MESG

MESTRADO EM ENGENHARIA DE SERVIÇOS E GESTÃO

The Quality Assurance for Agile: A New Quality Assurance Method for Agile Software Development

Yuri Caroline Calheiros Curty

Master Thesis

Supervisor at FEUP: Prof. Jorge Teixeira Grenha

Supervisor at Msg Life Iberia: Engenheiro Aníbal Couto

U.PORTO

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

2019-06-28

"If you hear a voice within you say 'you cannot paint,' then by all means paint, and that voice will be silenced." – Vincent Van Gogh

Abstract

Agile has become a popular concept in recent years, this strategy has been adopted by many organization for performance enhancement. It targets prompt development cycles to serve as a differentiator in the competitive market of software development and to attend the fast change in customer requirements. Quality Assurance activities are very important to the project execution and its success in terms of product and process quality. When considering QA activities in traditional software development approaches are performed by separated teams and the adoption of agile methods brings implications in quality management changing the roles and responsibilities into agile software development teams. In addition, agile main focus is collaboration, clear communication and constant customer feedback. Hence, this work aims to develop and apply a method to define QA activities to be performed in agile projects to better organize and standardized quality of processes. Making sure that the required activities are included and implemented promoting continuous improvement. This work present the implementation framework of these proposed method, named Quality Assurance for Agile (QA4A) and the application in a Softaware Development Company. The method leverages some standards not presented in agile approaches to achieve quality in the development process impacting in product quality and the dissemination of quality practices.

Acknowledgments

The realization of this master's thesis had important support and incentives without which it would not have become a reality and for which I will be eternally grateful.

I would like to start by thanking Msg Life Iberia for the opportunity to work on this project in this excellent company. I am grateful to Anibal Couto, my supervisor at Msg Life Iberia, for always being available supporting me every time I needed help, even with his busy agenda. I also would like to thank to the Customer Service Team who received me very well, providing all necessary information and resources to integrate the team introducing the company business. Especially, Daniela Varandas who was always available for advice,willing to share knowledge and a dose of daily motivation. In addition, I would like to thank the Voya Project Team for the great work atmosphere, for always giving me the support needed and for always being available to clarify any doubt. Especially Luis Marques and Isaac Souza for all their patience, technical inputs and their leadership qualities within the project.

I also want to thank to FEUP, my MESG colleagues responsible for making the whole study journey more pleasant and I would like to thank my academic supervisor, Professor Jorge Teixeira Grenha, for his guidance, support, availability, all knowledge transmitted, opinions and criticism and his full collaboration in solving doubts and problems that have arisen throughout this work.

I am deeply grateful to my family for the love and all the support they are always capable of providing, especially my parents and my grandmothers who were a great influence and always inspired me to be a better person. I am deeply thankful to my partner, his support and companionship was essential to conclude this journey. Last but not least another thanks goes to my long lasting friends for always being there no matter where in the world.

Thank you!

Table of Contents

List of Tables

List of Figures

List of abbreviations

QA – Quality Assurance

QM – Quality Management

PM – Project Management

TPM –Traditional Project Management

APM – Agile Project Management

SQ – Software Quality

IT – Information Technology

AP – Agile Project

SIT – System Integration Test

UAT – User Acceptance Test

BA – Business Analyst

PF – Project Facilitator

IP – Implementation Lead

CT – Customization Team

PMT – Product Management Team

1. Introduction

Over the years the constant changes in the field of Information and Communication Technologies is being a key enabler and differentiator for service providers.Offering a significant amount of opportunities for companies to develop new and improve their business (Comes and Berniker, 2008). This development creates new service opportunities, which make this sector more attractive and therefore more important in the global market. Digitalization and the speed of change are also creating a lot of pressure causing different management challenges. In software development the adoption of Agile philosophy (Beck et al., 2001) requires a different approach to project management (PM) ensuring the team empowerment facing decision making and the transparency of communication of project progress (Williams and Cockburn, 2003). The main purpose is to successfully deliver projects in terms of time and quality addressing the challenges in the best way possible to keep the platform improvement, maintainability and the control of the progress of the activities.

1.1. Project background

The International Standard Office (ISO) is a worldwide organization reference for certifications and it responsible to set commercial norms and patterns which are adopted by many companies in other to ensure quality. There is a need for very clear communication of the goals to guarantee the achievement of a project objective and all components of its success (Baccarini, 1999). In Agile approach it is not different, a great cooperation and support between the parties involved are required to address the challenges of a dynamic environment. This demand has a significant impact and it is considered a challenge in Agile Project Management (APM) to overcome the need of documentation (Highsmith and Cockburn, 2001). In addition, the flow of information is crucial in Traditional Project Management (TPM), this approach is comprehensively set in areas which comprise different roles and responsibilities to perform required activities (Munns and Bjeirmi, 1996), while APM is based on frequent iterations and constant communication to debureaucratize processes, incentive creativity and promote innovation. In turns, Project Management must be also successful on the control of information flow mixing agility and plan-driven practices (Boehm and Turner, 2003) jointly with the management of the expectations of those involved and to manage different variables in the project environment (Mossalam, 2019). It might influence required processes during the project life cycle, some studies conducted draw the attention to the gaps and similarities within agile project management as opposed to more traditional project management methodologies. They conclude that Agile Methods emphasize some specific knowledge areas more than other. These areas include requirement management, human resource management. Even though not being formally defined agile also address Quality Management (QM) aspects through the focus in perform constant test activities, frequent reviews and the application of standards to facilitate and to guarantee tasks execution. On the other hand it does not explicitly encompass risk management and cost management missing

the definition of important aspects to overlay the whole project management (Fitsilis, 2008). Agile development methodology is centered on people (Cockburn and Highsmith, 2001) consequently, changing the effective role of Software Quality Assurance in projects that adopt agile development creating the need to be more comprehended.

The project will take place in a software development company, Msg Life Iberia. For the company the ability to deliver stable software solving the customer needs and to be in compliance with the latest requirement is the key for the project success and quality achievement. This research was conducted to understand how a method can be established to implement standardized Quality Assurance activities and to use the role of the tester inside the configuration of the specific project implementation model adopted by an organization. It aims to provide a solution to allows data collection, the identification and responsibilities of the team and the main stakeholders, to provide a common understanding of the process to be followed, and assure the validation of the requirements and the performance of quality activities.

1.2. Problem Description

Agile philosophy is about continuous feedback and adaptation according to Highsmith (2001), in other hand quality is built into application of specific standards. While "agile approaches recommend a variety of practices for constant feedback on technical decisions, customer requirements, and management constraints" (Highsmith and Cockburn, 2001) Traditional approaches establish end-to-end quality process more concentrated on product and process control while agile focus on people iteration and customer satisfaction. Timely decision making is essential in agile projects due to its iterative nature, plus there are multiple ways to identify a path to deliver a working software enabling to approach different methodologies during the process that can help to define a unique structure to define a method. Nonetheless projects present different characteristics which make possible to combine aspects of different Project Management Standards with Agile Software Development in a complementary way. It is required to understand each project particularities to propose a unique solution to control the performance of the activities more efficiently to cover problems, such as the lack of information about project agreements and responsibilities. To provide guidance for Quality Assurance teams regarding the project deliverables, allowing the collection of data needed to access project metrics and delivery project progress update.

1.3. Research Question and Motivation

The role of Quality Assurance (QA) in agile methods change significantly splitting quality responsibilities with customer, developers and the integration with business units, transforming QA in a supporting role (Leon and Koch, 2004). It is necessary to understand how QA will be positioned in agile project (AP) and what is important to define which QA activities will be performed according to the specific characteristics of each project

requirements. These circumstances raise the motivation to pursue the following research question:

‘’How to develop a method that supports the Project Management to implement Quality Assurance Activities in Agile Software Development Projects?”

The main objectives is the creation of a structure to establish what is expected for QA activities. Identifying shared responsibilities, task definition, supporting tools and which deliverables are applicable to the project. Only standardizing the collection of important and valuable data, promoting guidance to gather the necessary information when starting a new project and to create a method to control the project progress providing clear communication for the main stakeholders, especially the team members.

1.4. Study Development at Company: MSG Life Iberia

Msg Life Iberia is a branch of the German Group Msg Life founded in the beginning of the 1980s. With more than 35 years in the market the organization is a leading provider of software, advice and cloud solutions. The company is located in the city of Porto and also has a second office in Spain. Msg Life Iberia has been developing IT systems solutions for the Insurance Market and successfully advising customers on the implementation of customized IT strategies to attend their business needs through the delivery of tailored solutions. Presenting a continuous growth, the company is investing in human capital and hiring more professionals to integrate into the teams in the upcoming projects. The company maintains focus on successfully performing the promised service while guaranteeing the quality and constantly the consistent delivery of working software, simultaneously they take care of the platform improvement and maintainability.

The customers of Msg life primarily consist of insurers focusing on life insurance and pension fund institutions in Europe and, in the United States, in particular health insurance providers. The services range from the development and implementation of standard software and the provision of consultancy services to the handling of full IT solutions. On the basis of their extensive market experience, highly qualified employees develop state-of-the-art solutions to provide customers with comprehensive support. “The software architecture used is component-based and service-oriented and can be used in digitization projects providing a combination of specialist knowledge, process skills and IT expertise” (Annual Rep 2018).

1.5. Report Outline

The structure of this thesis is as follows: Section 2 introduces the key concepts covering the literature review with the necessary foundations addressing the main themes that are relevant to the study. Section 3 gives a detailed description of the methodology used to conduct the work presenting the design research approach justifying the reason for the investigation method used in the project of this master thesis and the design search process. After that in Section 4, the description of search process, the method created are presented. It also presents

the assigned project which the method is applied. Followed by the results and the evaluation of developed solution application presented in Section 5. The final sections discuss the proposed solution, as well the lessons learned on the study, pointing out future developments in Section 6, before concluding the paper in Section 7 presenting the proposed solution achievements and its impact.

2. Literature Review

It was required to focus in few key concepts for this research and this chapter will present the theoretical topics, as well the relevant work produced in the field of expertise. Presenting some relevant information about project management (PM) and the quality management applied in traditional projects. Followed by the most important objectives and concepts behind the agile philosophy and the introduction main characteristics of the scrum framework together with the main challenges related to the theme. Also, approach the most common software quality management aspects to provide the basis to understand quality in software development. Finally, introduce the quality practices applied in agile and scrum.

