can support that methodology ends up getting something very loose, because people need a flow to be followed, and the tools are made on top of flows. So, say if it's Agile, but without a flow, something is missing.

***Luis Mattos*:** I want to explain why this focus group is limited to discussing the aspects that may help in the choice of an Agile approach without specifying the methodology or tools to be used. Because in this case, it would require a knowledge in this group that only people with deep expertise in the four methodologies that are the focus of the study have, like the specialist I interviewed. In the background, the tool will tell you: I will go for Agile or not, and, within these four methodologies, which one is the most indicated and the degree of adherence. It's like this, the result can point out the use of Scrum with 70 percent of confidence, but there are other features that combined may also point out Kanban with 50 percent of confidence. The tool will not decide for you, it will give you subsidies so that you decide. It will to the level of indicating an Agile methodology or even more than one, but not so much at the level of tools. This may be a tip for future study.

***Participant 6*:** One thing that you have to consider somehow is a way to parameterize the organisation itself. You have to understand how the organisation is structured. Whether it's a matrix or projectized organisation, because that influences the choice of an Agile or a Waterfall approach. Then, when you go down to the next level, there's already a culture of projects and people, teams as they are, how you intend to communicate… more in-depth knowledge that the teams themselves can provide.

Participant 2**:** A non-essential requirement, but one that would be nice-to-have and I think it is complementary, is the analysis of the resources that make up the team. If we don't have defined teams, what is the history of a particular professional regarding projects with such characteristics? The variables that we are not using to determine if the way is Agile, but instead which Agile methodology we will use, can also be useful in choosing the team. It’s a non-essential requirement. I just come across it.

Participant 4**:** I think the operation of this system looks a bit like a neural network, in which all the features we talked about would be the input nodes, and the decision would be the output node. Also, feeding the system with lessons learned from similar projects which have performed poorly in the past, aiming not to have the same results.

--- // ---

Appendix H – Focus Group Transcript - 3rd Group [Portuguese]

Condições da atividade	
Moderador	Luis Fernando Bordeaux Mattos
Participantes	Cinco desenvolvedores de software de uma organização brasileira do sector de Telecomunicações (sob acordo de não divulgação). Nível elevado de experiência com metodologias Agile (média de cinco anos). A organização utiliza amplamente e há muitos anos metodologias Agile variadas em seus projetos de TI.
Local	Rio de Janeiro, Brasil.
Data e hora	20 de março de 2019.
Duração	1h40min
Comentários	A língua utilizada na atividade foi o Português (Brasil).

Parte I

Engloba as questões que irão explorar os aspetos relativos à escolha de uma abordagem Agile como guia para as ações de gestão e execução de um projeto de desenvolvimento de software. O objetivo é coletar as impressões e opiniões dos participantes às questões pertinentes formuladas. Tais questões foram elaboradas com base na revisão e análise de literatura sobre Agile.

Questão 1/5

***Luis Mattos*:** Em que condições sugeriria a uma organização a opção por uma metodologia Agile?

***Participante 1*:** Pensando numa organização, eu acho que, independente da complexidade, da hierarquia ou do tamanho dela, pois eu acho que isso não interfere, eu acho que um aspeto fundamental para a organização usar uma abordagem Agile é, primeiro, que as equipes estejam fisicamente no mesmo local e que não sejam equipes muito grandes. Ela [equipe] deve ser reduzida para que você possa ter interações mais frequentes, para você poder tomar decisões mais rapidamente e tratar os problemas de uma forma mais ágil.

Para mim, ter a equipe trabalhando próxima também ajuda na questão da multidisciplinaridade do time, o que é uma característica do Agile. O tamanho da equipe não deve passar dos 10-12 integrantes. O time também precisa ser o máximo possível blindado de interferências externas para que ele possa dar foco no trabalho a ser feito. A tomada de decisão tem que ser feita em grupo, de preferência no mesmo local de trabalho do time, através de interações frequentes e rápidas. De uma forma geral, para a organização trabalhar com uma metodologia Agile, na minha opinião, seriam essas algumas condições importantes.

Participante 2: Eu concordo com o Participante 1 na questão da equipe reduzida, principalmente levando em consideração os canais de comunicação que existem em função do tamanho desse time. Quanto mais pessoas numa equipe, mais complexa é a comunicação, então, é um aspeto que influencia bastante. Uma sugestão para esse tipo de ambiente é tentar fomentar um ambiente menos burocrático. Outro ponto que eu penso que contribui é a empresa ter uma visão clara de qual direção ela quer seguir, onde ela quer chegar, e comunicar essa visão, e para isso nós temos um benefício no Agile pelo fato das suas metodologias pregarem maior transparência. Outra questão é que no Agile nós temos a oportunidade de errar o quanto antes e não sermos reprimidos por isso, pelo contrário, porque isso ajuda a identificar problemas mais cedo. O fato de no Agile nós trabalharmos com iterações curtas e frequentes também nos permite errar e corrigir o rumo das coisas em tempo de não comprometer o projeto. No Agile tem essa questão bem forte de ser adaptável, e isso traz vantagens quando você tem um grau de incerteza mais elevado, principalmente no início do projeto. A organização para ser ágil precisa reagir rapidamente ao ambiente externo e precisa produzir ao mesmo tempo, e o Agile proporciona esses benefícios. A forma de pensar, independente da ferramenta ou do processo que você vai utilizar é que é o core no Agile. O Agile mudou o modo de pensar e de agir sobre aquilo que é importante para desenvolver *software*.

Participante 3: Eu gostaria de complementar. Sobre as condições, eu vejo como principal uma necessidade, a necessidade de gerar valor contínuo em algum lugar, seja num software ou outro tipo qualquer de produto ou de serviço. Então, eu entendo que se a organização lida com um produto ou serviço que precisa gerar valor periodicamente, seja mensal, trimestral ou semestral, então eu acho que o Agile atende bem. Eu acho também que uma das condições é quando o escopo muda com frequência. Eu vejo assim, para construir um prédio as empreiteiras já partem de uma previsibilidade do preço e quantidade dos insumos, de um número médio de homens-hora envolvido em cada etapa da construção, e caminham com aquilo com um mínimo de mudança até o final, dependendo das influências externas, é claro. Na construção de software é muito diferente. Na construção de software em geral o cliente faz a demanda, mas ele não sabe muito bem o que ele quer ainda, os requisitos não estão muito bem definidos ainda, e nesse caso o Agile estimula o time a construir o requisito junto com o cliente e não ficar numa posição passiva, esperando que o cliente defina [os requisitos] e entregue para o time, mas, para isso o time precisa de um bom grau de autonomia, o que também é estimulado no Agile. Tem que ser possível que as partes assumam os riscos envolvidos, que tenham voz ativa e que tomem suas próprias decisões, porque se o time não puder fazer isso ele vai precisar de suporte. Na minha opinião, são essas condições, ou seja, valor tem que ser gerado de forma constante e o cliente pode querer ter algo que ainda não está muito claro para ele.

Participante 4: Para mim, uma das primeiras condições é o grau de a maturidade da empresa na metodologia [Agile]. Se ela [empresa] possui um conhecimento mais profundo ou mais superficial do tema. Considero também primordial a postura da liderança da organização diante de processos de mudança e como ela [liderança] entende que essa mudança pode ajudar a alavancar os seus resultados. Então, complementando o que já foi dito pelos colegas, na minha opinião, as condições estariam também nesses dois aspetos: o grau de conhecimento, de domínio, da organização na metodologia [Agile] e a maturidade dessa organização para assumir o compromisso de implantar um processo novo, algo que é sempre crítico porque envolve mudança.

Participante 5: Na minha opinião, a primeira coisa a se avaliar é se a empresa tem a capacidade de fazer mudanças rapidamente. Fala-se muito no Agile sobre a questão da capacidade de adaptação para que se possa entregar de forma mais rápida, então, a primeira questão que me vem à cabeça é saber se a empresa é capaz de fazer esses ajustes com a agilidade necessária. Se a resposta para essa pergunta for "sim", então eu considero que o primeiro passo para a empresa adotar uma metodologia Agile foi dado.

Participante 1: Eu gostaria de acrescentar mais uma coisa. De tudo que foi colocado aqui, falou-se muito da questão do escopo volátil, falou-se da questão da comunicação, da liderança, de autoridade, no final das contas isso tudo está centrado nas pessoas. São as pessoas é que vão fazer isso tudo acontecer. Se você não tem um ambiente colaborativo, nada funciona. Então, eu acho que as pessoas são importantes.

<u>Questão 2/5</u>

Luis Mattos: Em relação aos aspetos da ORGANIZAÇÃO, mais especificamente no âmbito da LIDERANÇA, quais são as características e comportamentos que julga que os líderes da organização devem ter / apresentar para trabalhar com alguma metodologia Agile?

Participante 3: Eu entendo que a liderança não deve ter apego ao cargo porque o Agile proporciona o espaço para o grupo se posicionar. E não é dar espaço fingindo estar dando espaço, mas efetivamente fazer com que o grupo seja ouvido, e isso é um desafio porque ainda há em muitas organizações aquela questão de quem indica o caminho, quem diz como as coisas devem ser feitas, é o chefe, é a liderança. Então, estar aberto a ouvir, ser paciente, considerar as opiniões de todos os envolvidos é o tipo de comportamento esperado da liderança para trabalhar com Agile. Eu vou usar uma palavra que já está um pouco desgastada hoje em dia, mas seria o "servir". Eu acho que é uma mudança de comportamento neste sentido. É você virar para o seu time e perguntar, "O que vocês precisam para executar o seu trabalho da melhor forma? Como eu posso ajudar vocês?" Eu acho que essa é uma mudança muito significativa da forma de pensar e de se comportar, isso somada a uma presença física constante. É muito comum ouvir do seu líder, "a porta da minha sala está sempre aberta", quando na verdade eu penso que nem deveria existir uma porta, porque na verdade não deveria existir uma sala separando a liderança da equipe. Então, eu penso que é importante que a liderança esteja presente, mas que seja também uma presença "servidora", porque o líder não consegue enxergar toda a complexidade. Ele não é e nem tem que ser onipresente, isso é impossível, mas o time somado é muito mais do que uma única pessoa, e isso faz com que os olhares se multipliquem e permite que cada um possa, por exemplo, identificar aspetos de risco que seriam menos passíveis de identificação individualmente. E, nesse sentido, também é bastante positivo que cada um tenha a liberdade para apontar esses riscos sem receio. Eu penso que essa servidão do líder, nesse sentido, ajuda muito.

***Participante 4*:** Tentando resumir um pouco, essas características e comportamentos estão bem ligadas com o que o Participante 3 falou agora, saber lidar com pessoas. Se o líder não souber lidar com pessoas, não entender que pessoas erram, que isso é natural e saudável num ambiente ágil, a metodologia [Agile] não vai funcionar. O líder precisa não só entender, mas conhecer as pessoas com quem ele trabalha. Isso é primordial. Não só no Agile, mas principalmente no Agile.

***Participante 2*:** Pegando esse gancho do lidar com pessoas, eu penso que a questão da dinâmica, de você conseguir "ler" os comportamentos ajuda muito na gestão individual dos integrantes do time. Complementando o que os meus colegas disseram, saber delegar, saber ouvir e se comunicar de maneira clara para mim é fundamental. Muitas vezes é difícil para nós exteriorizar o que está na nossa mente, então, é importante procurar uma linguagem comum a todos. Os times multidisciplinares são formados por pessoas com crenças, valores, e experiências diferentes, e por isso podem vir a entender a mesma coisa de maneiras diferentes, por isso é importante que o líder saiba se comunicar de forma clara, passando uma visão única sobres os temas para o time.

***Participante 1*:** Só para complementar, eu acho que quase todos os aspetos e características já foram falados aqui. Quando se fala de um líder, ele estando nessa posição, já deveria estar claro que entre seus papeis e responsabilidades está o fato de que ele é um gestor de pessoas. E isso é um bocado do estar presente, como fá foi dito aqui. Quando você tem por exemplo, um *Product Owner* ou um *Scrum Master* na equipe, o líder precisa dar autonomia para que essas pessoas tomem decisões. Eu acho que as entregas acabam acontecendo de forma mais eficaz. O líder atribui a responsabilidade, ele deixa que as pessoas sigam o seu caminho, então elas vão ganhando mais conhecimento e mais confiança. Então eu penso que o líder tem que deixar as pessoas trabalharem. As coisas vão fluir de uma forma mais tranquila.

Participante 5: Nós já falamos bastante sobre a visão do líder no Agile, como o líder deve se comportar e tal. Algumas das características mencionadas eu diria que nem seriam específicas do Agile já que são características que um líder deve apresentar no dia-a-dia. O que eu queria dizer, é que eu penso que no Agile o líder deve estar disposto a enfrentar mudanças frequentemente, porque, assim como nós estamos discutindo agora, nós estamos de forma quase permanente envolvidos num processo constante de amadurecimento e de consolidação da metodologia [Agile]. Não existe um modelo fechado no Agile, onde se dize toma aqui essa receita e segue. Então pensando nisso, eu acho que a liderança tem que ter em mente que é importante estabelecer um processo de melhoraria contínua, questionar sempre o que pode ser feito para ampliar os benefícios já obtidos. Esperar que haverá mais mudanças e saber controlar essas mudanças, e aí sim, investido do papel que todo líder deve ter, saber conduzir essas questões de forma natural.

<u>Questão 3/5</u>

Luis Mattos: Em relação aos aspetos da ORGANIZAÇÃO, mais especificamente no âmbito da EQUIPE, quais são as características e comportamentos que julga que a equipe do projeto deve ter / apresentar para trabalhar com uma metodologia Agile?