2.1. Project Management

The International Standards Office (ISO) is a worldwide organization for standardization. They define a project in the following way "A project consists of a unique set of processes consisting of coordinated and controlled activities with start and end dates, performed to achieve project objectives." ISO introduced the Guidance on Project Management (ISO 21500) describing the relation of Project Management concepts, strategies and introduce opportunities to develop business cases and achieve the organization objectives. The guide presents the project as a system with boundaries where projects interact with the project environment and this same environment may impact project performance and success. The annex I provide an overview of project management concepts and their relationship (ISO, 2012). In Software Development the project management method were aligned with the waterfall model, introduced by Winston W. Royce in the 1970s. Frameworks, such as PMBOK or PRINCE2 started being adopted by the practitioners due to the facilities managing individual phases responding a plan (Fitsilis, 2008) (Stoica et al., 2013) (Tomanek et al., 2015). However, it resulted in many cases of failures even though these fails do not demerit the cases of success, TPM approaches will work well for certain types of projects and some success elements can be integrated with different Project Management methods (Fitsilis, 2008).

According to Badiru (2011) project management should achieve a given goal rapidly and efficient. Even though the objectives which constitute the specified goal may differ in terms of complexity, cost, time and results. (Badiru, 2011). The Project Management Institute (PMI) define at their webpage the "global standards as the foundation work process to help ensure the organization's knowledge and frameworks are up-to-date." PMI introduced the Project Management Body of Knowledge (PMBOK), a broadly widely accepted instrument applied to guide, rule and list the characteristics distinguishing the main components of Project Management necessaries to achieve professional excellence (Brioso, 2015).

Experiences from successfully managed projects are collected in what is commonly designated as project management standards (Fitsilis, 2008). The compatibility of different management systems are possible through ISO standards once the sequence and process

adaptation are flexible which "processes may be combined and arranged in sequences according to what the management system has anticipated. ISO 21500 does not describe the inputs and outputs of processes, which increases their capacity to self-adapt to any management system. This integration would allow flexibility to integrate elements commonly used in different standards of Project Management to convert conventional processes." Although the writer highlights the similarities between ISO 21500 and the PMBOK, he also names multiple reasons as to why these management systems and enable the use of methodologies make it compatible if used in an adequate way (Brioso, 2015).

However. Fitsilis (2008) compares a generic set of project management processes (PMP) as defined in two different project standard. Approaching agile project management (APM) in comparison with traditional project management (TPM) methodologies. From the TPM point of view APM methodologies cannot be considered complete, since there is not enough information about the process description or it is missing entirely while in TPM having complete information is obligatory. Using PMBOK as the reference to compare with APM methodologies he argues that agile methods do not define all aspects necessary in the traditional aspects to overlay the whole project management aspects and indicates the focus of agile in the following knowledge areas as being (Fitsilis, 2008):

- Scope Management, since emphasis is given in managing requirements.
- Human resource management, since emphasis is given in team work.
- Quality management, even though not formally defined, use of standards, testing and frequent reviews are promoted.

And by contrast present the information that agile methods do not fully address the knowledge areas indicated below:

- Risk is not managed explicitly
- Cost management is not part of the agile methodologies
- Procurement management is not addressed at all.

As presented in the figure below, the project management aspect involving quality highlighted in comparison made by Fitsilis (2008) of the PMBOK and Agile Methods Processes

QUALITY MANAGEMENT APPROACH (Fitsilis, 2008)	
TRADITIONAL	**AGILE**
• Quality Planning	• Domain analysis for domain model • Comprehensive product backlog list • Functionality and release definition • Immediate development selection
• Quality Assurance	• Testing
• Quality Control	• Backlog items progress review

Table 1: Comparison table of PMBOK and Agile Methods Processes (Fitsilis, 2008)

The project management processes goals differ from project to project in software development and some consideration should be made to simultaneously guarantee the process alignment and process adaptability. Werder et al. (2019) suggests doing tradeoffs to balance incremental and breakthrough changes as an approach to increase the performance of software development project consolidating the need of the organizations became more innovative to accommodate changes. Conversely Abrahamsson et al. (2017) argue about the mechanicism of traditional methodologies and the challenges to use the approach for complete. In addition to this Boehm and Turner (2005) recommend the integration of agile practices with traditional processes to better configure the activities according to the project requirements needs providing proper alignment by building up processes selecting indispensable assets (Boehm and Turner, 2005). However, it is necessary to reinforce that selection and prioritization is not an easy task nor a static process due to the constant modification and evolution of requirements, not being the same case with the traditional development methodologies (Aamir and Khan, 2017).

2.2. Agile Project Management

During the past years the constant advance in technology and the increasing competitiveness of the industry led to the emergence of new services and software products. To address these changes, it was required to modify the traditional project management format, classified as indifferent to business condition, dismissing essential modifications occurred on a volatile environment. Agile proposes a different strategy, recognizing the benefit of accepting these changes while concentrating the effort to reduce the cost of responding to required modifications, once the elimination of changes is not possible. Hence, the concept of agile methodology emerged based on interactions to facilitate the flow of information and allowing fast adaptation to changing circumstances. (Highsmith and Cockburn, 2001).

Agile is a software development approach consisting of a set of methods with empirical lightweight process characteristics. It is introduced by four comparative values and has been

used in the past years to attend the unforeseeable and often times conflicting demands of the industry aiming at customer satisfaction and quality (Williams and Cockburn, 2003). Its design encompass the human element factors to enhance each individual strength inside Agile projects teams to achieve success. (Cockburn and Highsmith, 2001). The Agile Manifesto was written under the following values:

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

Following the manifesto principles:

- "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them an environment, support their needs and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
- While there is value in the items on the right, we value the items on the left more." (Beck et al., 2001).

In the history of the Manifesto (Highsmith, 2001) ensures that "The Agile movement is not anti-methodology" and presents the need to establish an equilibrium between planning and documentation, taking into consideration the turbulent characteristics of the environment and types of a software development project. The Manifesto does not exclude the use of planning and documentation, although it is necessary to carefully evaluate and propose those to comply

with the values determined by agile methodology. In addition, technology is promoting globalization, decreasing the costs and raising the incentives for leverage software development in distributed environments. Therefore, cultural integration, communication and trust becomes a vital aspect in Agile application increasing the attention to managerial implications in different environments and locations (Sutherland et al., 2008). Reinforcing the demand for a better understanding of the challenges related to agile application Hoda et al. (2008) highlights the importance of the role of the project manager in an agile projects in management of off-shored or outsourced agile software projects while Lee and Yong (2019) suggests efficient methods for software development in globalized project management and distributed environments pointing out the best practices to overcome management implications (Lee and Yong, 2009). Agile methods encompass wide-raging life cycles of software development through different practices with varying focus, such as Extreme Programming, Feature-driven development, Scrum and Kanban among others. (Abrahamsson et al., 2017)

2.3. Scrum

Scrum is a light-weight agile methods commonly used in small or middle size development software development projects. It involves collaborative work and acknowledgement of self-organization (Kniberg and Skarin, 2010). Agile Software Development has improved the product delivery process by making software engineering practices simpler through the application of methodologies able to achieve a higher success ratio than traditional projects. Scrum is an simple Agile framework within which can be applied various processes and techniques “for developing, delivering, and sustaining complex products” (Schwaber and Sutherland, 2017). Composed by Teams and their associated roles, events, artifacts, and rules serving as essential components to scrum application and success.(Lacey, 2012). It is used to manage and to plan software projects from an empirical perspective where Scrum theory is sustained by three pillars (Schwaber and Sutherland, 2017):

- Transparency to guarantee the visibility of the process to those responsible for the outcome. Through the definition of standards to share a common understand between all aspects and members involved.
- Inspection of Scrum artifacts and progress toward a Sprint Goal detecting variances but must not interfere with work.
- Adaptation enabling prompt adjustment to minimize deviation.

2.3.1. Roles, Artifacts and Ceremonies

The framework of Scrum is constituted by three components including roles, ceremonies and artifacts. Roles are Scrum master, Product Owner and Development team which are responsible to organize priorities, define process and facilitate tasks to be executed.

Ceremonies are time framed events and artifacts are tools which help inspection and adaptation promising interactive and incremental deliveries forming self-organized and cross-functional teams designed to optimize flexibility, creativity and productivity (Schwaber and Sutherland, 2017).

Roles: There are three different roles on the Scrum, the *Product Owner* is responsible for maximizing the value of the product to the customer while providing the work for the development team (Schwaber and Sutherland, 2017). He also takes care of the project's overall requirements creation and the return on investment goals for the company (Cho, 2008). The second is the *Scrum Master who is* accountable for Scrum dissemination and understanding, responsible for resolving impediments and support the product owner and development team. Last in sequence but not least in importance, the Development Team is responsible for delivering a potentially releasable Incremental working software at the end of each iteration (Schwaber, 2004) grouping all the need skills to develop a product with quality. It is typically composed out of small numbers of members, but larger projects might require an increased amount of developers. (Rubin, 2012)

Artifacts: The main objectives of the artifacts in Scrum is to promote the flow of information to maximize transparency and provide a common understanding about the goals, updates and any other information required during the development. Artifacts creates opportunities for inspection and adaptation on the project course (Schwaber and Sutherland, 2017). Artifacts are composed out of a total of four components, starting with a collection of functional and non-functional requirement (Cho, 2008) called Product backlog. They are controlled by the product owner according to the business value to the customer encompassing all requirements that must be implemented. The next artifact is The Sprint backlog which categorize the list of items that defines the objective of sprint prioritizing those items that have highest business value as categorized by the product owner. Followed by the definition of done, a common understanding is required and may vary significantly per Scrum Team. It will document and explain the definitions of completed or ready for the user stories. (Schwaber and Sutherland, 2017) To complete the set (Lacey, 2012) approach the use of Burndown Charts as an artifact to track sprint progress and rastreability.

Ceremonies: The ceremonies are events that help to promote the transparency on the project evolution to all involved in the software development(Megahed and Ramadan, 2016). Pre-established time-boxed events used to minimize the urge for encounters not defined in scrum framework and create opportunities to inspection and adptation. It must include all events to mainting the transparency and avoide failures on the monitoring process. The crucial events are (Schwaber and Sutherland, 2017):

- Grooming is ceremony involving the product owners and the development team. It is held to present and estimate the stories based on the changes taking into account the complexity and the level of abstraction. The estimates of stories are made considering

that may, or may not bet part of the next sprint, its priory rely on negotiation between the product owner and the customer.

- Sprint Planning is the sprint planning ceremony where the user stories delivery are defined. Once the definitions of what user stories will be part of the sprint the development are done the team decides how to implement it during the sprint. The user stories are broken into smaller tasks detailing the activities that will be base for the story completion.
- Daily Scrum is in fact brief daily meetings, with maximum 15 minutes, of the scrum team comment and observe the progress of the product development. Each member of the development team basically report to the others the progress of their work, answering three main questions: What has been done since the last meeting, what do is intend to do until the next meeting and what are the impediments that block the work.
- Sprint Review Meeting is the ceremony which the user stories developed are presented to the product owner and others involved as a kind of showcase where the product owner analyzes if what was developed can be considered completed. The result of this meeting is to have a feedback and receive the definition of having increment of the software approved or not by the product owner.
- Sprint Retrospective is an open meeting where everyone should express their opinions in order to implement improvements for the next sprints collecting the raised issues to create an action plan with items to be improved.

2.3.1.1. Scrum Framework Flow

Typically, divided in cycles between 15 to 30 days called Sprints that delivery the work in an iterative mode. The Sprint depicts a time-box within which a set of activities must be performed. The functionalities to be implemented are kept in a list known as Product Backlog and in the beginning of each sprint a Planning Meeting where the product owner indicate and prioritize the items belonging to the product backlog. Then the team select the activities to be implemented in the sprint that will be initiated moving the items from the Product Backlog to the Sprint Backlog. Every day in the sprint cycle the team will do a brief meeting in the morning called Daily Scrum. The main reason is to provide each team member a moment to share important information about the activities performed in the previous day, identifying issues that may rise and help to prioritize the work at the current moment. At the end of each Sprint the team will introduce the functionalities implemented in a meeting called Sprint Review. Lastly, a Sprint Retrospective is made and the team start working for the next Sprint planning, then the cycle will be initiated again (Schwaber and Sutherland, 2017). Rubin reinforce that the framework is favorable to the project management and its execution (Rubin, 2012). The figure below present the scrum framework flow:

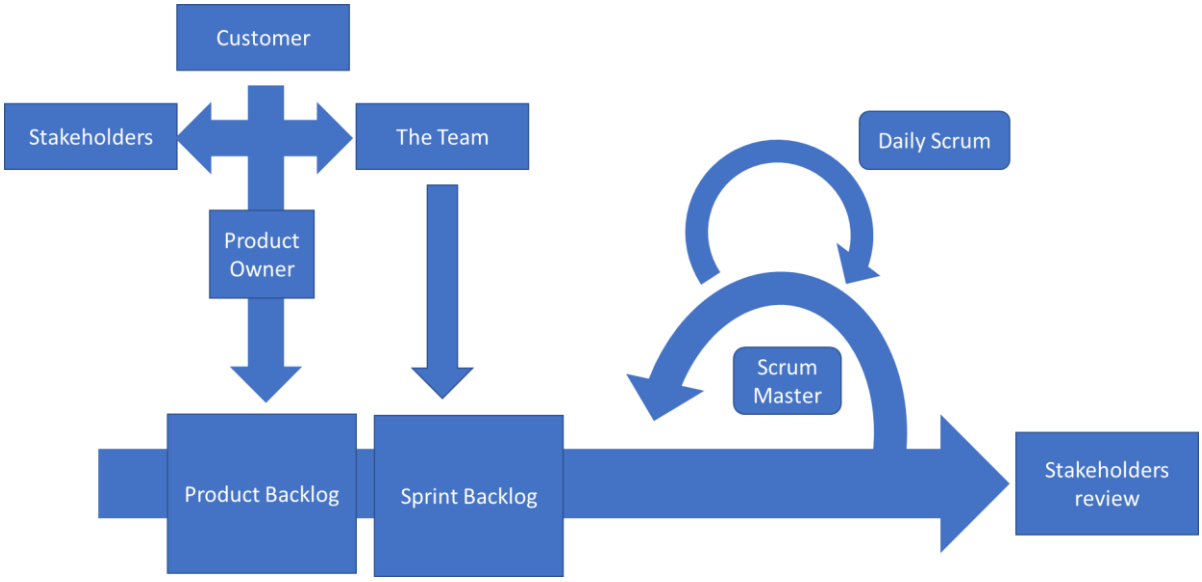


Figure 1: Scrum Framework

2.3.2. Challenges in Scrum

In this subchapter is mentioned the current challenges and work on the field Although scrum is considered simple to understand and a Lightweight framework, it is also difficult to master (Schwaber and Sutherland, 2017). Igaki et al. (2019) address some challenges to scrum´s practitioners and points out the need to understand the components and concepts of the framework to overcome these issues to perform their activities better. The first challenge is the concept of self-organization, aiming to promote empowerment of each member of a scrum team facing decision making to spontaneously find better ways to fulfill their tasks becoming more independent. The second challenge brings the attention to parity in learning opportunities, to promote and improve the distribution of knowledge within the team (Igaki et al., 2014). Scrum will incur many different challenges, according to Cho (2008) the organization will have less distress if they address prior to project initiation issues comprising areas, such as documentation, communication, user involvement, working environment where Scrum ceremonies are well-structured and resolved before the project starts (Cho, 2008). A different angle is approached by (Megahed and Ramadan, 2016) bringing up challenges from the requirement engineering perspective as the lack of requirements techniques, Non-functional requirements and scalability issues to understand how scrum framework leverage benefits of requirement engineering. Many other challenges are raised by the community, however it is important also to address a very important element to project management: the need to meet the customer needs moving the software development towards Agile User-centered Design approach (Sy, 2007). Lacey reinforces that the efficacy of Product management and work techniques used is relative to the team, working environment and the application of tools to facilitate the work to be done (Lacey, 2012).

Addressing these challenges there is an emerging body of knowledge arousing within the agile community. The boost of technologies raises opportunities to promote studies in the Agile field influencing the appearance of new methods including those related to Scrum, such

as, Scrumban (Ladas, 2009) a hybrid approach combined with Kanban methods that creates complementary capabilities to best adapt to organization's needs. Making use of a different approach focused on testing the security of software in Scrum projects Mohamed and Ramadan (2016) propose another framework as an effective solution to enhance software product security minimizing risk and reducing defect management costs(Mohamed and Ramadan, 2016). Targeting the costumer need as a focus of development to try and fill the gap of an agreed-upon term under (UCASD) User centered Agile Software Development (Sy, 2007) (Miller, 2005) (Losada et al., 2012). Brhl et al. explores the principles of UCASD provide contextualization to enhance integration of concepts to address challenges from user-centered design in Scrum while improving the understanding of actual user needs, fostering new practices to adapt software development activities into a dynamic process bounding the development to rigorously execute activities embedded with business value for the costumer. (Brhel et al., 2015). A different approach focused on satisfying the end user needs applying Design Thinking iterative software development to describe requirements, organize teams, and follow the project progress including end-user feedback during all the project development phases. The framework makes the use of scrum to integrate design thinking into the agile projects (Lucena et al., 2016). Tomanek et al. (2015) lay emphasis on the combination of the principles and processes of project management framework and scrum practices with particular emphasis on projects accommodating business changes is an effective way (Tomanek et al., 2015).

2.4. Software Development Quality

In modern society Software Systems are used to facilitate people's daily life, work and societal functions. Therefore these systems must be reliable while meeting people's expectations (Tian, 2005), in turns, quality is multifaceted and can be perceived from different perspectives. Five different conceptions of quality are identified by Garvin (1984). Firstly, the transcendental view which consider that quality cannot be defined, although can be recognized. The product perspective defines quality in terms of vital characteristics of the product itself. Followed by the user view that defines quality as something that is adequate for a purpose which will have variation according the user needs. On the other hand the manufacturing view defines quality as conformance strictly to requirements and specification. Lastly, the value-based view approach quality by customer wiliness to pay (Garvin, 1984). On the other hand quality considered a concept a part of quality assurance processes and other means of monitoring. It indicates that the difference between quality and quality assurance is that the second is responsible for verify the quality of a process or an outcomes (Harvey, 2006). Presented bellow are the general themes of Software Quality Engineering, which comprises three fundamental areas (Tian, 2005):

- Software testing is essencial to ensure software quality;

- Other alternatives for quality assurance (QA), including defect prevention, process improvement, inspection, formal verification, fault tolerance, safety assurance, and damage control;
- Measurement and analysis to close the feedback loop for quality assessment and quantifiable improvement

In addition the organization must promote the continuos quality improvemet mind set to guarantee quality of all system, processes and activities reducing the need for large number of inspection is as a way to achieve quality (Neave, 1987).

2.4.1. Agile Quality Practices

Regardless the approach selected for developing software applications, it involves the management of many elaborated activities that are prone to errors. Stoica et al. (2013) highlight the importance of testing and validation before the delivery of working software. Being necessary to guarantee the transparency and that the customer will receive the requested and correct functionalities (Schwaber and Sutherland, 2017). It is reinforced that all software development approaches chosen and the process adopted will influence the software quality (Stoica et al., 2013).

A distinct element of agile methodoly compared to traditional approaches is the capacity to endure unstable requirements across the development lifecycle and deliver products in concise timeframes ensuring quality. Agile embrace technologies and requirement changes during the projects causing relatively short projects plans updates (Williams and Cockburn, 2003). Agile teams have freedom to choose architecture, requirements and design (Dovleac and Ionică, 2017), even though quality management is not formally defined or standardized it makes more difficult to control the quality during development process. It make necessary to discuss an important factor in software development, the role of quality management in agile software projects and the definition of QA process. In addition, quality activities will be defined according to the time that the development team will have available to tailor their process into agile practices and how much they will know about the agile methodology application (Fitsilis, 2008).

QUALITY MANAGEMENT APPROACH		
SOFTWARE QUALITY (Tian, 2005)	**TRADITIONAL (Fitsilis, 2008)**	**AGILE (Fitsilis, 2008) (Schwaber and Sutherland, 2017)**
• Quality planning	• Quality Planning	• Domain analysis for domain model • Comprehensive product backlog list • Functionality and release definition • Immediate development selection
• Selection and execution of Quality Assurance • Software validation and verification activities	• Quality Assurance	• Testing
• Measurement and analysis • Evidence to demonstrate software quality	• Quality Control	• Backlog items progress review

Table 2: Quality Management Description

Analysing agile software quality assurance and Software Quality Huo et al. (2004) investigates quality-centric software development supporting processes and presents the quality practices which agile methods have incorporated into their processes to ensure quality while accommodating the time factor and the volatile requirement environment. These authors identifies that agile methods move some QA duties and liabilities to the developer and argue about the importance of each member of agile project to understand more about the quality of the software produced and emphasizes the importance of the developers to acknowledge how to modeler their agile methods to grant the quality required during the interations and mentioned some quality practices, as higlighted below, adopted by agile methods that have QA potential (Huo et al., 2004):

On-site customer, where customers and developers working together. The customer support the development team throughout all development process to refine and correct requirements.

Pair programming, two programmers working on the same code to improve design quality aiming to reduce the amount of defects through code review process.

Refactoring is a restructuring technique for an existing code, where its internal structure is modified without promoting changes in its external behavior while preserving transformations to produce a significant restructuring.

Continuous integration, a practice to constantly keep the system fully integrated to promote early identification of problems.

Acceptance testing occurs earlier and more times in an agile development being done several times after unit test gathering more frequently customer feedback. (Huo et al., 2004).