Participante 2: Eu penso que a equipe precisa ser experiente, ela precisa constantemente buscar se atualizar. Na comunicação ela deve sempre buscar entender o que outras equipes Agile dentro e fora da organização estão fazendo e mais, entender que as equipes têm que ser transcendentes. Uma equipe Agile tem que ser capaz de iniciar um trabalho e saber terminá-lo. A equipe precisa estar ciente dessa responsabilidade. A partir do momento que eles [membros da equipe] sentem que são parte de uma coisa maior, o trabalho acontece de uma maneira mais fluida. Cada integrante vai puxando o outro, principalmente quando vê que tem alguém destoando, saindo do rumo, e elas têm que se comunicar entre si sem a necessidade de um facilitador. Às vezes pode ser necessário alguém para facilitar essa comunicação, mas é importante que, de uma forma geral, ela flua sem a necessidade dessa pessoa. Então eu penso que a comunicação deve ser clara. Como característica, eu acho que as equipes precisam ter conhecimentos diversos para poderem trabalhar de forma transcendente. A equipe deve ser autônoma, mas ela precisa buscar também essa transcendência. E o líder, ele precisa deixar a equipe sobressair. O bom líder não aparece. Ele faz o trabalho dele, quase não é visto para que a equipe seja a protagonista, mas as pessoas que fazem parte da equipe precisam buscar isso, as pessoas têm que buscar o ciclo de responsabilização.

Participante 3**:** Eu penso que a liderança tem um papel chave na montagem da equipe Agile, mas o integrante da equipe precisa mostrar uma característica importante que é ser proativo. Ele precisa saber que pode e deve se expressar livremente quando achar necessário. Outro aspeto importante é o membro da equipe saber o que significa trabalhar em equipe, porque muitas vezes a decisão que ele vai tomar vai impactar o trabalho do colega do lado. Então a decisão não deve ser egocêntrica porque o risco dessa decisão pode impactar não somente nele, mas todo o projeto.

Participante 5**:** Mas isso que vocês estão pontuando não é importante somente nas abordagens Agile. Na minha opinião, mesmo não se tratando de uma abordagem Agile eu acho que o profissional que integra uma equipe de projeto precisa ser responsável, precisa saber trabalhar em equipe, e cumprir o que foi acordado para as entregas. Na minha opinião não é apenas o Agile que requer isso.

Participante 2**:** Realmente Participante 5, não é. O que ocorre é que a forma de pensar e agir no Agile propicia isso. No Agile essas características ficam muito mais evidentes porque elas são estimuladas e são parte essencial da metodologia.

Participante 3**:** Sim, existem equipes de trabalho que já se comportam dessa forma, que tem essas características sem mesmo saber que a metodologia [Agile] existe. Para mim no Agile é, vamos extrair conscientemente essas características e estimulá-las de forma consciente e proposital. Mas tem também ainda, e sempre vai ter, aquele clássico desenvolvedor que senta na cadeira, bota o fone de ouvido e esquece do planeta. Às seis horas da tarde em ponto ele levanta e vai embora. E tem o outro desenvolvedor que está genuinamente interessado em saber a importância, o valor daquilo que ele está desenvolvendo. Qual o valor que vai trazer para o cliente aquela funcionalidade que ele está desenvolvendo. Eu acho que para trabalhar em um ambiente com Agile, esse desenvolvedor que eu chamei de clássico não vai durar muito tempo porque ele vai ter que tirar os fones de ouvido e pedir ajuda, oferecer ajuda.

Participante 5**:** Aí você levantou um ponto que eu concordo totalmente, mas, por exemplo, a questão da responsabilidade, a questão da comunicação, a questão de assumir um papel onde é permitido que ele se expresse livremente, okay, todos esses pontos favorecem o uso do Agile, mas não exclusivamente o Agile.

Participante 3**:** Sim, é verdade, mas como eu falei, o Agile permite que esses comportamentos se manifestem. Eu acho que sem o estímulo, sem a ênfase que o Agile dá a esses comportamentos e características, na minha opinião, seriam mais baixas as chances delas surgirem de uma forma espontânea. O Agile estimula essas atitudes e características.

Participante 1**:** Resumindo, eu acho que não tem muito mais o que aprofundar nesse assunto. Mais uma vez eu acho que nós estamos no âmbito das pessoas. Se a pessoa não é colaborativa, se ela não é transparente, se ela não está disposta a compartilhar informação, ou seja, se ela não consegue se adaptar a esse novo contexto de trabalho em equipe, não vai funcionar porque você pode ter a melhor metodologia, o melhor processo, mas no final vão ser as pessoas que vão ser responsáveis por fazer acontecer.

Questão 4/5

***Luis Mattos*:** Em relação aos aspetos do PROJETO, mais especificamente no âmbito do AMBIENTE, quais são os fatores ambientais para o projeto que julga que são mais característicos das iniciativas candidatas a utilização de uma metodologia Agile?

Participante 3: Eu penso que uma organização que tem uma estrutura hierárquica, como por exemplo, o Exército, esse tipo de metodologia seria uma das últimas opções, embora as forças armadas em geral apliquem e incentivem o uso de equipes multidisciplinares. Tem o conceito do especialista sim, mas mesmo o especialista é treinado para executar bem outros tipos de tarefas, mas eu acho que para você trabalhar bem dessa forma tem que haver uma hierarquia flexível que permita que haja uma comunicação e colaboração livre entre os membros da equipe, e que isso não seja considerado uma quebra de regra ou política interna. Então eu acho que quando uma organização tem uma estrutura muito hierárquica, não favorece a utilização do Agile.

***Participante 1*:** Eu acho que essa questão da hierarquia de fato faz toda a diferença. Nós já falamos aqui da questão da liderança, então você usar isso de forma negativa, como por exemplo, se toda vez que for necessário tomar uma decisão ela precisar ser tomada exclusivamente pela liderança, o que é bastante normal em empresas grandes, eu acho que não favorece em nada o uso do Agile. Mais especificamente na questão de projetos de software com escopo muito volátil, essa característica do alto volume de mudanças é outro fator ambiental que favorece sim o uso do Agile, mas eu ainda penso que a estruturação física da equipe e a governança do projeto são os principais aspetos ambientais que trabalham em favor do uso de uma metodologia Agile, e as lideranças da organização têm um papel fundamental para que essas coisas aconteçam de fato.

***Participante 5*:** Eu vou reforçar esse ponto da proximidade da equipe de projeto. O fato de as pessoas estarem próximas trabalhando no mesmo projeto permite disseminar a informação de forma muito mais assertiva e mais rápida. Os ajustes necessários acontecem de uma forma mais fácil e nos permite também mitigar os riscos existentes. Muitas vezes uma empresa encontra dificuldades pelo fato de ainda trabalhar com equipes de projeto na forma de silos, o que não favorece uma boa troca de informações. Eu cito como exemplo uma experiência que nós tivemos aqui na empresa, onde uma área cliente acionava uma primeira equipe que era responsável por fazer uma, digamos assim, triagem da demanda. Essa área que avaliava a demanda então identificava outras possíveis áreas que deveriam atuar naquela demanda e ela acionava essas segundas e/ou terceiras áreas. Só que a comunicação inicial era feita apenas entre o cliente e a primeira área, e quando havia necessidade de mudança de escopo, de uma repriorização, despriorização ou mesmo do cancelamento da demanda, a comunicação muitas vezes falhava e não chegava às demais áreas que estavam envolvidas de forma sequencial. Então isso gerava muitos problemas com relação às mudanças de escopo, pois elas não seguiam o fluxo adequadamente por falhas de comunicação. Houve um caso inclusive de termos uma equipe trabalhando uma semana em um tema que já tinha sido cancelado. Então eu concordo sim com a importância de ter equipes pequenas para que a comunicação aconteça de forma mais fluída. Eu penso assim.

Participante 2**:** De tudo que já foi falado até agora sobre essa questão do ambiente, o que eu pude captar é que é bastante importante ter um ambiente menos burocrático possível. Juntando tudo o que foi falado, na verdade em vez de termos uma visão vertical, no sentido *top down*, é importante nós termos uma visão horizontal. Na qual a liderança e as equipes de projeto são vistas pelo restante da organização como um organismo único trabalhando com objetivos claros e alinhados entre todos, e a comunicação é bem difundida e madura.

Participante 4**:** Uma linguagem clássica que é usada mais habitualmente na empresa é quando nós afirmamos que "estamos todos no mesmo barco". Se as pessoas não souberem a direção do barco nem a sua função no barco, ou o barco afunda, ou muita gente vai abandonar o barco no meio da viagem.

Participante 2**:** Esse exemplo é excelente.

Participante 1**:** Aí você tem o turnover e a questão motivacional também.

Participante 5: Eu penso que é fundamental que a empresa tenha maturidade suficiente para trabalhar com uma metodologia Agile, senão ela vai ter problemas. E porquê? Primeiro porque você pode ter todas as ferramentas e informações necessárias para implantar e usar um *framework* Agile na empresa, mas se você não tiver uma visão, uma cultura, uma forma de pensar muito alinhada com os valores e princípios do Agile, é muito provável que a empresa não vá colher bons resultados. Se a empresa não tem um *mindset* Agile, as dificuldades vão ser muito maiores porque antes tem toda essa questão de, vamos dizer assim, curva de mudança de pensamento. Não digo nem de curva de aprendizado, pois isso se a empresa tiver profissionais competentes vai acontecer de forma natural e estruturada, mas o Agile demanda mesmo uma mudança na forma de ver e fazer as coisas, e para isso acontecer a empresa precisa estar aberta para essa mudança de *mindset*. E isso para mim passa pela questão da maturidade dela [empresa].

Participante 3: Eu queria complementar com duas coisas. A primeira é que eu penso que não existe uma receita de sucesso para se trabalhar com Agile. A metodologia pode funcionar bem num time e não funcionar em um outro time com perfil semelhante na mesma organização. E a outra coisa seria, quando se submete um grupo de pessoas a um processo de trabalho novo, como o uso de uma metodologia de desenvolvimento Agile, é natural que haja uma reação. Essa reação pode vir do tempo de adaptação que o grupo precisa mesmo para assimilar e começar a trabalhar de uma nova forma, ou pode ser que a coisa toda já tenha desandado, que o processo de mudança esteja indo numa direção que não é a correta porque as pessoas estão reativas a ele. Pra mim esses dois pontos estão relacionados com o que vocês falaram sobre a maturidade da empresa, porque se você não estiver atento a isso, a primeira reação é achar que, e aí eu estou falando do primeiro aspeto, o problema está na equipe onde o Agile não está funcionando e não com algo que não foi feito da maneira adequada no momento da implantação. E no segundo ponto, você precisa ter alguém fazendo esse monitoramento que seja capaz de perceber se a reação a um processo de mudança como esse vem de uma dificuldade natural e inerente à mudança, ou se vem da resistência das pessoas à mudança. Então, para mim, esse são dois aspetos importantes a observar relativamente ao ambiente para se trabalhar com Agile.

***Participante 2*:** Exatamente. Esse ponto que você levantou é muito bom. No Agile não existe essa coisa de não documentar ou de você não ter controle do que está sendo feito no projeto ou de também não ter controle de versões. A questão é que nas metodologias tradicionais essas coisas acontecem na base do comando e controle, hoje é assim ainda. As pessoas precisam de um cronograma para seguir. O cronograma faz parte de um modelo preditivo e a gente sabe que o desenvolvimento de software é um processo empírico. Mudanças vão ocorrer ao longo do projeto. Você ter uma linha base para seguir e um deadline, okay, é parte do plano, só que o período necessário de adaptação para ver se estamos no caminho certo faz parte de um processo de *setup* onde você vai descobrir de fato se a metodologia [Agile] vai atender, se vamos encontrar a nossa cadência ideal, se vamos conseguir determinar a nossa capacidade de entrega. E os diferentes times dentro de uma mesma organização podem ter números diferentes para cada um desses indicadores. Mas concluindo o que eu queria dizer, no Agile nós não vamos deixar de fazer coisas como documentação, controle de atividades ou versionamento, nós vamos fazer tudo isso, mas apenas o que for realmente necessário. Algumas pessoas têm uma visão distorcida desses conceitos do Agile e por isso sim eu concordo que é fundamental ter pessoas experientes na implantação e consolidação da metodologia [Agile], principalmente quando o nível de maturidade da organização é baixo.

***Participante 4*:** E ainda tem a questão do custo envolvido numa implantação desse porte. Dependendo da organização e da abrangência, é claro, embora isso só vá influenciar se vai se perder mais ou menos dinheiro. Mas o ponto aqui é que se trata de um investimento alto que se não for feito da maneira correta, vai se perder o investimento e ainda se corre o risco de levar a empresa a perder os investimentos diretos nos projetos que usaram uma metodologia mal definida.

***Participante 3*:** Existem alguns modelos de maturidade que ajudam a medir o grau de capacidade de uma empresa para implantar e absorver uma nova metodologia como o Agile. Eu não vou aqui citar nomes desses *frameworks* nem me aprofundar em como eles funcionam, mas de uma forma bastante resumida, o limite passa na maturidade dos seres humanos, na maturidade das pessoas que tomam as decisões na empresa. Então, se a maturidade deles for baixa, quais são as chances de a implantação dar certo? Muito baixas. Então parte dessa questão do ambiente é a organização ter essa maturidade.

Participante 1**:** Para fechar essa questão, tem uma frase que eu gosto muito de usar no dia-a-dia e que tem muito a ver com a questão cultural da organização, que é "Pare de começar e comece a terminar".

Questão 5/5

Luis Mattos**:** Em relação aos aspetos do PROJETO, mais especificamente no âmbito do PRODUTO, quais são as características relativas ao produto do projeto que julga que são mais relevantes na determinação de uma abordagem Agile como metodologia de desenvolvimento?

Participante 2**:** Pela minha experiência eu diria que o Agile está mesmo muito ligado a uma visão de produto, do ciclo de vida de um produto, de melhoria contínua do produto. Existe até uma máxima muito utilizada no mercado que é "o uso define o produto." Eu vou citar um exemplo de um aplicativo de um banco. Existe uma propaganda na TV sobre esse aplicativo onde uma pessoa que quer pagar uma conta diz, "Poxa, ter que digitar tantos números pra pagar uma conta pelo celular é muito ruim." E aí uma outra pessoa diz, "Você sabia que o aplicativo do [nome do banco] lê os dados do seu boleto? Você não precisa digitar nada." É essa visão onde eu vou entregar valor para o cliente, vou entregar aquilo que ele necessita para resolver essas "pequenas" coisas cotidianas, como pagar uma conta via celular, é que é característica de uma abordagem "produtizada". Isso vai ao encontro do que foi colocado pelo Participante 3 aqui na primeira questão, quando ele falou que os projetos em Agile têm como característica gerar um produto que tem que entregar valor constantemente. Então, se temos uma empresa que vai entregar um produto como resultado de um projeto, mas não se espera que esse produto vá ou possa evoluir depois de lançado, então nesse caso eu penso que uma abordagem Agile talvez não seja necessária. Pode ser utilizada? Sim, pode. Mas eu penso que o Agile oferece mais vantagens quando se pensa num produto de longo prazo, que vai evoluir ao longo do tempo.