2.4.2. Agile Testing

In traditional software development approaches analyzing test as a solo phase is considered a heavyweight process performed by separated teams where stories are expanded, coded and tested (Crispin and Gregory, 2009). Crispin and Gregory (2009) states that the core practices for agile testing success is the commitment of the entire team with quality, a team responsible for testing, provide support and training needed for testers and developers, promoting integration of the testers and developers to work in pair. The organization must select the best combination of people to compose the team (Crispin and Gregory, 2009). However, agile approaches present some implications related with planning, requirements management, estimation, task division, working collaboration and it also include testing (Puleio, 2006). Hendrickson (2015) present four statements to create a responsive system for agile testing approach:

Recovery over Perfection: If when developing the software the Team focus in delivering perfection it is going to became a hard mission, since it is not possible to predict the software behavior when it goes to production due to the amount of factors involving its use.
Predictability over Commitment: Use historical information and systems to promote visibility and predict outcomes improving future deliveries instead of committing with one thing. Allowing adaptation when assumptions are not realistic and embrace failures as a learning component.
Safety Nets over Change Control: Instead of rigorously control the nature of the changes, make it safer to change things.
Collaboration over Handoffs: Bring people working together to save time instead of throwing a set of deliverables on the next member of the team (Hendrickson, 2015).

According Crispin (2009) agile change the whole team approach transforming code and testing in one process where feedback is the main collaboration key. Test-infected developers promote the use of better tools and better designed tests (Crispin and Gregory, 2009)

Agile Testing Approach (Crispin and Gregory, 2009)	
Phase	**Activities**
Project Initiation	• Get an understanding of the project
Release Planning	• Participate in estimating stories • Create Test Plan
Each iteration	• Write and execute story tests • Write and execute new functional test cases • Pair test with other testers, developers • Automated new functional test cases • Run automated regression test cases
System Test / End Game	• Perform Load Test • Complete Regression Test • Perform UAT • Perform Mock Deploy • Participate in Release Readiness
Release to Production / Support	• Participate in Release to Prod • Participate in Retrospectives

Table 3: Agile Testing Approach

2.4.3. Scrum Quality Challenges

Scrum as an agile framework has it quality build in the interactions through the transparency of the communication to overcome the lack of documentation and pre-established process. The scrum process outcome is to delivery functionalities in form of time framed interactions (Schwaber and Sutherland, 2017). The relationship between process quality and product quality is difficult to manage, in scrum framework quality assurance ends up being overlooked due to the short amount of time and the scarcity of fully committed Quality Assurance representative in scrum projects. Aamir (2017) states the need to focus not only on the stories success criteria and user acceptance testing to deliver a valuable solution and argue that incorporate other activities to monitoring quality in scrum may benefit the project quality management and the software quality. Introducing a concept to sustain test cases, the test backlog consist in validate the product before its release. It provides quality assessment on the basis of frequency of remaining functional bugs verifying deviation for the requirements (Aamir and Khan, 2017). Also approaching the backlog and quality activities, Keyes et al (2006) present an additional framework for quality management proposing a measure based on the complexity of the Product Backlog and test assessment rating. Rather than measure product quality the framework focus on the evaluation of software testing process quality, concluding that "the quality of a testing process cannot be precisely measured as a lot of development-driven factors are involved. These factors can alter the quality of the product. Not only development-driven factors, but also deployment-driven factors could jeopardize the quality of the product. For example, a bug in the automated deployment scripts could end up not deploying some modules of a product. In this case, all functionalities dependent on those modules will malfunction. Our hope is to assess the test quality up to some degrees" (Kayes et al., 2016).

3. Methodology

There are different design research approaches each one adequate better to different fields and it is adapted according to the theme of the each investigation (Winter, 2017). Information System is a field composed by a large variety of disciplines and being agile software development designated to be the application area of this study, and to comply with the environment conditions where the research was conducted this study will use Design Science Research (DSR) methodology. DSR work as problem-solving process applied through the design and construction of innovative artifacts to solve specific and meaningful problems in the organization, consequently impacting on financial aspect, such as it profit and (Von Alan et al., 2004) resolving, in an innovative manner, problems never solved before. Also being applied to efficiently sort out already known, or previously addressed but not completely fixed problems (Gregor and Hevner, 2013) . These problems are characterized by having one, or more, of the following characteristics (Hevner and Chatterjee, 2010):

- Unstable requirements and restrictions based on poorly defined environments.
- Complex interactions between the subcomponents of the problems and their solution.
- Flexibility enabling to support changes to the design process as well as to the design of the artifact.
- A critical dependence on human cognitive abilities (e.g., creativity) for the production of effective solutions.
- A critical dependence on social skills (e.g. teamwork) for the production of effective solutions.

Some guidelines are establish s to conduct the process of DSR (Hevner et al., 2004) (Peffers et al., 2007). These guideline have references on prior literature about Design Science and Information system aims to help with the definition of a mental model to structure the studies output (Von Alan et al., 2004). Peffers et al. (2007) suggest the sequence of six activities as methodological basis highlighted below in blue contain a brief description of the activity on the side and in bullets points how it will connect the Knowledge of this work. Aims to promote the understanding of the set of tools used as material to perform each activities and present the processes iteration to provide a clear and consistent definitions for the design and execution of the research (Hevner, 2007). Further is presented a proposal for the design methodology of QA4A.

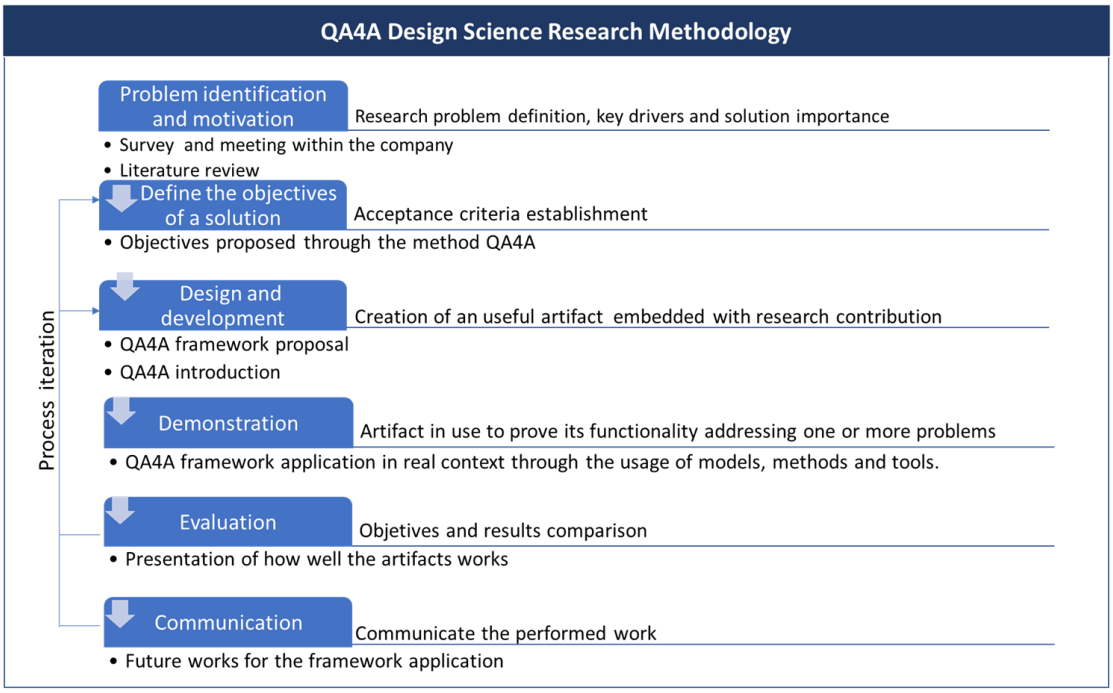


Table 4: QA4A Design Science Research Methodology

4. Solution Design and Development

Aiming to organize the investigation process it was necessary to define a proposal to perform the research activities in phases to support the design of the solution to implement QA activities, referenced as the QA4A. Considering the scope, dependencies and limitations of this research an adapted solution based on the activities proposed by (Peffers et al., 2007) was used. The model was divided in three phase: problem identification, solution design and development followed by the demonstration and evaluation.

The first phase, *Problem Identification* - A diagnosis is carried out in a way to identify the existing problems and their origin to clarify how QA4A can solve them. Working together with the main stakeholders, accessing the literature and current projects environment to collect, organize and treat information from different sources to define the main goals for the solution. This phase consist in two activities:

Activity 1: *Problem discovery* – This activity it involves collect information from the experts in the company through one-to-one meetings. Accessing the company's procedures and standards to understand and define the problem, describing the scope of the problem and solutions objectives. It is necessary to understand how can be established a method to implement QA activities and use the role of the tester inside the configuration of the specific project implementation model adopted by an organization to establish a method that will define the activities for QA team and the tester that will be member of a project software development team. In order to understand the problem in question in greater detail it was necessary to follow how some of the daily tasks were being performed by individuals in current Msg Life Iberia understanding the processes adopted for certain task execution. In addition, an online survey using google forms (Appendix A) was conducted with the main stakeholders to respond open-ended-questions to try to capture more value for the solution to be designed. To contribute with the augmentation of knowledge base through a scientific process the literature was also used to access the problem state-of-the-art and support the problem discovery by the study of existing work as presented in the chapter 2. The following table present the issues that were raised during this activity and for a better understand it was separated by areas related to new project implementation aspects, quality assurance activities and the supporting tools.

Area	Issues identified
Project	• Lack of identification regarding main stakeholders; • Ensure validation of the requirements; • Customer communication delay; • Lack of structured and well-informed planning; • Lack of requirements description and priorities; • Uncertainty of requirements; • Lack of a guide when collection project information; • Visible and updated timelines; • Lack of common understanding of new development process; • Team members communication issues; • Time loss with incomplete information and incomplete explanations; • Lack of visualization of project progresso; • Lack of information about costumer's expectations and needs; • Feature development wrongly estimated; • Changes in team members.
Quality Assurance	• Lack of QA plan; • Lack of QA processes definition; • The need of integrate QA team with business line; • Lack of requirements understanding per part of team members; • Lack of defect Management; • Implication on test management • Assure compatibility of functionalities with previous versions; • Usability test is not performed; • Environment Issues; • Lack of standard on ticket management; • Lack of metrics; • Lack of test automation; • Lack of organization and document control; • Lack of clear documentation to support the process
Tools	• Lack of training on JIRA

Table 5: Raised issues summary

Activity 2: ***Objectives definition*** – In this activity is necessary to search for new sources of information to increase the knowledge on the problem understanding the current situation of the company regarding project implementation standard, activities and what are the available information for the problem resolution. It takes into consideration the existing solutions and the company state-of-the-art where previous designed artifacts interacts with the supporting literature and internal stakeholders to establish its relevance. Aiming to define which QA activities will be performed in each project through a structured way. Establishing which deliverables is required and which data will be collected to that specific project. Promoting a guidance to gather the necessary information when starting a new project and to create a method to control the project progress and promote a clear communication for the main stakeholders, especially the team members. It also necessary to fit the tester activities inside the process of software development in alignment with the strategy view of the company to bring a valuable solution that can be used for each new agile project implementation. In summary the solution needs to provide the support required by the Leader(s) involved dedicating some attention to the need to measure the project progress while keeping the quality of the information tracked.