***Participante 3*:** Esse exemplo do aplicativo bancário é um bom exemplo de entrega de valor constante, porque esse banco está sempre buscando agregar uma nova funcionalidade tendo em mente facilitar a vida do cliente. O aplicativo desse banco inclusive oferece todas as funcionalidades para pessoas com deficiência de visão. Isso é gerar valor para o cliente. Eu acho que outra característica do produto é não oferecer funcionalidades que muito dificilmente vão ser utilizadas. Funções que não tem utilidade não geram valor para o usuário. Isso também vai contra outra característica de produtos resultantes de desenvolvimentos que usam Agile, que é a simplicidade da solução. E a simplicidade está intimamente ligada a qualidade, porque código não implementado é código que não dá erro.

***Participante 4*:** Essa questão do produto oferecer simplicidade e apenas as funcionalidades essenciais está ligada ao conceito de MVP, que é outra prática do Agile. Quão modular um produto consegue ser e quão rápido eu consigo entregar valor para o meu cliente, apenas aquilo que é mesmo necessário. Eu não posso gastar o meu tempo para desenvolver algo que não vai ser usado. Outro ponto é que entregar um MVP ajuda a colher feedback antecipado e melhorar o sistema, criar mais valor pro cliente, nas entregas posteriores.

***Participante 1*:** Eu concordo com essa afirmação sobre o MVP, pois tem tudo a ver com a questão, já que é uma característica do produto. Mas eu queria acrescentar um ponto. Falando sobre o produto, para usar uma metodologia Agile o ideal é que ele possa ser quebrável de forma a permitir entrega faseadas. Se você consegue fazer a entrega de um sistema em momentos diferentes, sendo a primeira entrega considerada um MVP, esse MVP vai para o mercado e me permite colher informações de como o mercado recebeu esse novo produto. Isso me dá muito mais insumo para evoluir o produto e entregar algo melhor nas fases posteriores.

Participante 2: Eu acho que uma questão ligada ao produto é onde você deve focar para encontrar a melhor solução, o produto ideal. Eu acho que o Agile tem uma característica que não é exclusiva, mas que se observa com mais frequência nas organizações que trabalham com Agile, que é o fato de o time focar no problema a ser resolvido através do produto e não no produto em si. O cliente pode te pedir para desenvolver um carro, mas se você não perguntar para que uso ele quer esse carro, corre o risco de você entregar para ele uma Ferrari, quando ele precisava mesmo era de um 4x4. Então essa abordagem com foco na resolução do problema é sim uma questão de produto. Não tão relacionada às características do produto em si, mas com a abordagem para o seu desenvolvimento. Eu acho que o que o Participante 1 falou sobre modularização do produto e o que o Participante 3 falou sobre a entrega de valor são coisas que estão conectadas. O princípio de um produto modularizado que começa a gerar valor o mais cedo possível é focar na simplicidade e resolver o problema a que ele se propõe. É isso, o foco na hora de se desenhar e desenvolver o produto tem que ser no problema e não na solução.

Parte II

Levantamento dos requisitos funcionais do sistema de informação que será projetado a fim de implementar o modelo de decisão derivado da análise dos aspetos discutidos na Partes I. O objetivo é coletar a opinião dos participantes sobre quais seriam as funcionalidades ESSENCIAIS para este sistema de informação.

Questão 1/1

Luis Mattos: Um sistema de informação será desenvolvido de forma a implementar o modelo de decisão resultante da análise dos aspetos discutidos na parte I. O objetivo principal do sistema é indicar se um dado projeto de desenvolvimento de software é candidato a uma abordagem Agile ou não. Neste contexto, quais seriam os requisitos funcionais que julgam ESSENCIAIS para este sistema?

Participante 2: Pensando em sistemas que fazem análises, a primeira coisa que eu diria é que ele teria que ter uma funcionalidade de coleta de informações e teria as regras e métricas para escolha da metodologia cadastrada previamente. Como se fosse um sistema de diagnóstico.

Participante 1: Eu penso o seguinte. Essa dinâmica toda está baseada em perguntas e respostas para que se possa chegar a um diagnóstico que possibilite uma tomada de decisão, então eu imagino um sistema onde eu tenha um questionário. Onde eu vou fazendo as perguntas que eu preciso e a partir do momento que eu vou respondendo, no final, vai ter uma regra onde sistema vai fazer esse cálculo e vai gerar um resultado indicando se uma abordagem Agile se aplicaria ou não. Basicamente eu penso que a funcionalidade principal do sistema seria essa.

Participante 3: Eu entendo que as perguntas têm que ser objetivas e algumas perguntas talvez devam ter um peso maior do que outras. A medida que a avaliação vai seguindo, vai chegar em um ponto em que o resultado pode estar muito próximo, nesse caso eu acho que as perguntas vão precisar ser mais específicas de forma a conseguir diferenciar as metodologias, mas eu acho que essas perguntas tem que somar a favor de uma metodologia ou de outra, mas não deve funcionar como um divisor de águas apontando exclusivamente para uma única opção. Eu acho que as perguntas têm que direcionar para um resultado mais voltado para uma boa probabilidade do que uma completa certeza.

Participante 1: Para resumir, eu acho que os requisitos funcionais seriam algo na linha de um sistema que deve permitir um cadastro de um *template* de questionário, com perguntas onde cada pergunta deveria ter o seu respetivo peso e tipos de respostas diferentes. E então tem que ter a lógica por trás fazendo os cálculos em função do que foi respondido para cada pergunta. Tipos de respostas binárias, sim ou não, e também com escalas, para você poder capturar, por exemplo, a frequência com que determinado evento ocorre. Cada resposta dessa vai ter um peso diferente que vai direcionar você para um ou outro caminho.

Participante 2**:** Eu penso que essas perguntas teriam que ser separadas pelo perfil do respondente. Eu vejo inicialmente dois perfis, uma no nível da gestão que talvez vá englobar questões mais voltadas para a estratégia do projeto e da organização, e um outro mais técnico, onde é a camada executora do projeto que vai ter mais subsídios para responder.

Participante 1**:** Essa quebra poderia ser feita por dimensões onde cada um inclusive poderia ter seu respetivo peso. É lógico que no caso desse sistema seria algo mais elaborado, mas só para trazer uma referência seria algo parecido com um *Survey Monkey* onde você cria o seu template de perguntas só que por trás vai ter uma árvore de decisão para te apontar um resultado.

--- // ---

Appendix I – Focus Group Transcript - 3rd Group [English]

Focus Group conditions	
Moderator	Luis Fernando Bordeaux Mattos
Participants	Five software developers from a Brazilian organisation from the Telecommunications sector (under non-disclosure agreement). High level of experience with Agile methodologies (average of five years). The organisation has been extensively using Agile methodologies in IT projects for many years.
Location	Rio de Janeiro, Brazil.
Date and time	20th of March 2019.
Duration	1h40min
Comments	The language used in the activity was Portuguese (Brazil).

Part I

It encompasses questions that will explore the aspects relating to choosing an Agile approach as a guide to the management and execution of a software development project. The objective is to collect the impressions and opinions of the participants on the pertinent questions formulated. These questions have been elaborated based on the review and analysis of Agile literature.

Question 1/5

***Luis Mattos*:** Under what conditions would you suggest an organisation to opt for an Agile methodology?

***Participant 1*:** Thinking about an organisation, I believe that regardless of the complexity, hierarchy, or size of the organisation, because I don’t think it interferes, I think that a key aspect for the organisation to use an Agile approach is, first, that the members of the team are next to each other, in the same room, and that the team isn’t very large. The team should be reduced so that you can have frequent interactions, so you can make decisions faster and deal with problems in a more agile way. Having people working next also helps with the multidisciplinary aspect of Agile teams, which is a main characteristic of Agile. Team size should not exceed 10 to 12 members. The team also needs to be as much as possible shielded from outside interference so that it can focus on the work to be done. Decision-making has to be done in a group, preferably in the same workspace of the team, through frequent interactions. In general, for the organisation to work with an Agile methodology, in my opinion, these would be some of the important conditions.

***Participant 2*:** I agree with Participant 1 on the point of having a reduced team, especially considering the communication channels that exist due to the size of this team. The more people on a team, the more complex the communication is, then, it’s a very influential aspect. One suggestion for this kind of environment is trying to foster a less bureaucratic environment. Another point that I think contributes is that the company has a clear vision of what direction it wants to go, where it wants to go, and communicate that vision, and for that, we have a benefit in Agile because its methodologies demand more transparency. Another issue is that, in Agile, we have the opportunity to make mistakes as soon as possible and not be repressed because of this, on the contrary, since this helps to uncover problems sooner. The fact that in Agile we work with short and frequent iterations also allows us to make mistakes and correct the course of things just in time not to jeopardize the project. Agile has this strong point of being adaptable, and this brings advantages when you have a higher degree of uncertainty, especially at the beginning of the project. An organisation to be agile needs to react quickly to the external environment and must produce at the same time, and Agile provides these benefits. The way of thinking, regardless of the tool or process you are using is core in Agile. Agile has changed the way we think and act on what is important in software development.

***Participant 3*:** I would like to complement. On the conditions, I see as a key point a necessity, the necessity to generate continuous value somewhere, be it in software or any other kind of product or service. So, I understand that, if the organisation deals with a product or service that needs to generate value periodically, be it monthly, quarterly, or semi-annually, then I think

Agile fits well. I also think that one of the conditions is when the scope changes frequently. I see that in order to build a building, the contractors already start from a predictability of the price and quantity of the raw material, from an average number of person-hours involved in each stage of the construction, and go with that with minimum changes until the end, depending on external influences of course. In software development this is very different. In general, in software development the customer doesn’t know very well what he wants yet, the requirements are not very well defined yet and, in this case, Agile encourages the team to build the requirement together with the client and not expect for the client to define them [the requirements] and deliver them to the team, but for this, the team needs a good degree of autonomy, which is also stimulated in Agile. It must be possible for the parties to assume the risks involved, to have an active voice, and to make their own decisions because if the team can’t do these things, they will need support. In my opinion, these are some conditions, that is, value has to be generated steadily, and the customer may want to have something that is not yet very clear to him.

Participant 4**:** I think one of the first conditions is the degree of the maturity of the company in the methodology [Agile]. If it [company] has a deeper knowledge of the subject. I also consider the leadership support in face of change processes and how it [leadership] understands that changes can help in leveraging the company’s results. In addition, in my opinion, the conditions would also be in these two aspects: the degree of knowledge of the organisation in Agile methodologies, and the maturity of that organisation in order to commit to a new process, which is always critical because it involves changes.

Participant 5**:** In my opinion, the first thing to be evaluated is whether the company has the ability to implement changes quickly. There is a lot of talk in Agile about the ability to adapt so that you can deliver faster, then, the first question that comes to my mind is whether the company is able to make these adjustments with the necessary agility. If the answer to that question is "yes," then I consider that the first step for the company to adopt an Agile methodology has been given.

Participant 1: I would like to add one more thing. Of all that has been said here, much is related to the question of having a volatile scope. We have talked about the matter of communication, leadership, authority, but in the end, this is all about people. People will make it all happen. If you don’t have a collaborative environment, nothing works. So, I think people are important.

Question 2/5

Luis Mattos: Regarding the aspects of the ORGANISATION, more specifically within the MANAGEMENT level, what are the characteristics and behaviours that you think the leaders of the organisation should have to work with an Agile methodology?

Participant 3: I understand that the leadership should not be attached to the position because Agile gives room for the group to speak for themselves. And it’s not giving room pretending to be giving room, but effectively making the group to be heard, and this is a challenge because there are still in many organisations the sense that who points the way, who says how things must be done, is the boss, is the leader. So, being open to listening, being patient, considering opinions from everyone involved are the behaviours expected from the leadership to work with Agile. I'm going to use a word that is already a bit worn out these days, but it would be the "serving". I think it's a changing in the behaviour in this sense. It's about to turn to your team and ask, "What do you need to do your job in the best way? How can I help you?" I think this is a very significant change in the way of thinking and behaving, coupled with a constant physical presence. It’s very common to hear from the leader that "the door of his room is always open," when in fact I think that there shouldn’t be a door there, because, in reality, there shouldn’t even be a room separating the leader from the team. So, I think it's important the leader be present, as it is also this "servant" presence because the leader can’t see all the complexity. He’s not, and doesn’t have to be, omnipresent. This is impossible. The team itself is much more than a single person, and this makes the overlooking capacity to be multiplied, for instance, allowing the team to spot any risks that would be less identifiable individually. In that sense, it’s also very positive that each one in the team feels free to point out these risks, without fear. I think that this servitude of the leader, in that sense, helps a lot.

***Participant 4*:** Trying to sum up a bit, I think these characteristics and behaviours are well connected with what Participant 3 just said about dealing with people. If the leader doesn’t know how to deal with people, if he does not understand that people make mistakes and this is natural and healthy in an agile environment, the [Agile] methodology will not work. The leader needs not only to understand that, but to know the people he works with. This is essential not only in Agile, but mainly in Agile.

***Participant 2*:** About dealing with people, I think the dynamics issue, when you are able to "read" people’s behaviours, this helps a lot in managing the team. In addition to what my colleagues have said, knowing how to delegate, to listen, and to communicate clearly is key to me. It’s often difficult for us to externalize what is in our mind, so, it’s important to look for a common language to communicate. Multidisciplinary teams are made up of people with different beliefs, values, and experiences, and because of that, they can understand the same thing in different ways. So, it’s important for the leader to know how to communicate clearly the themes.

***Participant 1*:** Just to complement, I think almost all aspects and characteristics have already been mentioned here. When it comes to the leaders, it should already be clear that among its roles and responsibilities lies the fact that he must lead people. And that's a lot of being there, as was said here. When you have, for example, a Product Owner or a Scrum Master in the team, the leader needs to empower them to make decisions. I think in this case the delivery ends up happening more effectively. The leader assigns the responsibilities to the team and lets people follow their way; then they gain more knowledge and more confidence. So, I think the leader has to let people work and then things will flow in a smoother way.