The second phase, ***Solution Design and Development*** – This phase will describe the state of the art and current enterprise reality. Reflecting on the organization and the treatment of the information reunited in the previous activities through the method and artifacts creation. The

focus is understand the tools and methods used to develop a common language to communicate with the internal stakeholders achieving the design of a meaningful solution.

The third phase, *Demonstration* – QA4A framework application in real context bring together models, methods and tools to provide a valuable solution and presenting the main results.

The fourth phase, *Evaluation* – The viability is presented in form of a matrix of evaluation, where the solution is compared to the main problems identified in the beginning of the research. It aims to verify the coverage of issues previously presented.

4.1. Quality Assurance for Agile (QA4A)

This chapter take into consideration that all information gathered from different sources is transformed and introduced in the form of a solution. A combination of artifacts to be used in further development of the QA activities implementation solution proposal.

The framework and the application of the artifacts developed as solution to help to identify and communicate the testing processes, QA deliverables and supporting tools to be used in QA activities during the implementation of a new project. Although some quality assurance responsibilities became more a development task within agile (Huo et al., 2004), the framework is adaptable to adopt a test focused QA activities according to project needs. It allows to bring the different tasks together aligned with business needs, reducing the potential misalignments between tasks.

4.1.1. Step 1 - Gathering QA project requirements

It is necessary to study software development environment, customer information, line of business output from customer interaction and the available information to input value to each new project. Therefore different configurations are possible. QA activities should adapt and align with the agreements made by the line of business with the customer getting to know more about the business concept in other to improve the critical judgment regarding the execution of the functional tests. It is important to be aware that some particular considerations will affect resource requirements, even can have schedule impact, requiring multiple collaboration within a distributed software development environment. For better understanding what is necessary to establish QA activities some question should be raised in this phase:

- Which Business Units are involved in the project?
- What it is known about each unit´s responsibility?
- Who will integrate the project development team?
- How the team organization fits on the Project Implementation Standard adopted by the company?
- What you should do to keep this information updated?

- Who is the customer and the business Rules?
- What is the project milestones?

4.1.2. Step 2 - Support and opportunities alignment

This part will describe the general approach and objectives of the test tasks in the project. It must define the testing processes and tools to be used, or testing defined interfaces to external systems. (E.g. dependencies on the test definition on the Service Level agreement, determination of company and customer responsibility even involves outsourced parts). It is necessary to have a mind-set toward continuous improvement, gathering data is important and the process must be reliable. Through the analysis of the data collected on process it is possible to identify change opportunities and supporting tools to help on the improvement of activities related to the project. The analyses of the project configuration and its requirements helps to identify the dependencies on the project, being sometimes extremely important to define a scope for QA activities (E.g. Line of business is located in another country and they are the responsible team facing the customers and delivery all the necessary feedback, the distance influence on the communication for the local team and requires an QA activity more focused in understanding the business problem being solved, thus limiting the functional test to what is written as expected functional behavior). Leveraging opportunities to explore and access supporting tools to test automation, reporting, documentation, supporting tools to facilitate the performance of routine task.

4.1.3. Step 3 - Create Evidence Plan

To facilitate and promote summary of QA activities to be implemented, identify the deliverables and communicate responsibilities the outcome of this step will be the use of the developed QA4A Matrix. It will help to list the deliverables and to define the type of test that the project requires providing a better understand about what and when it will be executed. It is important in this step to define the state of the art of QA activities and present the necessary information to the main stakeholder. (By this time all information necessary about the test scope agreement and its requirements must be known. E.g.: Unit test, Internal testing, Pair programing, Code Review, Code Refactoring, etc.). This central artifact is a top level plan that will be used by managers to govern the planning and control of the test activities and its effort. It defines the approach that will be employed to test the software and to evaluate the results of that testing in that specific project providing clarification for:

- Project Overview
    - Project Progress
    - Capacity Management
- Quality Assurance
    - Test Strategy / Detail Test Plans
    - Test Scenarios

- Test Cases/Test Scripts
- Requirements Traceability
  - Test Planning
    - Defect Management
  - Test preparation
    - Test Cycles
    - Issue Tracking
  - Test Execution
    - Defect Tracking
    - Metrics
  - Test Closure
    - Evaluation Summary Document
    - Lessons Learned

The matrix is the first tool of QA4A, it contains the major information about the project and uses the responsibility classification of RACI[1] and provide transparency over project QA tasks and responsibilities.

R = Responsible will perform the work to complete the task. They must complete the task or objective or make the decision.

A = Accountable also approving authority, only one for each task or deliverable ensures the prerequisites of the task are met and who delegates the work to those responsible. This person must make sure that responsibilities are assigned in the matrix for all related activities.

C = Consulted experts that opinions are sought. Must give input before the work can be done and signed-off on.

I = Informed people kept in the looping of progress up-to-date, do not need to be formally consulted or will have directly contribution.

[1] Role & Responsibility Charting (RACI) SMITH, M. L., ERWIN, J. & DIAFERIO, S. Role & responsibility charting (RACI). Project Management Forum (PMForum), 2005. 5.

QA4A Matrix								
Project Configuration:								
QA:								
QA Objectives:								
Test type (Cross, smoke, Stress, QA):								
Enviroment 1:								
Enviroment 2:								
					RACI			
Phase	Deliverable / Task	Activity Team 1	Activity TEAM 2	Automatic (Y/N)	TEAM 1	TEAM 2	Customer	Comment
Project Overview								
	Timeline							
	Progress Report							
	Status Report							
Test Levels								
	Unit Test							
	Manaual User Story Test							
	System Integration Test							
	Regression Test							
	User Acceptance Testing							
	Non/functional (Performance)							
Deliverables								
	Test Planning							
	Test Scenarios							
	Test Cases/Test Scripts							
	Requirements Traceability (to Test Cases)							
	Test Execution; Metrics, Status, Defect Reports							
	Test Evaluation Summary Document (TESD)							
Manage/Control								
	Change Request Log							
	Test Acceptance Criteria							
	Defect logs							

Table 6: Quality Assurance for Agile Matrix

4.1.4. Step 4 - Define QA Activities

After gathering all necessary information is possible modeling QA activities. It is necessary to select a supporting tool to make it visual. The basis for guidance QA activities should encompass at least some of the following tasks:

- Documentation Evaluation
- Creating Test Documentation
- Test Preparation
- Performing the tests
- Analysis of the results
- Categorization of results

Figure 2: Quality Assurance for Agile Framework

4.1.5. Step 5 – Measurements

It is difficult to guarantee accurate measurements when controlling is associated with the loss of time. However, it is necessary to establish a minimum of document control organization and also make sure the required documents are clearly and available for each phase of the project. The idea is to make the use of metrics flexible to the project condition not determining rigors on measurements, but rather collecting valuable information to create some historical data that will help visualize the project situation in a given time as needed, but the most important is consider quality: Conformity of the product developed not only with the requirements and the number of errors, but also targeting the customer satisfaction. Prior to each project it is necessary to discuss about the definition of done, identify what is the project requirements to define the approach about the important metrics regarding, processes, resources and time, so no time is spent collecting unnecessary information. The outcome will be the measurement of only the required metrics for the project supporting the changes if some modification is need, such as additional metrics or outdated and not applicable metrics.

4.1.6. Step 6 – Communicate

It is necessary to communicate the measurements and important issues related to QA. The method provide a flexibility on the results communication where the agile team can decide what better suits them, but information data must be available if need to be accessed (E.g. Issue tracking enable information on JIRA to be followed in real time, if everyone follows the same approach for issue tracking). It is important to have available the required information in a way that daily reporting can be issued as deemed necessary. If there is an issue that should be communicated immediately and cannot wait until a scheduled iteration to discuss or report release. In addition, the team can leverage different channels of communication. (E.g. scrum iterations, email, shared networks, etc.).

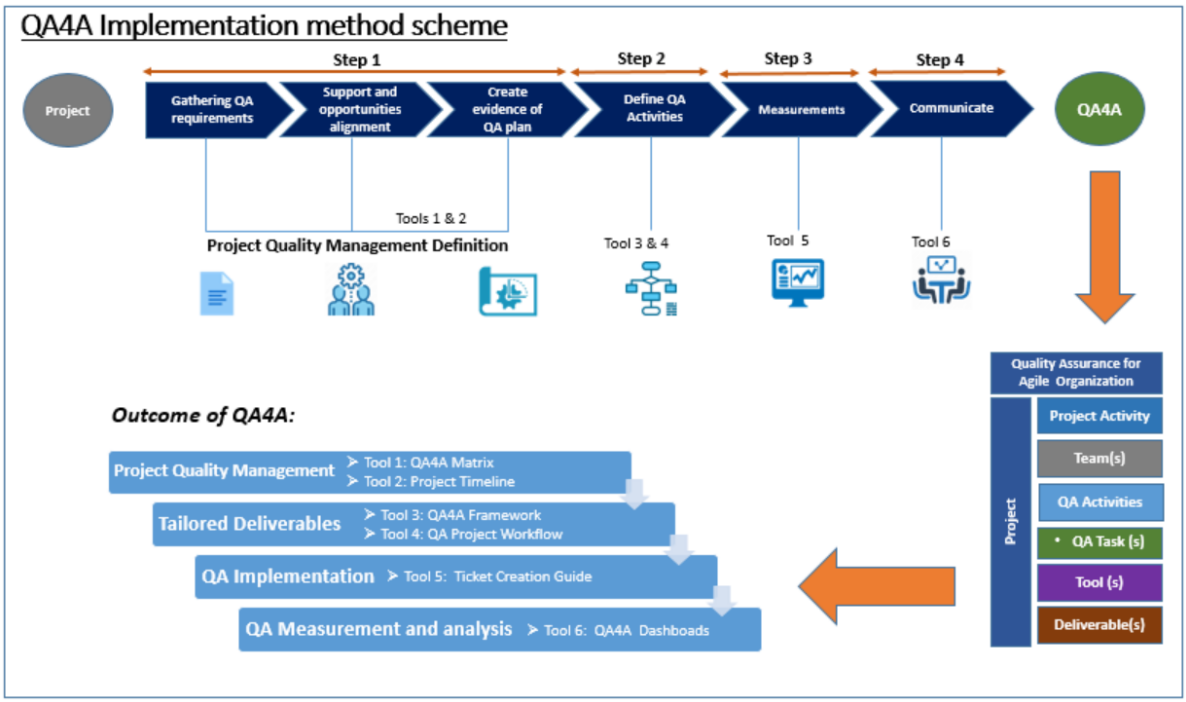


Figure 3: Quality Assurance for Agile Implementation Scheme

5. Solution Demonstration

This chapter will present the background of the project which the QA4A was applied considering the company requirements and constrains for the solution application. It will present the state of the art of performed by Msg Life Iberia QA activities, followed by the application of the method.

5.1. Quality Assurance and Testing activities at MSG

In fact, at Msg life Iberia each project is unique. The scope of the project are set as the customer's needs. In addition, there are different levels of customization for the solution offered by the company. For this investigation the Voya project was assigned to be the main object of this study serving as reference for data collection and the application of the solution proposed in this study. It is important to emphasize that this is a project which face a great challenge due to short delivery timelines, in addition to this not all requirements were defined when the development started presenting the need to be dynamic in terms of defining processes to be followed and defining the tools to support the team and Quality Activities Management. To give an overview on the range of product configuration possibilities the diagram below presented in general terms, as well the Voya project solution positioning regarding to Msg Life Iberia components delivery responsibilities.

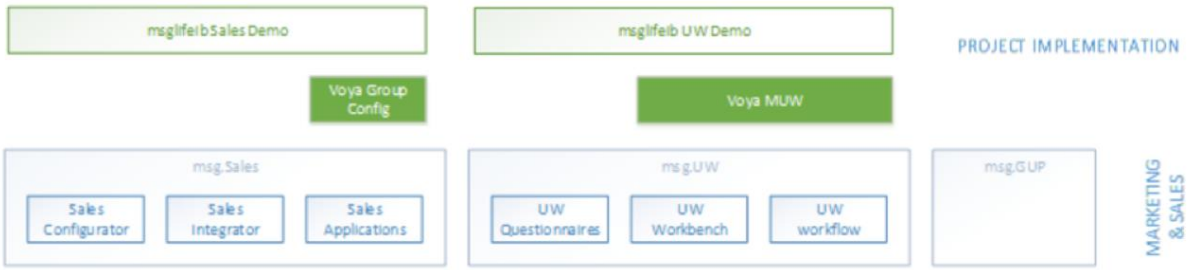


Figure 4: Msg Life Iberia Product Configuration

Msg Agile testing methodology is the approach used to perform testing in software development project at the company. The testing and development are breaking down in two weeks according scrum framework where definable and achievable goals are set. The tests are built not on Use Cases, but rather the more granular user stories, which allows to complete testing on a few user stories per Sprint. When all of the user stories that compose one use case are completed then it is combined to run integration and regression tests on the now completed use case.

Every new project at MSG Life Iberia encompass the general strategic aspects from the management point of view. The managers constantly encourages the empowerment of each individuals when facing decision making situations while remain available for support when required. Hence, the operational level requires that information and tasks are organized,

controlled and communicated to the main managers by the assigned team for that specific project.

5.1.1. Test Levels

The test levels describe bellow are recommended in the product test strategy provided by the Msg Life Iberia:

Unit Test to be completed by the development areas, not as a part of QA. Unit test is performed to validate conformance with the documented technical requirements and design. Unit testing verifies components function independently of other components. For each component the test should focus on all branches of code, conditional variations, loop limits, error handling, invalid entries, maximum/minimum values, window controls, file and data interaction, and user interaction.

Manual User Story Test will begin as soon as the developers finish the Unit Tests. Using the scrum framework, it will not be testing Use Cases in their entirety from the beginning. Each Use Case will be broken down into multiple user stories that will then be manually tested using the acceptance criteria we had set forth for them. They user stories will eventually be combined to form Use Cases that will be tested using:

System Integration Test (SIT) will confirm the product properly reflects the specified requirements. SIT will begin after the Manual User Story Tests are complete. These tests will be run for full Use Cases. SIT will test all changes to the systems impacted as well as their core integration/interface points some level of integration testing between systems may be performed, but the same data may not flow through from start to finish. SIT Testing will be performed in a risk based methodology, as high priority business and technical test cases will be executed by priority.

User Acceptance Test (UAT) will focus on demonstrating the product, as provided, will fulfill its intended use. UAT is testing the business functions of the software on a completely integrated system or systems to validate the system's compliance with the specified business requirements prior to the implementation of the system(s). User Acceptance testing confirms the software can be used by the end-users to perform their 'day-to-day' business functions and tasks specified in the approved requirements and design.

Automated Regression Test will be run as soon as user stories pass the initial tests. Once we are satisfied that a user story has passed individual as well as integration testing, it will then be automated so as to run regression tests on a sprint by sprint basis as new functions are added to ensure that the old functionality is still working properly." (SUSSMAN, 2012)

Out of Scope for the QA activities:

- Plan Configurator controlled by Business, not QA
- Unit test performed by the Development teams
- Build Verification Testing performed by the Development teams

- Environment performed by the Implementation Responsible

5.1.2. Test Cycles

Following the agile testing methodology the company perform continuous test, bug repairs, test cycles that will be prioritized and communicate during the stand-up meetings. Development will then address the highest priority fixes the following cycle and run down the rough timeline for the project on a Sprint to Sprint basis is as presented in appendix E.

The developers generally writes the user stories concurrently with QA responsible and the business analyst, also is defined the acceptance criteria and test scenarios for those stories to be tested in the next Sprint. QA run the test scenarios for the stories that were received from the developer's last sprint. Once QA is done with the manual testing, they then automate these for integration and regression testing.

5.1.3. JIRA

Jira is a supporting tool developed in 2002 by Atlassian JIRA is a ticketing system to track issues and it is considered a project management toolset. It provides advanced customization features to tailor agile project according to team's needs and project requirements (McKenzie, 2018). There are multiple types of project that requires different ways of organizations. In JIRA a project is a grouping of work items called "issues", this issues shares common configurations such as workflows, fields and screens among others. Although it stands alone in the authorization of what information which group will have access, versioning, release index and components (Wong, 2017).

5.2. Quality Assurance for Agile (QA4A) application

Further is presented a summary of each step execution and a final overview of the project itself:

Step 1 **-** Basically this steps means how the tester performed the evaluation of documentation and analyses of project requirement, the activities was separated with two focus:

Project: Access to project plan, timelines and project statement of work.

Project Implementation model: Even though only two components of Voya project was presented in the previously section, the project is composed by others components as described further in this section which contain a brief overview of each unit's responsibilities regarding development and QA. The project involves three units of Msg group: Msg Life Iberia, FJA and FJA QA Offshore teams respectively located in the Portugal, USA and India working together to deliver the software. The tools used for supporting the project organization and communication, as also presented in previous chapter, is JIRA for ticket tracking and confluence for documentation sharing. To preserve the confidentiality agreement between the company and the costumer, only information relevant to QA activities implementation will be presented in this section, clarifying the roles, task and responsibilities

within the project. When a project is initiated the managing director and main leaders define the responsible according to the following pre-determined implementation model (Appendix C) description:

VOYA PROJECT ROLES	
Project Role	**Msg Life Group Units**
Project Manager	FJA
Business Analyst	FJA
Project Facilitator	FJA & Iberia
Implementation Lead	FJA & Iberia
Customization Team	Iberia
msg.Sales Project Development Team	Iberia
Product Management Team	FJA & Iberia
Agile Coach	FJA & Iberia

Table 7: Voya Project Roles

Quality Assurance: An evaluation based on the Functional Specification Documents and artifacts was accessed to define the scenarios, flows and write the test cases. Thus, as soon as the artifacts were presented, the team started working to validate them, participating in the refinement and eliminating possible gaps, and presenting information beyond what is described in the documents. Reading and understanding of User stories. It is necessary to understand and analyze the user stories. In this function the tester should always keep in mind the view of the user and what are the possible real cases that he can perform. It is really important be aware about the business concepts, cases and rules.

Quality Assurance Terms: This section refers to the Project QA responsibilities assigned to each unit of Msg Life Group and also present an overview of how the customer participation will impact the implementation of QA duties and responsibilities in the project consequently establishing all testing activities.

FJA is the unit responsible for provide the quality teams and define how best to test Voya´s applications as contained in the Functional Specification documents signed off by the customer. FJA's Quality Assurance will also use these specifications for regression testing on each deliverable. FJA provided technical support to resolve issues and answer questions raised by the client, and also the internal teams involved in this distributed environment project that requires a clear communication through different time zones due to the location of each unit. The customer and FJA unit will work together to determine if it is part of the original scope, or a change control is required being the USA team responsible to make the bridge with the customer and communicate the defects founded to Iberia and India teams.

Testing Approach: All testing is based on functionality defined in approved Functional Specification Documentation. General Entry and Exit criteria is agreed between FJA and Customer whom will work together to define the test management plan.

- *Test Environment Setup,* FJA and Iberia is responsible to create and maintain the Quality Assurance Test environment within FJA's infrastructure for testing before delivery to production. The costumer is responsible to create and maintain a User Acceptance Test environment for integrated testing of each delivery from FJA, this environment must be created prior to the start of UAT and maintained throughout the duration of the project. FJA will create and maintain an UAT environment matching the version in Voya's UAT environment for issue reproduction.
- *Test Case Development,* FJA was responsible to develop functional test cases based on the approved Functional Specification Documents. These test cases was used by FJA QA offshore team to execute functional testing in QA environment before release and leveraged for regression testing in future releases.
- *Regression Testing,* FJA will develop regression tests to be executed in QA environment to ensure functionality previously delivered to production is working as designed after feature enhancements are complete.
- *Test Automation,* to ensure quality of each deliverable, FJA will develop automated tests based on regression and functional test cases defined above.
- *User Acceptance Testing,* The costume is responsible to develop User Acceptance Test (UAT) Cases to ensure that the delivered functionality meets their functional needs. It must be executed in clients UAT environment.
- *Testing Execution Cycles,* FJA will execute in QA environment predefined functional tests for new features one week before each delivery.
- *Regression Testing,* FJA will execute regression tests in parallel with Functional Testing on week before one before each delivery. The regression tests executed for each release will include a set of regression testing defined for every release and additional regression tests defined by FJA focused on functional areas impacted by code changes in the release.
- *User Acceptance Test,* The customer will execute UAT tests of each release made by FJA. After the UAT test cycle is completed FJA will guarantee that a document is signed and provided to FJA.

Defects Management: JIRA was used for all defect management and bug logging. Issue raised include items documented and reported to FJA. The information presented here refers to Iberia testing efforts before to Production implementation of the software. All defects must be accompanied with the appropriate documentation for FJA evaluation and prioritization. The documentation should include the information which the failure is associated with, the test cases in which a failure resulted, system logs, error messages, configuration or environmental setting where appropriate, and confirmation that the defect is reproducible. In addition to this Iberia QA and FJA QA offshore team should work together to guarantee that the Internal Functional Test are completed and the coordination of bug tracking is done.