***Participant 5*:** We have already talked a lot about the leader's vision in an Agile environment, about how the leader should behave and such things. Some of the mentioned characteristics are not only applied in Agile approaches since they are characteristics that every leader should have and show on a daily basis. What I wanted to say is that I think, in Agile, the leader must be willing to face changes frequently because, as we are discussing now, he should be frequently involved in consolidating the methodology [Agile] in the organisation. Agile doesn't have a prescription to be followed, a map where you say, "take this and follow it". So, thinking about it, I believe the leadership has to keep in mind that it's important to establish a continuous improvement process, always questioning what can be done to extend the benefits already obtained by the organisation and the project as well. It's also essential to know that more changes will come, and then deal with these changes naturally.

Question 3/5

Luis Mattos**:** Regarding the aspects of the ORGANISATION, more specifically within the TEAM, what are the characteristics and behaviours that you think the project team should have / present to work with an Agile methodology?

Participant 2**:** I think the team needs to be experienced; it must constantly seek self-improvement. In communicating it should always try to understand what other Agile teams within and outside the organisation are doing, and more, to understand that teams have to be transcendent. An Agile team has to be able to start a job and know how to finish it. The team needs to be aware of its responsibility. From the moment they [members of the team] feel they are part of something bigger, the work happens more fluidly. Each member is pulling the other, especially when they see that someone is messing up, getting off the ground, and they have to communicate with each other without the need for a facilitator. Sometimes it may be necessary for someone to facilitate this communication, but it's important that, in a general way, it flows without the need of that person. So, I think communication should be clear. As a characteristic, I think teams need to show diverse skills to be able to work in a transcendent way. The team must be autonomous, but it must also seek this transcendence. And the leader, he needs to let the team stand out. The good leader does not appear. He does his job, but he is barely seen so that the team might be the protagonist. The people who are part of the team need to seek this, people have to pursue the cycle of accountability.

***Participant 3*:** I think leadership plays a key role in assembling the Agile team, but the team member needs to show one important feature, that is being proactive. He [the team member] needs to know that he can and should express himself freely when he feels it's necessary. Another important aspect is the team member to know what means teamwork because very often the decision he will make will impact the work of a colleague on his side. So, the decision should not be egocentric because the risk of that decision can impact not only him but the whole project.

***Participant 5*:** What you're saying is not only important in Agile approaches. In my opinion, even if this is not an Agile environment, each professional who integrates a project team must be responsible needs to behave as a team player and fulfil what has been agreed on deliveries. In my opinion, it's not only Agile approaches that require these behaviours.

***Participant 2*:** That's true, Participant 5, it's not. What happens is that the Agile way of thinking and acting exalts these behaviours. In Agile, these behaviours become much more evident because they are stimulated and are an essential part of the methodology mindset.

***Participant 3*:** Yes, there are teams that behave like this, that presents these characteristics without even knowing that Agile methodologies exist. In Agile, we exalt these characteristics and stimulate them in a conscious and purposeful way. But there is also, and always will be, that classic developer who sits in his chair, put on earphones and forgets the planet while coding. At six o'clock sharp, he stands up and leaves. But there is also the other type of developer who is genuinely interested in knowing the importance, the value of what he is developing, what value will bring to the customer that functionality he's been working on. I think to work in an Agile environment, the developer that I called the "classic" one won't prevail, because he will have to put off his earphones and collaborate more and offer help to his colleagues, and if he does not do that, either he will leave or will be segregated by the team.

***Participant 5*:** You raised a point that I fully agree. However, the issues involving accountability, responsibility, communication, they are okay; all of these points favour the use of Agile, but not exclusively Agile.

Participant 3**:** Yes, that's true, but as I said, Agile helps these behaviours to manifest themselves. I think that without the stimulus, without Agile's emphasis on these behaviours and characteristics, the chances of them spontaneously raising would be much lower. Agile encourages these attitudes and characteristics in project teams.

Participant 1**:** I don’t think there’s much more to go into here. Once again, I think we are in the realm of people. If a team member is not collaborative, if he is not transparent, if he is not willing to share information, that is, if he is not able to adapt to this new context of teamwork, the whole thing will fail because even if you have the best methodology, the best processes, in the end, will be the people who will be responsible for making things happen.

<u>Question 4/5</u>

Luis Mattos**:** Regarding the aspects of the PROJECT, more specifically in the ENVIRONMENT, what are the environmental factors for the project that you consider to be more characteristic of initiatives eligible to use an Agile methodology?

Participant 3**:** I think that in an organisation that has a very hierarchical structure, such as the military, this type of methodology [Agile] would be among the least probable options, though it’s well known that the armed forces, in general, apply and encourage the use of multidisciplinary teams. They also believe in specialism, yes, but even the specialists are trained to perform other types of tasks very well. I believe that for you to work well with Agile methods, it’s important to have a more flexible hierarchy, allowing freely communication and collaboration between the members of the team and that this openness is not considered a breach of a rule or internal policies. So, I think when an organisation has a very hierarchical structure, it does not favour the use of Agile.

Participant 1**:** I think the issue of hierarchy actually makes a lot of difference. We already have talked about the issue of leadership, so if you use it negatively, for example, if every time you need to make a decision it needs to be made exclusively by the leadership, which is quite normal in large companies, I think it doesn't favour the use of Agile at all. More specifically in the case of software projects with a very volatile scope, the high volume of changes is another environmental factor that favours the use of Agile, but I still think that the physical structuring of the team and the governance of the project are the main environmental aspects that favour the use of an Agile methodology, and the leaders of the organisation play a key role in making these things happen.

Participant 5**:** I will reinforce this point of the colocation of the project team. The fact that people are working closely on the same project allows the dissemination of the information in a much more assertive and faster way. The necessary adjustments take place in an easier way and also allow the team to mitigate risks. Often a company faces difficulties whereas working with project teams in silos, which doesn't favour a good exchange of information. I can cite as an example an experience that we had here in the company, where a customer area triggered a first team that was responsible for making one, so to say, screening of the request. This area that assessed the request then identified other possible areas that should act in that request, and it triggered a second or a third area. Only the initial communication was made between the client and the first area, and when there was a need for a change of scope, reprioritisation, deprivation or even cancellation of the request, communication often failed and didn't reach the other areas involved sequentially. So, this created many problems with regard to scope changes, because the request didn't navigate through the flow properly due to the failures in the communication. There was another case where the team was working for a week on a topic that had already been cancelled. So, I do agree with the importance of having small teams so that communication happens more fluidly.

Participant 2**:** Of all what has been said so far about the environment issue, at least what I've been able to grasp, reinforces that it is quite important to have the less bureaucratic environment as possible. Putting together all that has been said, in fact, instead of having a vertical view, in a top-down direction, it is important that we have a horizontal view, in which leadership and project teams are perceived by the rest of the organisation as a single body working with clear and aligned goals among everyone, and communication is widespread and in a mature level.

Participant 4**:** A classic expression that is commonly used in a company is when someone says, "We are all in the same boat." If people do not know the direction the boat is heading nor their roles in the boat, whether the boat sinks or many people will abandon it in the middle of the trip.

Participant 2**:** This example is excellent.

Participant 1**:** Then, you have the turnover issue, the motivational issue...

Participant 5**:** I think it’s essential that the company has enough maturity to work with an Agile methodology; otherwise, it will have problems. And why? Firstly because you may have all the tools and information you need to deploy and use an Agile framework in your company, but if you don’t have the vision, the culture, the way of thinking that is very aligned with Agile values and principles, it’s very likely that the company will not reach good results. If the company doesn’t have an Agile mindset, the difficulties will be much greater because before it has that whole question of, let's say, curve-shift thinking. I do not say a learning curve, because if the company has competent professionals, this may happen in a natural and structured way, but Agile really demands a change in the way of seeing and doing things, and for that to happen, the company needs to be open to this mindset changing. And this for me is all about the maturity of the company.

Participant 3**:** I wanted to complement with two things. The first is that I don’t think there is a prescription to success working with Agile. The methodology [Agile] can work well with one team, but maybe cannot with another one with similar profile in the same organisation. The other thing would be, when submitting a group of people to a new work process, such as using an Agile development process, it’s natural that there’s a reaction. This reaction may come from the adaptation time that the target group needs to assimilate and begin to work in a new way, or it may be that the whole thing has already gone astray, that the process of change is going in a direction that is not correct because people are reacting to it. For me, these two points are related to what we discussed about the maturity of the company, because if you are not aware of it, the first reaction is to think like this, and then I'm referring the first aspect, the problem lays on the team where Agile is not working and not on something that has not been done properly at the moment of the deployment. And in the second point, you need to have someone monitoring the deployment process that is able to see if the reaction to the change comes from the natural and inherent difficulty in dealing with changes, or comes from people's resisting to the change. So, for me, these are two important things to pay attention to the environment issue to work with Agile.

Participant 2**:** Exactly. This point you just raised is important. In Agile there is no such thing as not documenting, of having no control over what is being done in the project or also of not having version control. The point is that in traditional methodologies these things happen on a command and control way, and today it is still like this in many organisations. People need a schedule to follow. The timeline is part of a predictive model and we know well that software development is an empirical process. Changes will occur throughout the project. To have a baseline to follow and a deadline is okay, it's part of the plan, except that the necessary adjustment period to see if we're on the right track is part of the setup process where you'll actually find out whether Agile will fit, if we will find the ideal cadence, if we will be able to determine our ability to deliver. And different teams within a single organisation may have different numbers for each of these indicators. But concluding what I wanted to say, in Agile we're not going to stop doing things like documentation, controlling and versioning, we're going to do all of this, but only the amount is really necessary. Some people have a distorted view of these Agile concepts, and I do agree that it’s critical to have people experienced in implementing and consolidating the Agile methodology, especially when the organisation's level of maturity is low.

Participant 4**:** And there’s also the matter of the costs involved in such initiative [Agile implementation]. Depending on the organisation and the scope, of course, although this will only influence whether more or less money will be lost. But the point here is that it still involves an investment that if it’s not correctly done, the organisation will lose money.

***Participant 3*:** There are some maturity models that help measure the ability’s degree of the company to deploy and absorb a new methodology such as Agile. I don’t go deep here mentioning names nor delving into the work necessary, but in a very brief way, the limit lays on the maturity of the human beings, the maturity of the people who make the decisions in the company. So, if their maturity is low, what are the chances that the implementation works out? Very low, in my opinion. So, part of this environmental issue is related to the maturity level of the organisation to handle new processes implementations.

***Participant 1*:** There is a phrase I like very much to apply in everyday life, and that has a lot to do with the cultural issue in organisations, which is, "Stop starting and start finishing".

<u>Question 5/5</u>

***Luis Mattos*:** Regarding the aspect of the PROJECT, more specifically in the PRODUCT scope, what are the characteristics related to the product of the project you think are most relevant in determining an Agile approach as a development methodology?

***Participant 2*:** From my experience, I would say that Agile is very closely linked to a product vision, product lifecycle, continuous product improvement. There is even a widely used expression that says, "The usage defines the product." I'm going to use an example of this banking application. There is a TV advertisement about this app where a person who wants to pay an account says, "Gosh. It sucks having to type so many digits only to pay a bill by phone." And then another person says, "Did you know that the [bank name] application can read the data from your bill? You don’t have to type anything." It’s this vision where I will deliver value to the customer, I will deliver what he needs to solve these simple everyday things, such as paying a bill by cell phone, this is proper of a "product-prone" approach. This matches what Participant 3 said here in the first question, when he said that the product of an Agile project has to deliver value constantly. So, if a company is going to deliver a product as a result of a project, but that product is not expected to go or evolve after it's released, then, in this case, I think an Agile approach may not be necessary. Can it be used? Yes, it can. But I think Agile offers more advantages when you think of a long-term product that will evolve over time.

Participant 3: This example related to the banking application is a good example of constant value delivery because this bank is always seeking to add new functionalities aiming to make the life of the client easier. The application of this bank even offers a whole package of functionalities for people with impaired vision. This is value for blind customers. I think another characteristic related to the product is not offering features that will not be used. Functions that are not useful do not generate value. This also counts against another key characteristic related to the product in Agile methods, which is the simplicity of the solution. And simplicity is closely linked to quality because unimplemented code is code that doesn’t generate errors.

Participant 4: This issue regarding the product offering simplicity and also just enough features is linked to the concept of MVP, which is another Agile practice. How modular a product can be and how fast I can deliver value to my customer, only what is really needed. I can’t spend my time developing something that will not be used. Another point is that delivering an MVP helps in gathering early feedback and improving the system, and it adds more value to the customer in later deliveries.

Participant 1: I agree with this statement about the MVP because it has everything to do with the issue since it’s a characteristic of products resulting from Agile approaches. But I wanted to add a point. Regarding the product, when using an Agile methodology, the ideal is that the product can be sliced in order to allow frequent deliveries. If you deliver a system at different times, the first delivery as an MVP, this MVP goes to the market and allows you to collect feedback from the market. This gives you much more input to evolve your product and deliver something even better in later stages.

Participant 2**:** I think a point that is related to the product is where you should focus to find the best solution, the ideal product. I believe Agile has a characteristic that is not unique, but that is more often observed in organisations that apply Agile, which is the fact that the team focuses on the problem to be solved by the product and not in the product itself. The customer may ask you to develop a car, but if you don't ask for what purpose he wants this car, you risk giving him a Ferrari, when what he really needed is an SUV. Thus, this approach focusing on solving the problem relates to the product issue. Not that related to the characteristics of the product itself, but with the approach to its development. I think what Participant 1 said about the product modularization and what Participant 3 said about delivery value are things that are connected. The principle of a modularized product that begins to generate value as early as possible is to focus on simplicity and solve the problem it was supposed to solve. That is, the focus when designing and developing the product has to be in the problem to be solved and not in the solution.

Part II

Requirements gathering of the information system that will be designed to implement the decision model derived from the analysis of the aspects discussed in Part I. The objective is to collect the opinion of the participants about the ESSENTIAL features for this information system.

Question 1/1

Luis Mattos**:** An information system will be developed, aiming to implement the decision-making model resulted from the analysis of the aspects discussed in the first part of this focus group. The main objective of the system is to indicate if the project is eligible for an Agile approach or not. In this context, what would be, in your opinion, the ESSENTIAL functionalities of the system?

Participant 2**:** Thinking about a system like that, the first thing I would say is that it would need a sort of data gathering feature and would have the rules and metrics to point out the best approach previously registered, like a diagnosis system.

Participant 1**:** I think the whole dynamic would be based on questions and answers to produce a diagnosis that allows decision making. So, I imagine a system where I'd have a questionnaire. Where I'd have some questions, and from the moment the user answers these questions there will be an algorithm where the system will calculate everything and will generate the result, indicating whether an Agile approach is suited or not. Basically, I think the main functionality of this system would be something like this.

Participant 3**:** I understand that the questions must be objective, and some of them should have different weights. As the evaluation goes on, it will reach the point where the result may be very close, and in this case, I think the questions need to be more specific in order to differentiate the methodologies. I think these questions have to add in favour of one methodology or another, and they shouldn't point out exclusively to a single option. I think the answer has to lead to a result more prone to a good probability than a complete certainty.

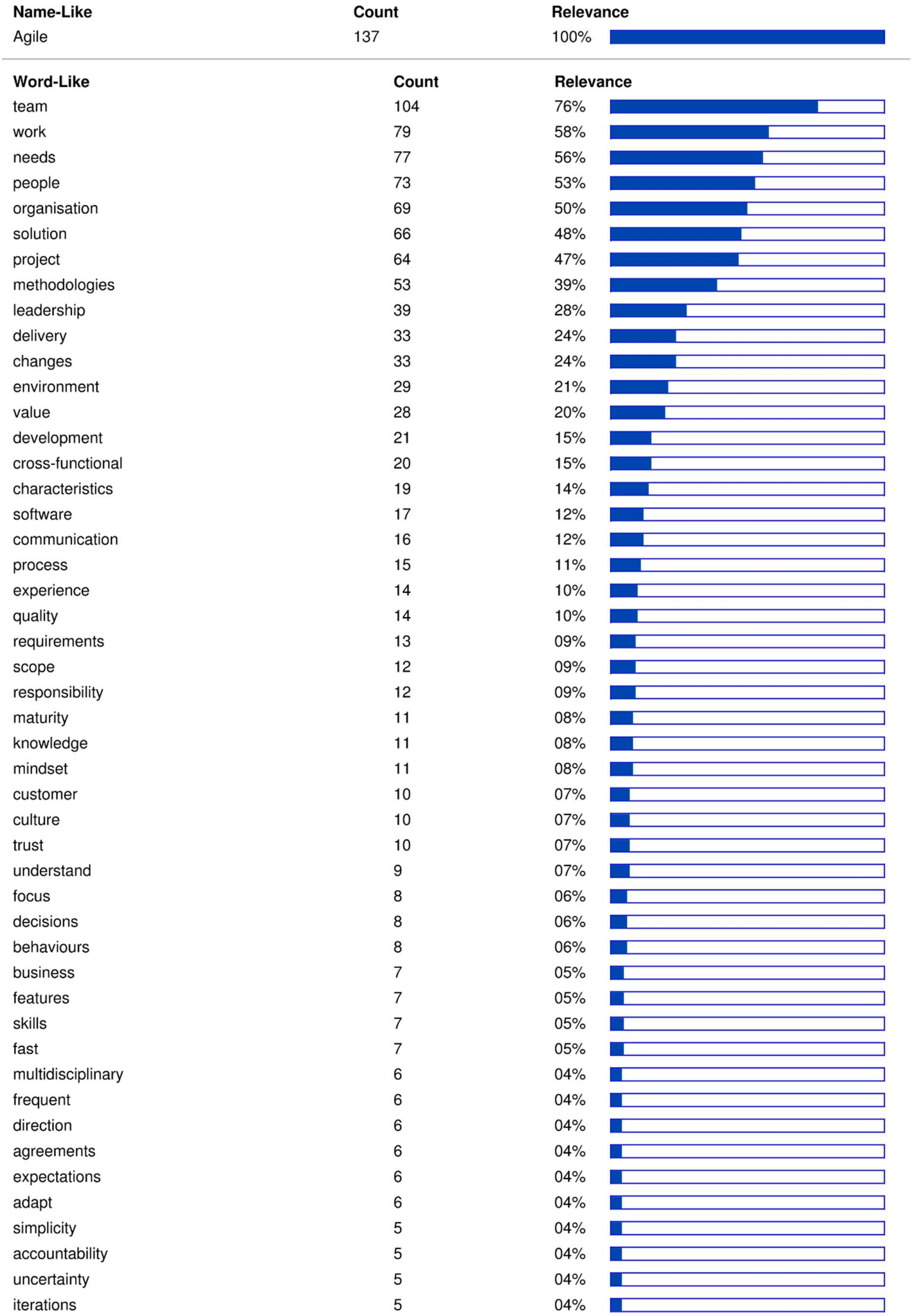
Participant 1**:** To summarize, I think that the functional requirements would be something in line with a system that should allow for the upload of a questionnaire template, containing weighted questions and different types of answers. And then you must have a logic behind doing all the calculations depending on what was answered for each question. Binary responses types, yes or no, and also with scales, so you can capture, for example, how often a given event occurs. Each response will have a different weight that will direct you to one or another way.

Participant 2**:** I think the questions would have to be separated by the respondent's profile. I initially see two profiles, one at the management level that may include issues that are more focused on the project strategy and the organisation itself, and a more technical one, where the executing layer of the project will be more prepared to answer.

***Participant 1*:** I think this division could be done by dimensions, where each one would have its weight. It’s obvious that in this case, the system would be more elaborated, but just as an example, it would be something like a Survey Monkey, where you can create questions, and there would be a decision algorithm to calculate and point out the result.

--- // ---

Appendix J – Leximancer Ranked Concept List - Aspects Related to the Choice of an Agile Approach



Name-Like	Count	Relevance
Agile	137	100%

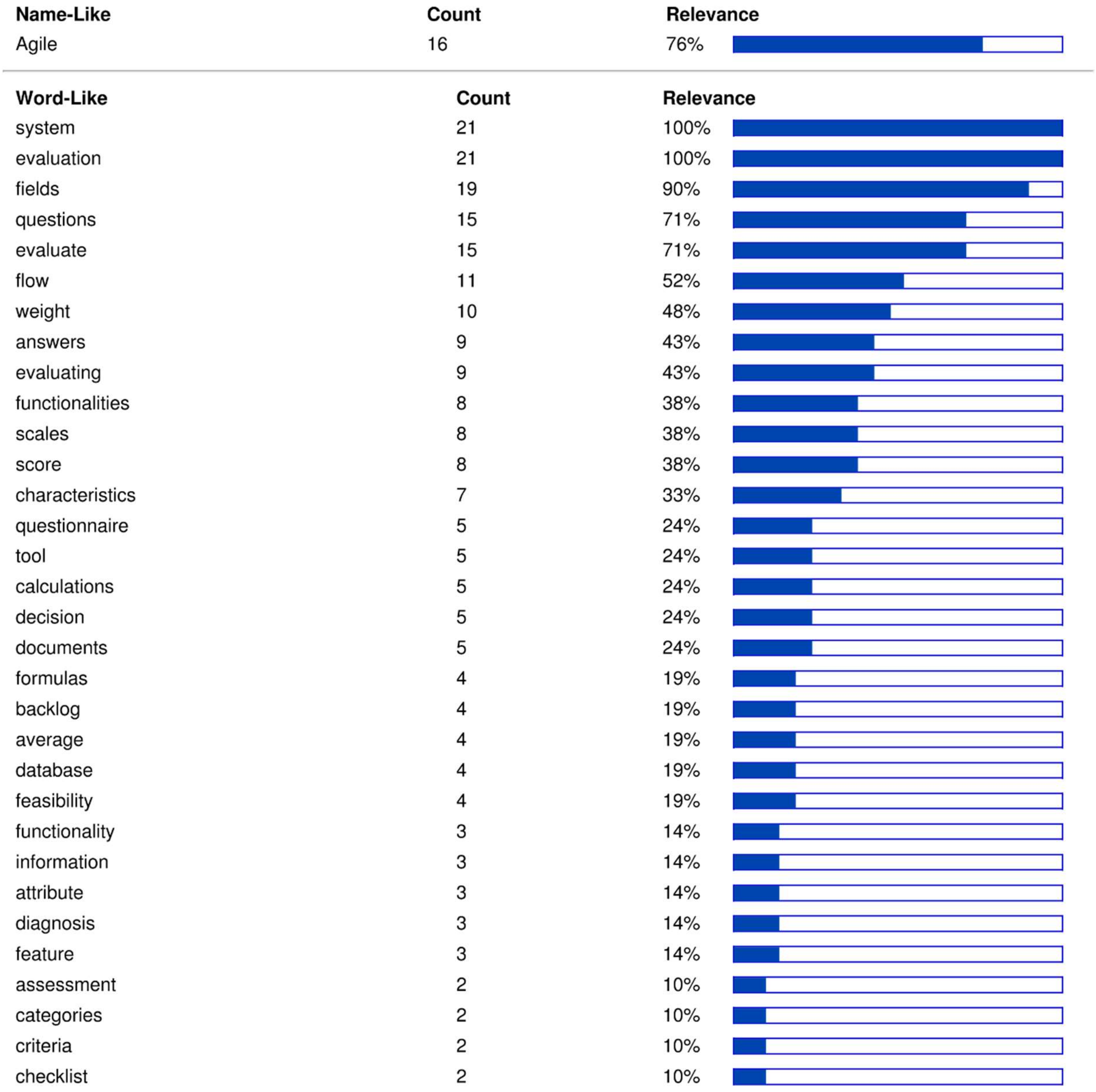
Word-Like	Count	Relevance
team	104	76%
work	79	58%
needs	77	56%
people	73	53%
organisation	69	50%
solution	66	48%
project	64	47%
methodologies	53	39%
leadership	39	28%
delivery	33	24%
changes	33	24%
environment	29	21%
value	28	20%
development	21	15%
cross-functional	20	15%
characteristics	19	14%
software	17	12%
communication	16	12%
process	15	11%
experience	14	10%
quality	14	10%
requirements	13	09%
scope	12	09%
responsibility	12	09%
maturity	11	08%
knowledge	11	08%
mindset	11	08%
customer	10	07%
culture	10	07%
trust	10	07%
understand	9	07%
focus	8	06%
decisions	8	06%
behaviours	8	06%
business	7	05%
features	7	05%
skills	7	05%
fast	7	05%
multidisciplinary	6	04%
frequent	6	04%
direction	6	04%
agreements	6	04%
expectations	6	04%
adapt	6	04%
simplicity	5	04%
accountability	5	04%
uncertainty	5	04%
iterations	5	04%

Appendix K – Leximancer Ranked Concept List - Aspects Related to Choosing an Agile Methodology

Name-Like	Count	Relevance
Scrum	25	62%
Kanban	19	48%
Extreme Programming	12	30%
Lean Software Development	2	05%

Word-Like	Count	Relevance
bottlenecks	25	62%
stories	21	52%
visibility	19	48%
software	16	40%
artefacts	15	38%
ceremonies	15	38%
engagement	13	32%
organized	13	32%
work	12	30%
delivery	11	28%
incrementally	11	28%
progress	11	28%
feedback	11	28%
business	9	22%
improvement	9	22%
capacity	9	22%
communicating	9	22%
innovation	8	20%
support	8	20%
engineering	8	20%
continuous	7	18%
integration	7	18%
management	7	18%
changes	7	18%
programming	7	18%
value	6	15%
interaction	6	15%
fast	6	15%
adaptability	6	15%
ownership	6	15%
focus	6	15%
early	5	12%
control	5	12%
flexibility	5	12%
methodologies	4	10%
customer	4	10%
techniques	4	10%
iterations	4	10%
design	4	10%
tools	4	10%
involvement	3	08%
quality	3	08%
user	3	08%
development	3	08%
optimization	3	08%

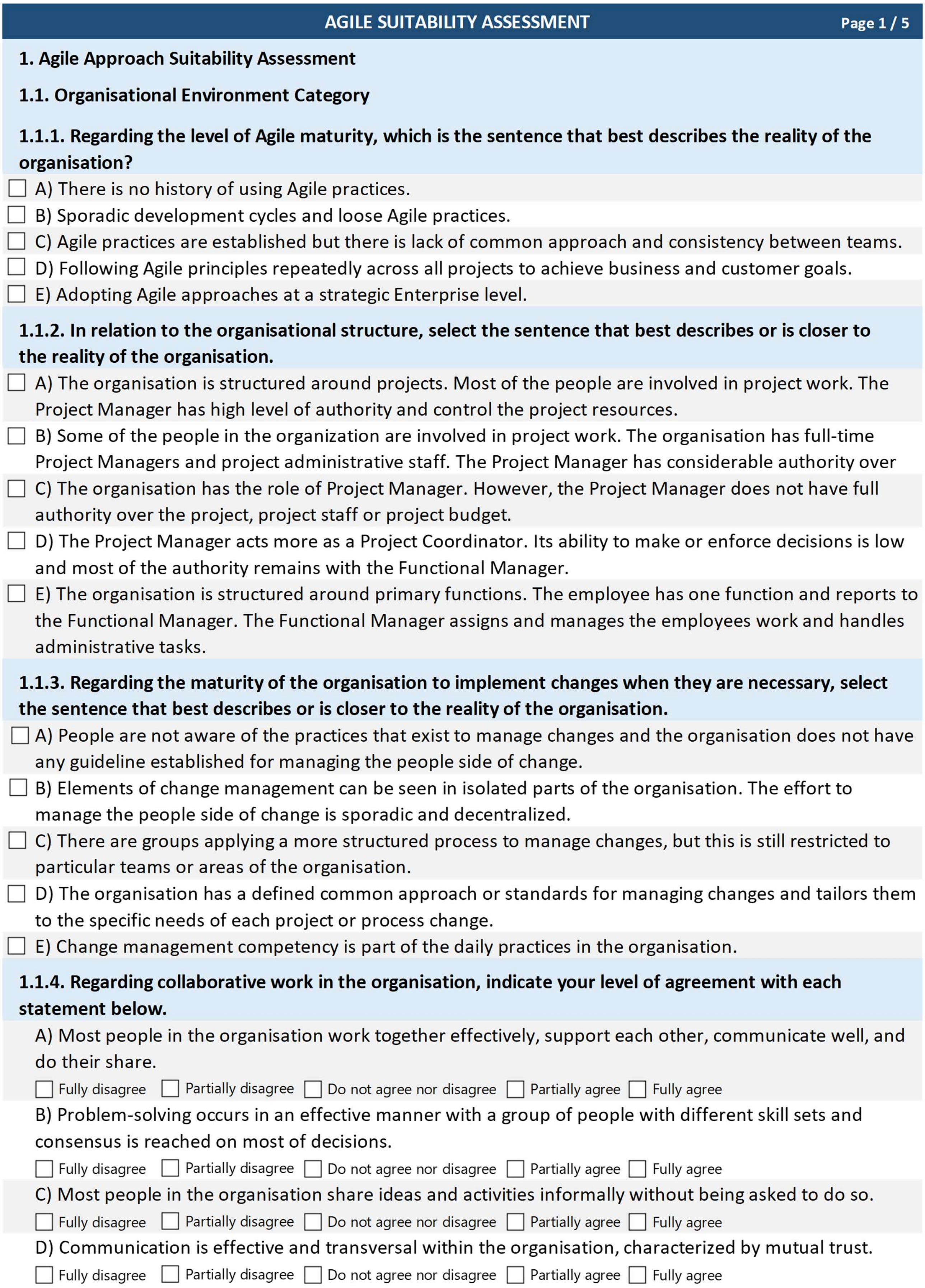
Appendix L – Leximancer Ranked Concept List - System Requirements