DEFECT DOCUMENTATION REQUIREMENTS
• Accurate and thorough description of the issue • External Issue ID • Affected version(s) • Steps to reproduce, including expected and actual results • Test cases or requirements in which a failure resulted • Supporting documents as applicable including system logs, error message screenshots, templates, XML, configuration • Confirmation that the defect is reproducible • Name of submitter

Table 8: Defect Documentation Requirements

For each defect raised as presented in this section refers to a classification will be assigned as mutually agreed the customer and FJA, plus a timeframe agreed with the customer from final delivery to validate functionality will be set and informed by FJA to other involved units.

DEFECT RESOLUTION	
Severity Type	Description
Blocker / Critical	The defect results in the failure of the complete software system with no workarounds/alternatives available.
High	The defect results in the failure of the complete software system; however, there are acceptable processing workarounds/alternatives that will yield the desired result.
Medium	The defect does not result in a failure, but causes the system to produce incorrect, incomplete, or inconsistent results, or the defect impairs the systems usability.
Trivial	Cosmetic problem such as misspelled words or misaligned text.

Table 9: Defect Resolution

Responsibilities are divided per each MSG life unit assigned to develop the components as listed below:

PROJECT RESPONSABILITIES PER UNIT			
Component	Release	Development	Quality Assurance
GC	1	FJA	FJA
GC	2	Iberia	Iberia & FJA
EOI	3	Iberia & FJA	Iberia & FJA
MUW	3	Iberia	Iberia & FJA
UC	4	FJA	FJA
RP	5	FJA	FJA

Table 10: Project Components and Responsibilities

Adapted to fit on the scrum framework used by the company QA activities on Voya Project was mostly focus in monitoring whether the team's activities are adhering to the QA process in addition to collecting the process metrics through the use of the artifacts and application of the method proposed. In the end after all required evaluation, the Matrix created as a tool for QA4A (appendix D) was filled serving as input to the next step.

Step 2 – This step was focused on the discovery of tools to support and facilitate the required QA tasks.

First, in terms of project it was identified the possibilities of supporting tools according to the needs of the project. It was established that all information about the project should be registered and updated on the JIRA project, documented on the confluence[2] and supported by any other tools that could improve the work efficiency.

Step 3 – The selection of QA activities required the attention to guarantee data collection of the performed tasks during the project. In this step also was defined how the tester would execute his task and how this should be integrated with the second team responsible for QA. Due to the project short timeline and the challenges in to work in a distributed development environment not all steps in scrum was followed requiring adaptation of the tester activities too, and there was no planning meeting considered create test plans. Basically, manual test was performed once the functionality was developed by the team. The test case was based on the combination of analyses of the stories produced and information contained in the last version of the functional specification document. In parallel QA FJA team prepares the test case for the whole documentation and test was performed and closed according the parameters were accepted in the test management. The test cycle for Voya project regarding Iberia QA task was established as presented in the figure below:

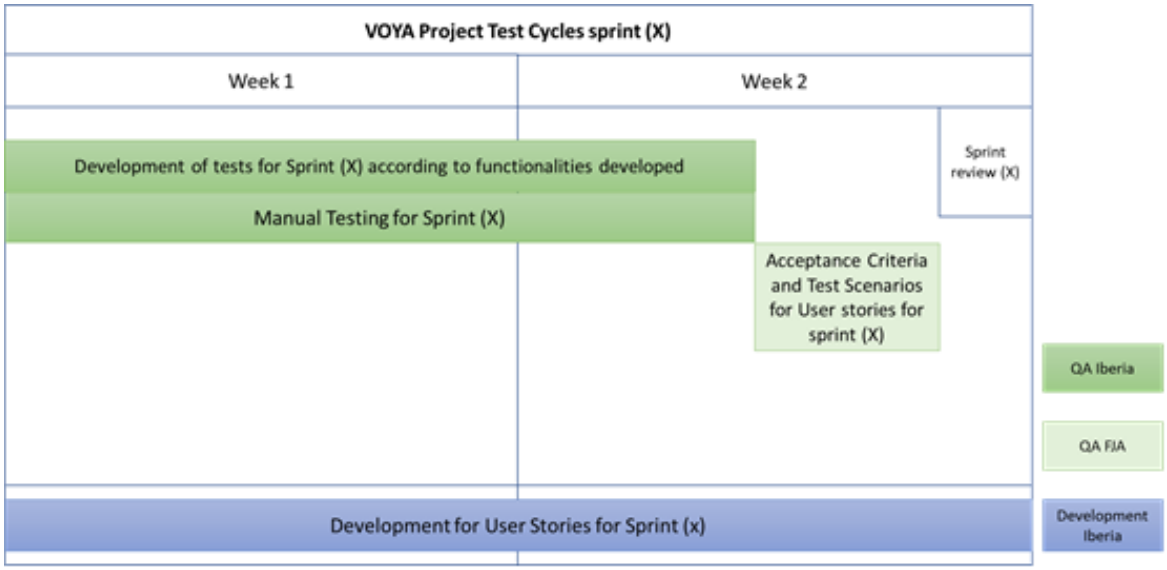


Figure 5: Project Test Cycle

[2] Documentation Management Software produced by Atlassian https://confluence.atlassian.com

Project: The changes in the configuration of the project was tracked and inputted on JIRA. Through Issue Management and the creation of a standard for ticket creation. The responsibility was shared across the team and QA support the management of the information on the system to ensure the data was correct when extracted. The tester was specific allocated for this purpose.

Quality Assurance: Performed QA activities making the use of tools created and the QA4A Scheme Resulting in exploratory test performed with a focus on good communication between testers, line of business and development team. Facilitating the information exchange, the analysis of what has changed, and raising other issues that could be addressed.

In this step is set the requirements of the ticket creation and also was defined the Metrics for the project configuration and the dashboards to be used to tracking the project, QA activities and defect management following the project report scope for data collection (Appendix F).

Step 4 **–** The communication and information sharing. The defect management leverage JIRA for bug tracking while as requested the project progress and status reporting was updated every Tuesday. The scrum standard, such as stand up and review meetings were held. The daily stand up meeting to discuss what we accomplished the following day as well as what we hope to accomplish in the day, and informed any QA important issues. The bug tracking was done together with the development team leader. In addition a bi-weekly meeting at the end of a Sprint to showcase what was accomplished, what went wrong, what could be done better was presented by different members of the development team. The project tracking took place in JIRA with each ticket containing all the necessary information as describe on the ticket creation guide presented in appendix G, assigned to a specific individual and the completion of these was reviewed daily in stand up meetings and bi-weekly the completed of functionalities on the sprint review. Presented below is the framework completed with the summary of the defined activities, process (appendix L), defined deliverables and tools used with the implementation of the QA4A method.

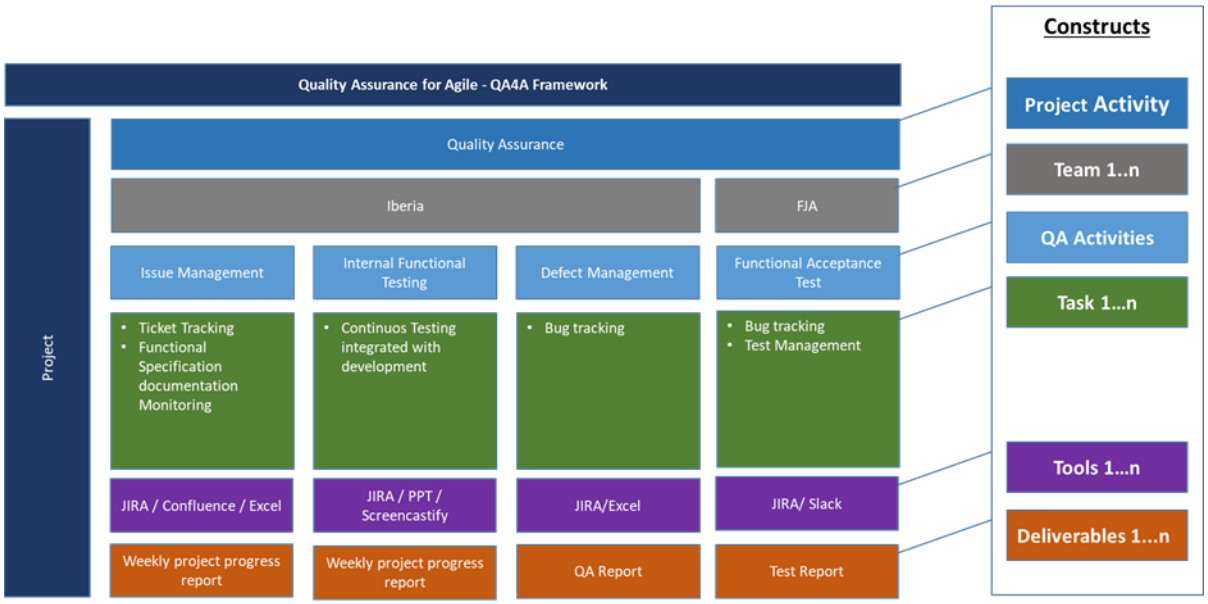


Figure 6: Voya Quality Assurance for Agile Framework

5.3. Voya Project Results Summary

In this section, a brief summary of the project progress until the date of this work delivery. The Project have five main milestones classified by the release version number related to specific components. This research covers the application of QA4A of the release of version 1.9.2.0.0 which is related to a component called Group Configuration. For this component a total of 187 tickets were registered on JIRA since the project kick-off in February and the release date in the beginning of June 2019. The unit responsible for the entire development was Msg Life Iberia. The project have very aggressive timelines and a very extensive list of requirements. It was required the Iberia team to start the development even before the Functional Specification documents (FSD) sign off. In the kick off the FJA had already provide some wireframes and FSD drafts from where initial estimate and storied started being created. From this stories a total of 828 days of work was initial set to organize team effort and after the release of the group configuration a 484 days was remaining to complete the project. Of course, this original estimation is not a fix number. It will change through time according customers need, but the initial idea to track this number is to see how the project will flow according the amount of tickets created on JIRA and the estimations set.