Name-Like	Count	Relevance
Agile	16	76%

Word-Like	Count	Relevance
system	21	100%
evaluation	21	100%
fields	19	90%
questions	15	71%
evaluate	15	71%
flow	11	52%
weight	10	48%
answers	9	43%
evaluating	9	43%
functionalities	8	38%
scales	8	38%
score	8	38%
characteristics	7	33%
questionnaire	5	24%
tool	5	24%
calculations	5	24%
decision	5	24%
documents	5	24%
formulas	4	19%
backlog	4	19%
average	4	19%
database	4	19%
feasibility	4	19%
functionality	3	14%
information	3	14%
attribute	3	14%
diagnosis	3	14%
feature	3	14%
assessment	2	10%
categories	2	10%
criteria	2	10%
checklist	2	10%

Appendix M – Agile Suitability Assessment Form (MS Excel)



AGILE SUITABILITY ASSESSMENT Page 1 / 5

1. Agile Approach Suitability Assessment

1.1. Organisational Environment Category

1.1.1. Regarding the level of Agile maturity, which is the sentence that best describes the reality of the organisation?

☐ A) There is no history of using Agile practices.

☐ B) Sporadic development cycles and loose Agile practices.

☐ C) Agile practices are established but there is lack of common approach and consistency between teams.

☐ D) Following Agile principles repeatedly across all projects to achieve business and customer goals.

☐ E) Adopting Agile approaches at a strategic Enterprise level.

1.1.2. In relation to the organisational structure, select the sentence that best describes or is closer to the reality of the organisation.

☐ A) The organisation is structured around projects. Most of the people are involved in project work. The Project Manager has high level of authority and control the project resources.

☐ B) Some of the people in the organization are involved in project work. The organisation has full-time Project Managers and project administrative staff. The Project Manager has considerable authority over

☐ C) The organisation has the role of Project Manager. However, the Project Manager does not have full authority over the project, project staff or project budget.

☐ D) The Project Manager acts more as a Project Coordinator. Its ability to make or enforce decisions is low and most of the authority remains with the Functional Manager.

☐ E) The organisation is structured around primary functions. The employee has one function and reports to the Functional Manager. The Functional Manager assigns and manages the employees work and handles administrative tasks.

1.1.3. Regarding the maturity of the organisation to implement changes when they are necessary, select the sentence that best describes or is closer to the reality of the organisation.

☐ A) People are not aware of the practices that exist to manage changes and the organisation does not have any guideline established for managing the people side of change.

☐ B) Elements of change management can be seen in isolated parts of the organisation. The effort to manage the people side of change is sporadic and decentralized.

☐ C) There are groups applying a more structured process to manage changes, but this is still restricted to particular teams or areas of the organisation.

☐ D) The organisation has a defined common approach or standards for managing changes and tailors them to the specific needs of each project or process change.

☐ E) Change management competency is part of the daily practices in the organisation.

1.1.4. Regarding collaborative work in the organisation, indicate your level of agreement with each statement below.

A) Most people in the organisation work together effectively, support each other, communicate well, and do their share.

☐ Fully disagree ☐ Partially disagree ☐ Do not agree nor disagree ☐ Partially agree ☐ Fully agree

B) Problem-solving occurs in an effective manner with a group of people with different skill sets and consensus is reached on most of decisions.

☐ Fully disagree ☐ Partially disagree ☐ Do not agree nor disagree ☐ Partially agree ☐ Fully agree

C) Most people in the organisation share ideas and activities informally without being asked to do so.

☐ Fully disagree ☐ Partially disagree ☐ Do not agree nor disagree ☐ Partially agree ☐ Fully agree

D) Communication is effective and transversal within the organisation, characterized by mutual trust.

☐ Fully disagree ☐ Partially disagree ☐ Do not agree nor disagree ☐ Partially agree ☐ Fully agree

AGILE SUITABILITY ASSESSMENT **Page 2 / 5**

1.2. People Category

1.2.1. Regarding the leadership characteristics and behaviours within the organisation or at the level of the projects, indicate your level of agreement with each statement below.

A) The leadership of the organization or the project encourages and supports the use of Agile approaches in software development.

□ Fully disagree □ Partially disagree □ Do not agree nor disagree □ Partially agree □ Fully agree

B) The leadership has confidence that the project team can transform the customer needs into a successful product, trusting the team has the best people to get the job done.

□ Fully disagree □ Partially disagree □ Do not agree nor disagree □ Partially agree □ Fully agree

C) The leadership delegates and empowers the team so that they can make their own decisions, letting the team stand out and acting more as a coach and facilitator.

□ Fully disagree □ Partially disagree □ Do not agree nor disagree □ Partially agree □ Fully agree

D) The leadership understands that Agile means experimentation and that it may lead to mistakes. They believe that these mistakes can be healthy to the project and that they should occur as soon as possible so the project team can anticipate issues and act on them.

□ Fully disagree □ Partially disagree □ Do not agree nor disagree □ Partially agree □ Fully agree

E) The leadership incentives and promotes cross-functional and multidisciplinary teams.

□ Fully disagree □ Partially disagree □ Do not agree nor disagree □ Partially agree □ Fully agree

F) The leadership knows very well your teams and stay close to them to be familiar with important themes and act to unblock issues as quickly as possible.

□ Fully disagree □ Partially disagree □ Do not agree nor disagree □ Partially agree □ Fully agree

G) The leadership is versatile at the decision-making level, being able to justify and support the reasons for changes and transitions.

□ Fully disagree □ Partially disagree □ Do not agree nor disagree □ Partially agree □ Fully agree

H) The leadership shields the team from external interference as much as possible.

□ Fully disagree □ Partially disagree □ Do not agree nor disagree □ Partially agree □ Fully agree

I) The leadership is willing to face frequent changes in the course of the project, knowing that more changes will come, and then dealing with them naturally.

□ Fully disagree □ Partially disagree □ Do not agree nor disagree □ Partially agree □ Fully agree

J) The leadership promotes a continuous improvement environment, often questioning what can be done to extend the benefits already obtained with new ways of work.

□ Fully disagree □ Partially disagree □ Do not agree nor disagree □ Partially agree □ Fully agree

1.2.2. Regarding the project team characteristics, behaviours and competences, indicate your level of agreement with each statement below.

A) The project team is experienced in Agile and constantly seek for self-improvement. They seek to know how other Agile teams (within or outside the organisation) have been working.

□ Fully disagree □ Partially disagree □ Do not agree nor disagree □ Partially agree □ Fully agree

B) The project team has the autonomy to make their own decisions about how to undertake the project work (self-organizing team).

□ Fully disagree □ Partially disagree □ Do not agree nor disagree □ Partially agree □ Fully agree

C) The project team has all competencies needed to accomplish the work without depending on others not part of the team (cross-functional team).

□ Fully disagree □ Partially disagree □ Do not agree nor disagree □ Partially agree □ Fully agree

D) The project team has frequent access to at least one business / customer representative in order to ask questions and get feedback on the work.

□ Fully disagree □ Partially disagree □ Do not agree nor disagree □ Partially agree □ Fully agree

AGILE SUITABILITY ASSESSMENT **Page 3 / 5**

E) The alternative that includes the size of the team that will undertake the project is (if not sure, choose the most likely option).

□ Up to 12 □ 13 to 20 □ 21 to 45 □ 46 to 80 □ More than 80

F) The team member is committed, engaged and focused. He/she avoids (as much as possible) context switch.

□ Fully disagree □ Partially disagree □ Do not agree nor disagree □ Partially agree □ Fully agree

G) The team member is accountable, responsible, transparent, and willing to share information.

□ Fully disagree □ Partially disagree □ Do not agree nor disagree □ Partially agree □ Fully agree

H) The team member shows team spirit, has sense of unity, seeks cohesion, and has good behaviour.

□ Fully disagree □ Partially disagree □ Do not agree nor disagree □ Partially agree □ Fully agree

I) The team member feels free to raise problems and risks without fear.

□ Fully disagree □ Partially disagree □ Do not agree nor disagree □ Partially agree □ Fully agree

J) The team member has an open mindset and quickly adapt to new work environments.

□ Fully disagree □ Partially disagree □ Do not agree nor disagree □ Partially agree □ Fully agree

1.3. Project Environment Category

1.3.1. Regarding the level of Project Management capabilities, which is the sentence that best describes the reality of the project?

□ A) There is no formal implementation of Project Management. The processes to be used and the effectiveness of the results will come from the experience and expertise of individuals and the team.

□ B) There are some Project Management capabilities defined and used at an organisational level, but they are incomplete or inconsistently applied. The project will have some level of formality, but not comprehensive nor fully applied.

□ C) There is a comprehensive Project Management process in place and consistently applied to all projects within the organisation.

□ D) There is a consistently defined Project Management process fully integrated into the management capabilities of the organisation and its lifecycle.

□ E) There is a comprehensive, fully integrated approach to manage projects, which exists within an ongoing cycle of continuous improvement.

1.3.2. For each of the aspects related to the project under evaluation listed below, select the answer that best represents or is closer to the reality of the project.

A) What is the degree of uncertainty on the system requirements before the project start or in the initial stages of the project?

□ Up to 5% □ 6% to 15% □ 16% to 30% □ 31% to 45% □ More than 45%

B) Is there a real necessity for collocating the project team and business / customer representatives in order to undertake this project?

□ Not at all □ It is indifferent □ It would be nice □ It would be important □ It is mandatory

C) Is there a need for the project to use tools (digital or not) and techniques in order to enhance the communication and collaboration between the project team and stakeholders?

□ Not at all □ It is indifferent □ It would be nice □ It would be important □ It is mandatory

1.3.3. For each of the aspects related to the product resulting from the project under evaluation, select the answer that best represents or is closer to the reality of the project.

A) Regarding the possible impact of defects presented by the software resulting from the project, what are the most likely related losses?

□ Time or business opportunities □ Discretionary funds □ Essential funds □ Single life □ Many lives

B) Can the product of the project be built and delivered in smaller chunks in different moments during the project?

□ Yes □ Very likely □ It is indifferent □ Very unlikely □ No

AGILE SUITABILITY ASSESSMENT Page 4 / 5

2. Agile Methodology Suitability Assessment

2.1. Based on the known information about the software development project, select the alternative that best reflects or is closer to the emphasis that the Agile methodology to be applied must have so that it meets the needs and objectives of the project.

□ A) The project requires mainly a managerial framework. The focus is on productivity, ensuring the software is built quickly and with minimum complexities. An iterative and incremental approach is required in order to periodically release software increments.

□ B) The project needs a framework whose priority is the high quality of the interim and final software. In order to have this, the methodology must focus on applying sophisticated software engineering practices and work with short development cycles.

□ C) The project requires a methodology with focus on improving the workflow and refraining from producing a surplus throughout full transparency of work and real time communication of capacity. It is key to have a clear view of the development process and address issues as quick as possible.

□ D) The project needs a framework whose focus is on streamline the development process, cutting away all activities that do not directly contribute to the final software. The methodology does not necessarily have to work with fixed-length development cycles, but relies on an experienced team to apply best software engineering practices in order to deliver the project.

2.2. Based on the known requirements for this project and on what is known about the stakeholders as well, select the alternative that best reflects or is closer to the expected reality of the project in respect to the frequency the system requirements may change.

□ A) Changes in requirements are expected, but the project approach will not embrace these changes once the planning for the development cycle (iteration) is done nor during an ongoing cycle. If any big change is needed, the project team typically stops the current cycle and starts a new one with new requirements.

□ B) Changes in requirements are expected, but while working with stable fixed-length development cycles (iterations), embrace these changes during cycles will be accepted when strictly necessary, with the project maintaining the planned workload throughout the reprioritization of features.

□ C) Frequent changes in requirements and project scope are the norm. These changes will typically be accepted, implemented and delivered as fast as possible since one of the project's priority is to deliver value to the customer as soon as possible.

□ D) Frequent changes in requirements are expected. These changes will be accommodated in the product backlog according to the priority and will be executed as soon as there is available capacity in the team.

2.3. Based on the known systems requirements for the project and on the general project objectives, select the alternative that best matches the context of the project regarding the delivery approach.

□ A) The project requires an approach that enables frequent delivery of parts of the software in a planned manner. The scope of the releases will mostly contain business features defined by the client. There is flexibility for changing requirements, but the negotiation for the implementation of such changes must occur in order to maintain the balance of the product backlog, minimizing risks for the continuity of the delivery and the quality of the software.

□ B) The project requires an approach that enables the delivery of parts of the software as fast as possible. There must be high flexibility for the implementation and rapid availability of changes. For this, it is essential to have an optimized and constantly revised workflow, minimizing risks of creating bottlenecks or performing tasks that do not directly contribute to the software.

AGILE SUITABILITY ASSESSMENT **Page 5 / 5**

2.4. Based on the needs and objectives of the project and on what is known about the stakeholders as well, select the alternative that reflects or more closely matches what is required or expected in terms of customers or business representatives engagement in the project.

□ A) The customer or business representatives will be actively engaged in the project in a daily basis, integrating the project team.

□ B) The customer or business representatives will interact frequently with the project team and participate in important project activities (like iteration planning and reviews), providing frequent feedback on software increments.

□ C) The customer or business representatives will be mainly involved in the project to identify value streams and to contribute to the optimization of the development process.

2.5. Based on the project needs and objectives and also on what is known about requirements, select the alternative that represents or more closely matches what is expected regarding how business requirements, system features, or project activities should be prioritized in the project.

□ A) The customer or business representative will define the priority of the features within a development cycle, but the project team will perform the work in the sequence they deem to be most adequate for the objectives set for the cycle.

□ B) The project team will work on the planned features for the development cycle in the order defined by the customer or business representative. The project team will not be allowed to modify the order of the

□ C) The prioritization of the job will be done by whoever is in charge of it, in accordance to the project needs. The main concern is pulling work and often reprioritize remaining tasks in order to keep a paced

□ D) The prioritization of the work will be done by whoever is in charge of it, according to the needs of the project. The main concern is to avoid unnecessary work and prioritize tasks that directly add value to the software.

Appendix N – Equations of the Agile Suitability Assessment

- **Organisational Environment category equation**

$$\text{Organisational Environment Score} = (\text{Q01 Points} \times 30\%) + (\text{Q02 Points} \times 20\%) + (\text{Q03 Points} \times 20\%) + (\text{Q04 Points} \times 30\%)$$

- **People category equation**

$$\text{People Score} = \left[\frac{(\Sigma \text{ Q01 statements})}{10} \times 50\% \right] + \left[\frac{(\Sigma \text{ Q02 statements})}{10} \times 50\% \right]$$

- **Project Environment category equation**

$$\text{Project Environment Score} = (\text{Q01 Points} \times 15\%) + \left[\frac{(\Sigma \text{ Q02 questions})}{3} \times 50\% \right] + \left[\frac{(\Sigma \text{ Q03 questions})}{2} \times 35\% \right]$$

- **Agile approach suitability assessment equation**

$$\text{Agile approach suitability assessment score} = (A \times 33{,}33\%) + (B \times 33{,}34\%) + (C \times 33{,}33\%)$$

A = Organisational Environment score; B = People score; C = Project Environment score.

- **Agile methodology suitability assessment equation**

$$\text{Methodology Score} = \frac{(\text{Q01 Points} \times 3) + (\text{Q02 Points} \times 2.5) + (\text{Q03 Points} \times 2) + (\text{Q04 Points} \times 1.5) + \text{Q05 Points}}{10}$$

- **Organisational Environment category questions and statements percentages**

Table 26 - Organisational Environment category detailed percentages

Organisational Environment	Percentage
Question 01	30%
Question 02	20%
Question 03	20%
Question 04	30%
Statement A	*7,5%*
Statement B	*7,5%*
Statement C	*7,5%*
Statement D	*7,5%*

Note: The sum of the percentages of the statements in question 4 results in the total value (%) of the question.

- **People category questions and statements percentages**

Table 27 - People category detailed percentages

People	Percentage
Question 01	50%
Statement A	*5%*
Statement B	*5%*
Statement C	*5%*
Statement D	*5%*
Statement E	*5%*
Statement F	*5%*
Statement G	*5%*
Statement H	*5%*
Statement I	*5%*
Statement J	*5%*
Question 02	50%
Statement A	*5%*
Statement B	*5%*
Statement C	*5%*
Statement D	*5%*
Statement E	*5%*
Statement F	*5%*
Statement G	*5%*
Statement H	*5%*
Statement I	*5%*
Statement J	*5%*

Note: The sum of the percentages of the statements in questions 1 and 2 results in the total value (%) of these questions.

- **Project Environment category questions and sub-questions percentages**

Table 28 - Project Environment category detailed percentages

Project Environment	Percentage
Question 01	15%
Question 02	50%
Sub-question 1	*16,67%*
Sub-question 2	*16,67%*
Sub-question 3	*16,66%*
Question 03	35%
Sub-question 1	*17,5%*
Sub-question 2	*17,5%*

Note: The sum of the percentages of the sub-questions in questions 2 and 3 results in the total value (%) of these questions.

Appendix O – Conceptual Model Validation Results

Table 29 - Agile approach suitability results (56 projects)

Project ID	Country	Industry	Project Type	Approach applied	Assessment Score
P01	Brazil	Energy-Oil & Gas	Compliance	Non-Agile	28
P02	Canada	Computer-Software	Processes / Efficiency	Agile	72
P03	United States	Media-all types	Processes / Efficiency	Agile	75
P04	United States	Internet	Sales / Revenue	Agile	78
P05	United States	Food	Sales / Revenue	Agile	69
P06	Brazil	Energy-Oil & Gas	Processes / Efficiency	Agile	58
P07	Canada	Media-all types	Processes / Efficiency	Agile	79
P08	Canada	Financial Services	Sales / Revenue	Agile	50
P09	United Kingdom	Financial Services	Processes / Efficiency	Non-Agile	45
P10	Brazil	Energy-Oil & Gas	Processes / Efficiency	Agile	61
P11	Brazil	Mining	Processes / Efficiency	Agile	64
P12	Germany	Other	Sales / Revenue	Agile	74
P13	Germany	Computer-Software	Sales / Revenue	Agile	73
P14	Canada	Manufacturing	Processes / Efficiency	Agile	63
P15	Germany	Internet	Processes / Efficiency	Agile	82
P16	Germany	Computer-Software	Sales / Revenue	Agile	84
P17	Brazil	Financial Services	Processes / Efficiency	Agile	71
P18	United Kingdom	Other	Processes / Efficiency	Agile	62
P19	Belgium	Govern	Processes / Efficiency	Agile	70
P20	Germany	Media-all types	Sales / Revenue	Agile	76
P21	Canada	Internet	Sales / Revenue	Agile	78
P22	Canada	Media-all types	Processes / Efficiency	Agile	73
P23	Belgium	Govern	Compliance	Non-Agile	44
P24	United Kingdom	Food	Sales / Revenue	Agile	69
P25	Brazil	Financial Services	Sales / Revenue	Agile	75
P26	Brazil	Telecommunications	Processes / Efficiency	Non-Agile	51
P27	United Kingdom	Health care	Processes / Efficiency	Agile	67
P28	Portugal	Financial Services	Processes / Efficiency	Agile	81
P29	Canada	Internet	Processes / Efficiency	Agile	79
P30	United States	Govern	Compliance	Non-Agile	35
P31	Belgium	Govern	Processes / Efficiency	Agile	71
P32	Portugal	Financial Services	Processes / Efficiency	Agile	67
P33	United States	Health care	Research & Development	Agile	84
P34	United States	Energy-Oil & Gas	Compliance	Non-Agile	33
P35	Belgium	Govern	Processes / Efficiency	Agile	62
P36	United Kingdom	Internet	Sales / Revenue	Agile	83
P37	United States	Food	Processes / Efficiency	Agile	68
P38	Portugal	Financial Services	Processes / Efficiency	Agile	76
P39	United States	Financial Services	Processes / Efficiency	Agile	68
P40	United States	Financial Services	Processes / Efficiency	Agile	78
P41	Germany	Financial Services	Processes / Efficiency	Agile	71
P42	United States	Energy-Oil & Gas	Processes / Efficiency	Agile	80

P43	Canada	Internet	Sales / Revenue	Agile	82
P44	Germany	Health care	Processes / Efficiency	Non-Agile	47
P45	Portugal	Financial Services	Sales / Revenue	Agile	69
P46	United States	Hospitality	Sales / Revenue	Agile	68
P47	United States	Health care	Processes / Efficiency	Agile	88
P48	Canada	Computer-Software	Sales / Revenue	Agile	82
P49	United States	Food	Sales / Revenue	Agile	80
P50	United States	Energy-Oil & Gas	Processes / Efficiency	Agile	79
P51	United States	Computer-Software	Processes / Efficiency	Agile	86
P52	United States	Internet	Processes / Efficiency	Agile	86
P53	Brazil	Telecommunications	Sales / Revenue	Agile	82
P54	United States	Media-all types	Sales / Revenue	Agile	84
P55	United States	Food	Sales / Revenue	Agile	71
P56	United States	Internet	Sales / Revenue	Agile	82

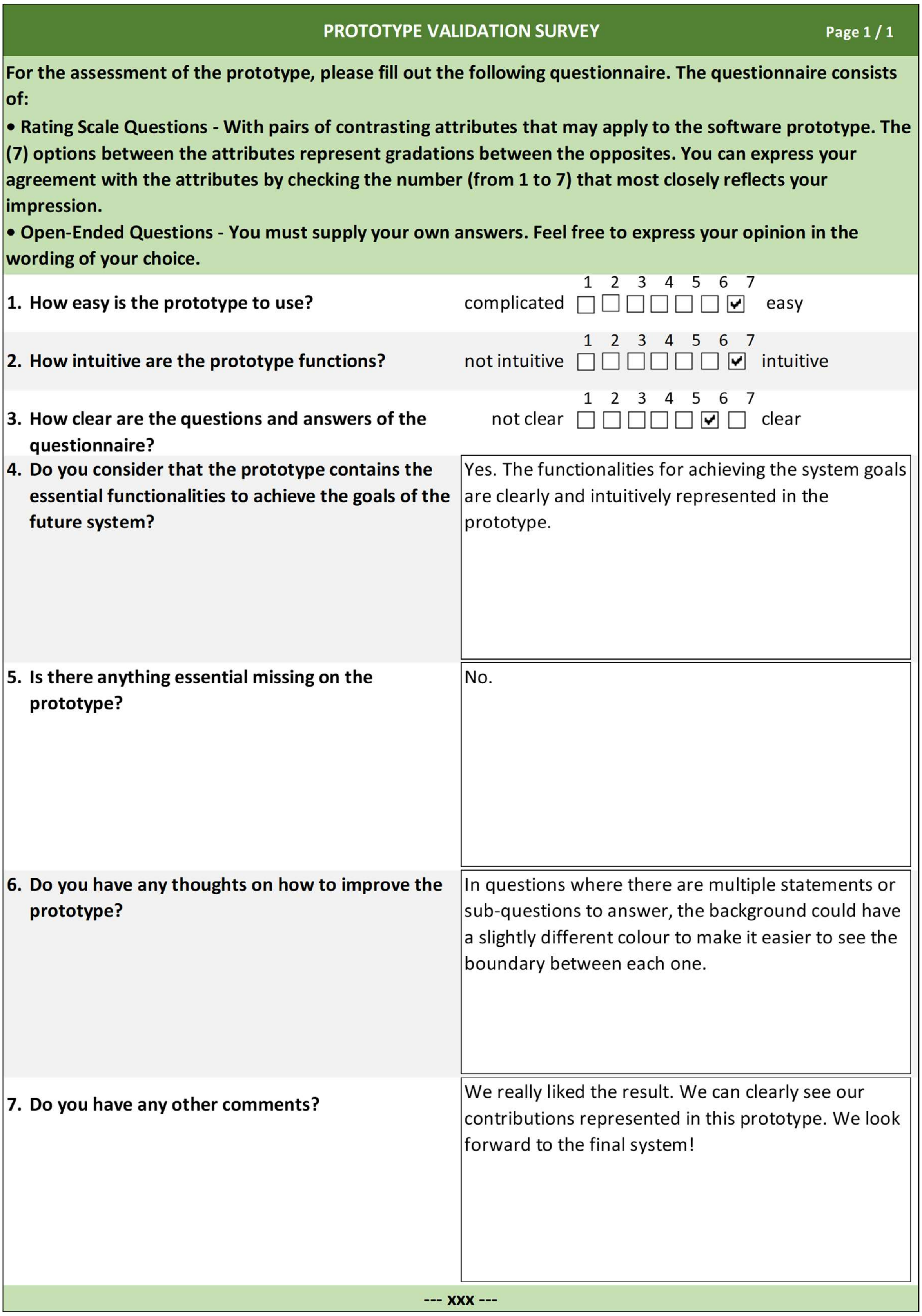
Table 30 - Agile methodology suitability results (49 projects)

Project ID	Country	Industry	Project Type	Methodology Used	Assessment Result (score)
P02	Canada	Computer-Software	Processes / Efficiency	XP	XP (88)
P03	United States	Media-all types	Processes / Efficiency	Scrum	Scrum (100)
P04	United States	Internet	Sales / Revenue	Scrum	Scrum (100)
P05	United States	Food	Sales / Revenue	Scrum	Scrum (75)
P06	Brazil	Energy-Oil & Gas	Processes / Efficiency	Kanban	Kanban (65)
P07	Canada	Media-all types	Processes / Efficiency	Scrum	Scrum (100)
P08	Canada	Financial Services	Sales / Revenue	Scrum	Scrum (55)
P10	Brazil	Energy-Oil & Gas	Processes / Efficiency	Scrum	Scrum (88)
P11	Brazil	Mining	Processes / Efficiency	Scrum	Scrum (88)
P12	Germany	Other	Sales / Revenue	XP	XP (100)
P13	Germany	Computer-Software	Sales / Revenue	Scrum	Scrum (75)
P14	Canada	Manufacturing	Processes / Efficiency	Kanban	Kanban (90)
P15	Germany	Internet	Processes / Efficiency	Scrum	Scrum (100)
P16	Germany	Computer-Software	Sales / Revenue	Scrum	XP (75)
P17	Brazil	Financial Services	Processes / Efficiency	XP	XP (75)
P18	United Kingdom	Other	Processes / Efficiency	Kanban	Kanban (100)
P19	Belgium	Govern	Processes / Efficiency	Scrum	Scrum (90)
P20	Germany	Media-all types	Sales / Revenue	Scrum	Scrum (100)
P21	Canada	Internet	Sales / Revenue	XP	XP (100)
P22	Canada	Media-all types	Processes / Efficiency	Scrum	Scrum (100)
P24	United Kingdom	Food	Sales / Revenue	Kanban	LSD (78)
P25	Brazil	Financial Services	Sales / Revenue	Scrum	Scrum (75)
P27	United Kingdom	Health care	Processes / Efficiency	LSD	LSD (100)
P28	Portugal	Financial Services	Processes / Efficiency	Scrum	Scrum (100)
P29	Canada	Internet	Processes / Efficiency	Scrum	Scrum (90)
P31	Belgium	Govern	Processes / Efficiency	Kanban	Kanban (73)
P32	Portugal	Financial Services	Processes / Efficiency	Scrum	Scrum (100)
P33	United States	Health care	Research & Development	LSD	LSD (100)

P35	Belgium	Govern	Processes / Efficiency	Scrum	Scrum (70)
P36	United Kingdom	Internet	Sales / Revenue	Scrum	Scrum (88)
P37	United States	Food	Processes / Efficiency	Kanban	Kanban (65)
P38	Portugal	Financial Services	Processes / Efficiency	Scrum	Scrum (90)
P39	United States	Financial Services	Processes / Efficiency	Scrum	XP (75)
P40	United States	Financial Services	Processes / Efficiency	Kanban	Kanban (90)
P41	Germany	Financial Services	Processes / Efficiency	Scrum	Scrum (100)
P42	United States	Energy-Oil & Gas	Processes / Efficiency	Kanban	Kanban (100)
P43	Canada	Internet	Sales / Revenue	Scrum	Scrum (100)
P45	Portugal	Financial Services	Sales / Revenue	Scrum	Scrum (75)
P46	United States	Hospitality	Sales / Revenue	Kanban	Kanban (100)
P47	United States	Health care	Processes / Efficiency	LSD	LSD (100)
P48	Canada	Computer-Software	Sales / Revenue	XP	XP (100)
P49	United States	Food	Sales / Revenue	Scrum	Scrum (100)
P50	United States	Energy-Oil & Gas	Processes / Efficiency	Scrum	Scrum (100)
P51	United States	Computer-Software	Processes / Efficiency	XP	XP (100)
P52	United States	Internet	Processes / Efficiency	Scrum	Scrum (100)
P53	Brazil	Telecommunications	Sales / Revenue	Scrum	Scrum (100)
P54	United States	Media-all types	Sales / Revenue	Scrum	Scrum (70)
P55	United States	Food	Sales / Revenue	Kanban	Kanban (90)
P56	United States	Internet	Sales / Revenue	Kanban	Kanban (100)

Appendix P – Prototype Validation Surveys

- **Group 1 Responses**



PROTOTYPE VALIDATION SURVEY Page 1 / 1

For the assessment of the prototype, please fill out the following questionnaire. The questionnaire consists of:

• Rating Scale Questions - With pairs of contrasting attributes that may apply to the software prototype. The (7) options between the attributes represent gradations between the opposites. You can express your agreement with the attributes by checking the number (from 1 to 7) that most closely reflects your impression.

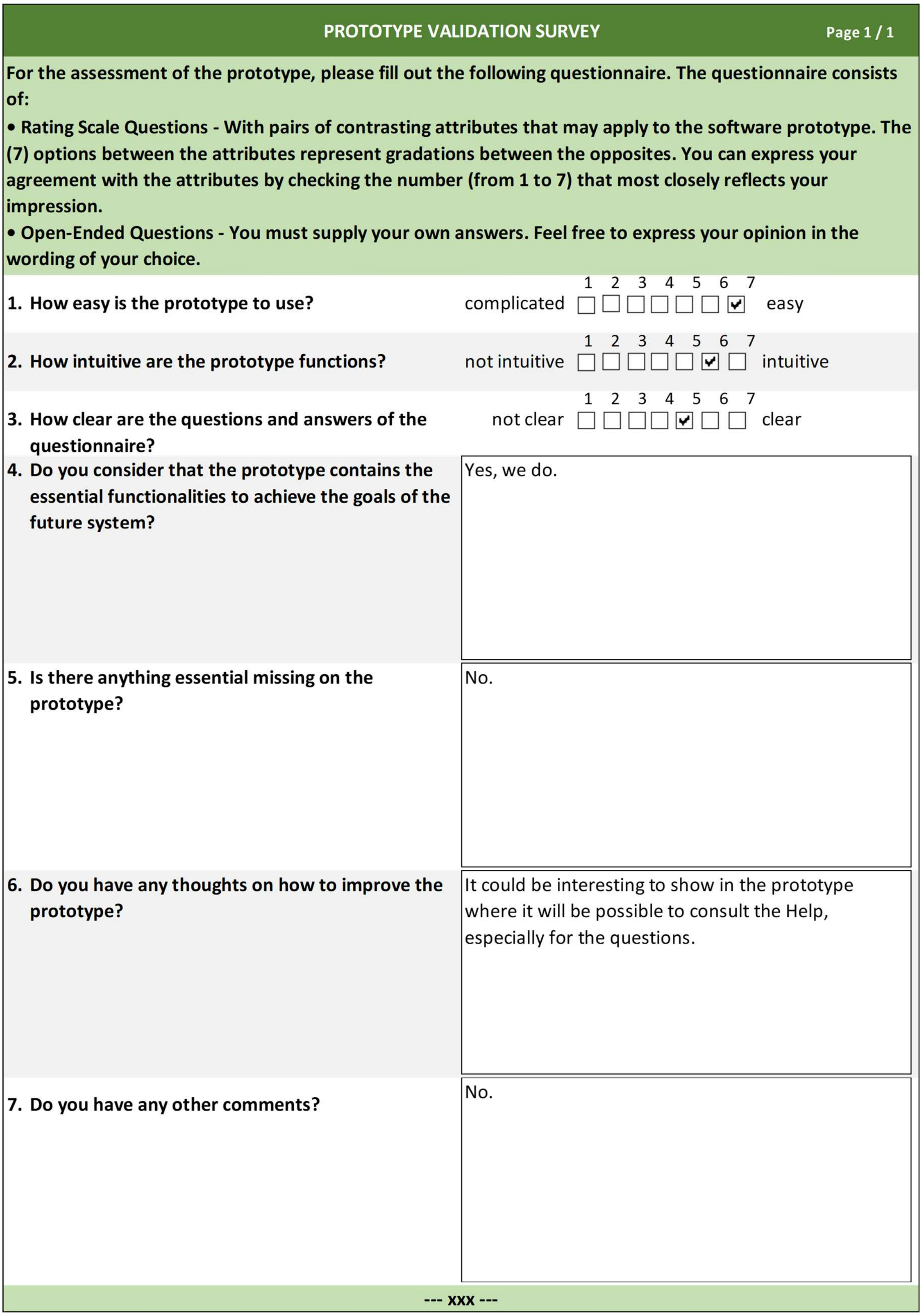
• Open-Ended Questions - You must supply your own answers. Feel free to express your opinion in the wording of your choice.

		1	2	3	4	5	6	7	
1. How easy is the prototype to use?	complicated	□	□	□	□	□	□	☑	easy
		1	2	3	4	5	6	7	
2. How intuitive are the prototype functions?	not intuitive	□	□	□	□	□	□	☑	intuitive
		1	2	3	4	5	6	7	
3. How clear are the questions and answers of the questionnaire?	not clear	□	□	□	□	□	☑	□	clear

4. Do you consider that the prototype contains the essential functionalities to achieve the goals of the future system?	Yes. The functionalities for achieving the system goals are clearly and intuitively represented in the prototype.
5. Is there anything essential missing on the prototype?	No.
6. Do you have any thoughts on how to improve the prototype?	In questions where there are multiple statements or sub-questions to answer, the background could have a slightly different colour to make it easier to see the boundary between each one.
7. Do you have any other comments?	We really liked the result. We can clearly see our contributions represented in this prototype. We look forward to the final system!

--- xxx ---

- **Group 2 Responses**



PROTOTYPE VALIDATION SURVEY **Page 1 / 1**

For the assessment of the prototype, please fill out the following questionnaire. The questionnaire consists of:

• Rating Scale Questions - With pairs of contrasting attributes that may apply to the software prototype. The (7) options between the attributes represent gradations between the opposites. You can express your agreement with the attributes by checking the number (from 1 to 7) that most closely reflects your impression.

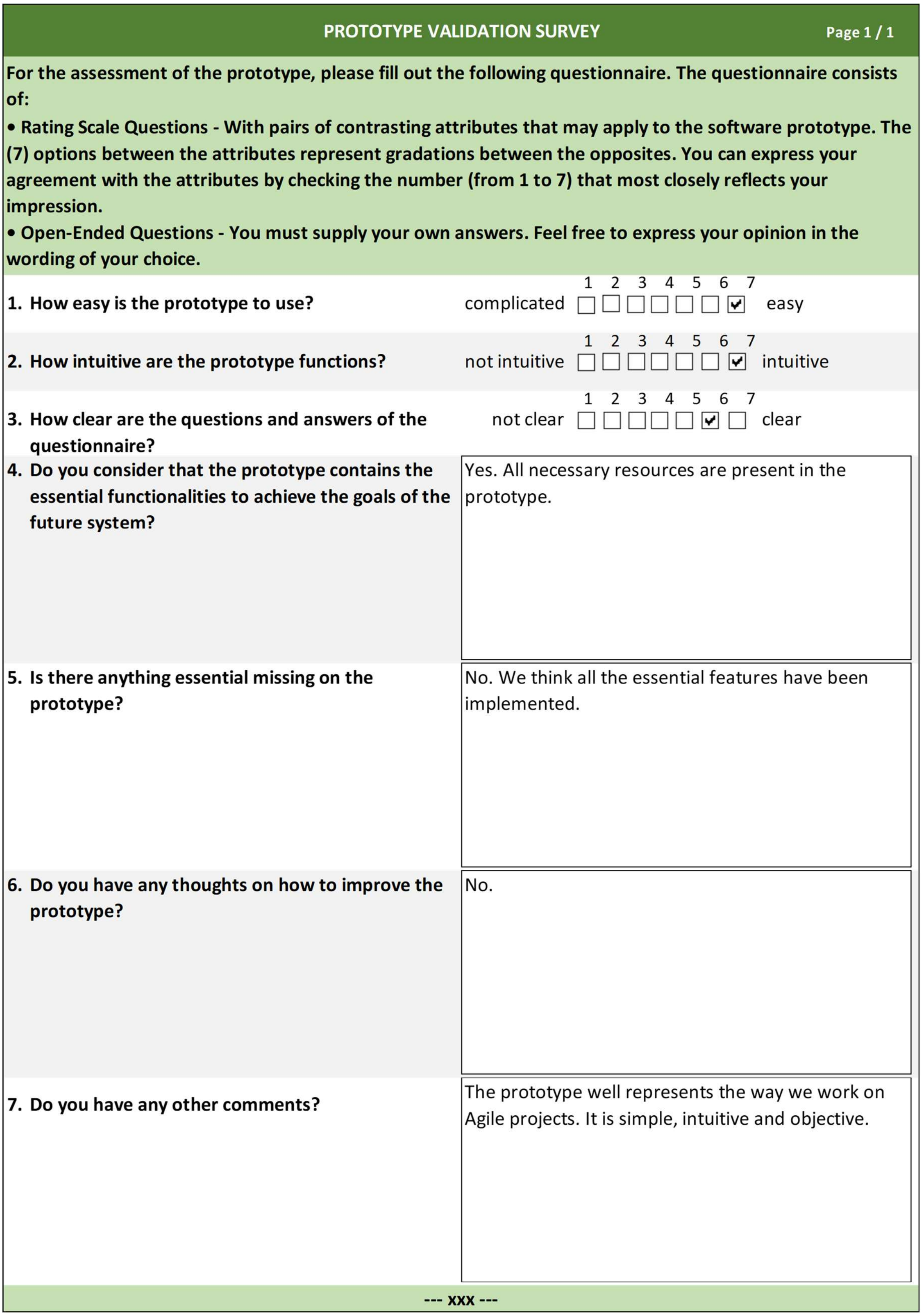
• Open-Ended Questions - You must supply your own answers. Feel free to express your opinion in the wording of your choice.

		1	2	3	4	5	6	7	
1. How easy is the prototype to use?	complicated	□	□	□	□	□	□	☑	easy
2. How intuitive are the prototype functions?	not intuitive	□	□	□	□	□	☑	□	intuitive
3. How clear are the questions and answers of the questionnaire?	not clear	□	□	□	□	☑	□	□	clear

4. Do you consider that the prototype contains the essential functionalities to achieve the goals of the future system?	Yes, we do.
5. Is there anything essential missing on the prototype?	No.
6. Do you have any thoughts on how to improve the prototype?	It could be interesting to show in the prototype where it will be possible to consult the Help, especially for the questions.
7. Do you have any other comments?	No.

--- xxx ---

- **Group 3 Responses**



PROTOTYPE VALIDATION SURVEY **Page 1 / 1**

For the assessment of the prototype, please fill out the following questionnaire. The questionnaire consists of:

• Rating Scale Questions - With pairs of contrasting attributes that may apply to the software prototype. The (7) options between the attributes represent gradations between the opposites. You can express your agreement with the attributes by checking the number (from 1 to 7) that most closely reflects your impression.

• Open-Ended Questions - You must supply your own answers. Feel free to express your opinion in the wording of your choice.

		1	2	3	4	5	6	7	
1. How easy is the prototype to use?	complicated	□	□	□	□	□	□	☑	easy
2. How intuitive are the prototype functions?	not intuitive	□	□	□	□	□	□	☑	intuitive
3. How clear are the questions and answers of the questionnaire?	not clear	□	□	□	□	□	☑	□	clear

4. Do you consider that the prototype contains the essential functionalities to achieve the goals of the future system?	Yes. All necessary resources are present in the prototype.
5. Is there anything essential missing on the prototype?	No. We think all the essential features have been implemented.
6. Do you have any thoughts on how to improve the prototype?	No.
7. Do you have any other comments?	The prototype well represents the way we work on Agile projects. It is simple, intuitive and objective.

--- XXX ---