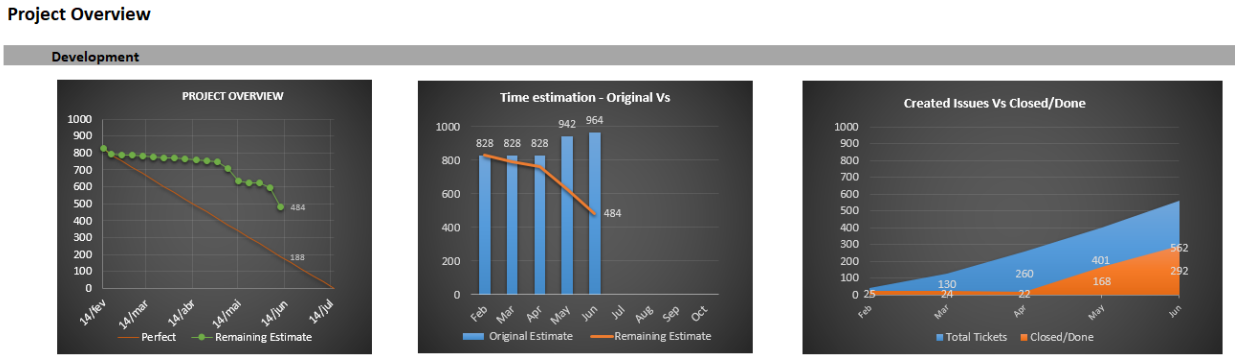


Figure 7: Graphs from Project Progress dashboard

The version 19.3.0.0 is set with the remaining days of work not completed at the first release, total of 56 days and the version 19.4.0.0 encompass the EOI and MUW components which have release date set to July.

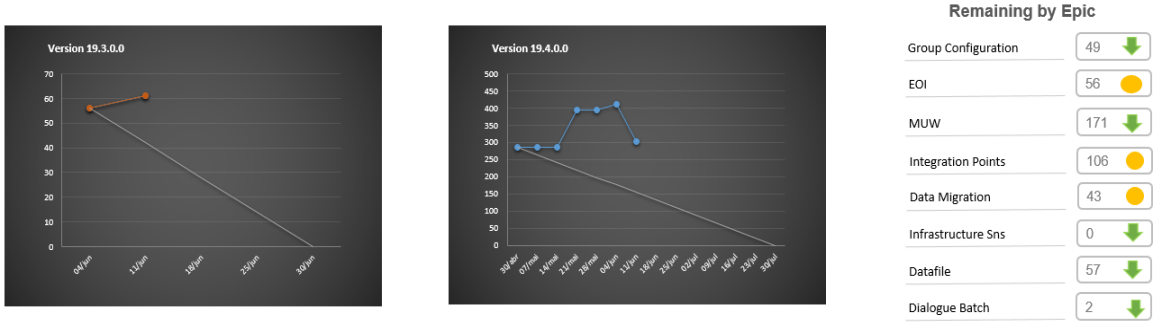


Figure 8: Graphs from Project Progress Dashboard

The total number of bugs tracked on the date of the release of group configuration was 44. These bugs were classified according the root cause identified by the developer when fixing the issue. The classification are splited in the follwing descriptions, code was inadequate or incorrect - Code Review did not catch it, code was inadequate or incorrect - code Review not held, Feature was not described in the requirements, Feature was working once but implementation of some other feature(s) broke it, Metadata Issues, Not Applicable and To Be Determined regarding to those which still missing information.

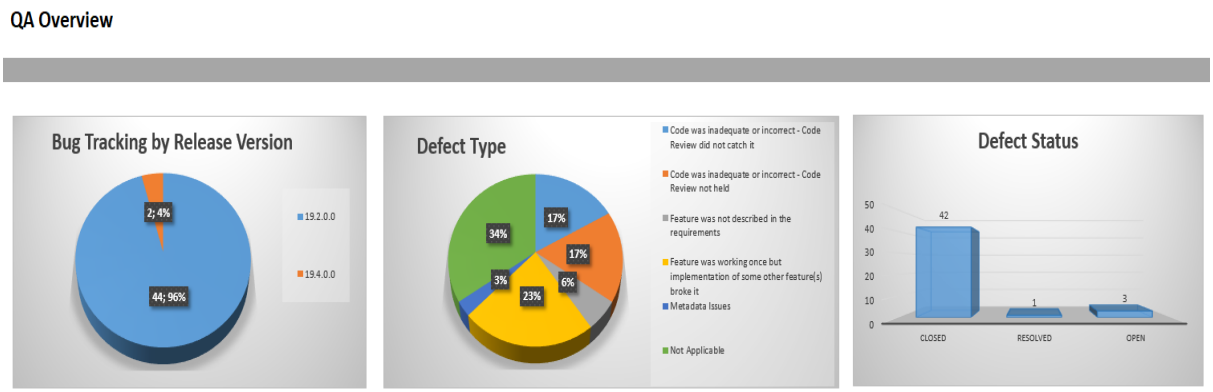


Figure 9: Graphs from QA Dashboard

Until the day of the the release a total of 46 defects have been found and 42 issues were already fixed, 1 issue was resolved, but it was in test and 3 remained open, being relocated to have fix priority on the version 19.3.0.0. The open issues were classified an categorized. Two issues as medium priority not resulting in a failure, but could cause the system to produce incorrect, incomplete, or inconsistent results, or also could be that the defect impairs the systems usability, and one classified as trivial being considered a cosmetic problem.

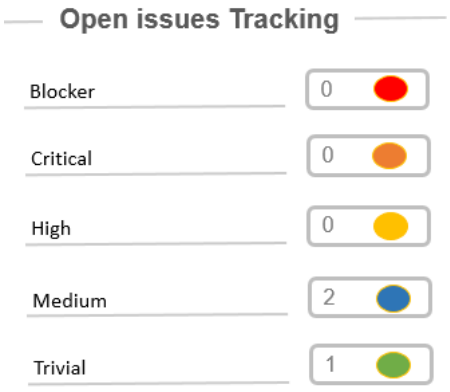


Figure 10: Defect Tracking

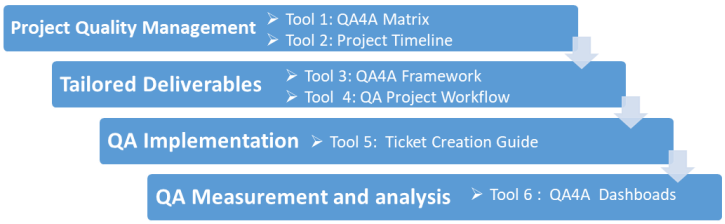
Regarding to test activities it was necessary to create a very simple test plan documentation, direct and clear presenting the area that the test case should cover, the test case, the preconditions if applied, description of the expected result, if the functionality was implemented or not on the day of the release, the date of the latest test executed and the JIRA

ticket number if any issue was raised and it was open on the day of the test. It was posted on confluence as internal smoke test (appendix I), but it was containing also information about the issues to be tracked if it was open until the date of the release. Also, in this step the documentation was created in the confluence and JIRA used by the company and was stored as history, later serving as a basis for testing and creating templates for more complex features. It was performed a "validation" of written test cases, issues tracking to verify if the coverage was complete, and or if was necessary to update, add or remove any test or scenario that was considered unnecessary, always with the help of the development team

6. Evaluation

The QA4A method aimed to establish a basic alignment between PM and software development in different kind of projects. It was necessary to identify the current problems faced by the company and understand the issues that cause the actual problems to find the solution by understanding the current processes. At the same time it was possible to establish means of measurements to provide strategy visibility providing a common and understandable language for the agile team's communication. Ensuring the measurement of the right and verifiable data making use of the QA4A tools to present the main aspects required for the project to define the organizations of QA activities.

The case application of QA4A described the general approach and objectives of the QA activities according to specific development project configuration. Msg Life Iberia participation on Voya project did not contemplate specific activities generally performed in QA. Activities such as unit test, system integration, performance and the cycles of testing of build verification test, regression test are performed by a secondary unit of the companies group. The initial idea was to compare the test results with customer's acceptance test criteria and be able to have an evaluation in terms of "quality", once the main objectives of the solution established was to delivery functionalities without defects as customer needs. As the responsibility of tests in the project object of study was duty of a secondary unit of the company, and due to time constrains, since until the delivery of this work the Voya had not delivered the results of executed test on the software after it was released and it was in production. To promote some overview of how the artifacts involved on the method proposed access the identified problems in the first phase of this work during the project discovery. A brief description of the issues identified in the company and if the artifact coverage or not it – answered by Y for yes and N for No – is presented in the table below.



QA4A ASSESSEMENT BOARD							
Area	Description	Tool					
		1	2	3	4	5	6
Project	• Main Stakeholders responsability identification	Y	N	Y	N	N	N
	• Allow configuration according project characteristics	Y	Y	Y	Y	Y	Y
	• Structured planning	Y	Y	Y	Y	Y	Y
	• Requirements follow up	N	N	Y	N	N	N
	• General Project information	Y	Y	Y	Y	N	Y
	• Timelines	N	Y	N	N	N	N
	• Process Clarification	Y	N	N	Y	Y	N
	• Team communication	Y	Y	Y	Y	Y	Y
	• Project progress information	N	N	N	N	Y	Y
	• Promote information for new member	Y	Y	Y	Y	Y	Y
Quality Assurance	• QA plan	Y	N	Y	N	Y	N
	• QA processes definition	N	N	N	Y	Y	N
	• Defect Management	Y	N	N	N	Y	Y
	• Test management	Y	N	N	N	N	Y
	• Ticket management	Y	N	N	N	N	Y
	• Metrics	Y	N	N	N	Y	Y
	• Document control	N	N	N	N	N	Y
		Y	N	Y	N	Y	N
Tools	• Issue management with JIRA	Y	Y	Y	Y	Y	Y

Figure 11: Quality Assurance for Agile Assessment Board

Main improvement: The method helped describe the general framework and the main objectives of the testing tasks for the designated project. Leveraging the use of data as a key resource to create value from understand how data contributes to quality management. During step one and two it was possible to gather enough information to have a global overview on the project creating a QA project Plan. It provided a better organization and track of the tasks. The project progress tracking was active and being updated once per week.

Regarding the JIRA tools used by the team for issue tracking and defect management had benefits due to the Standardization of the ticket creation and the required information fulfillment. It improved the early identification of problems and the access of project information. It allows the use of productivity and support tools utilizing technology to help the daily task executions and control.

Downsides

It is necessary to dedicate some time to ready the documentation, to set meeting with BA and developers to promote more improvements and align QA goals. The configuration of the project did not allowed to analyze in real time export graphs directly from JIRA and was necessary to use excel to reproduce the graphs regarding to information that the local unit needed.

Future Work

The QA4A encompass a global view of Quality Assurance activities and for the future the application can be studies in other contexts encompassing different knowledge areas of Project Management and also the QA activities not included in the scope of the project addressed in this work. To validate the method it is necessary to apply it in different types of projects in different kinds of organizations. Inside software development it also could be applied to help the development team with those levels of testing to be addressed and the cycles of testing to be performed. In addition, some combination and integration with other management tools can be beneficial to guaranteeing the quality of the data collected. Further studies can be done to select and integrate analytics tools with the method to provide the definition of the dataset for reporting automatization or help with the test management through to track test activities performance considering the workflow established.

7. Conclusion

The present project aimed to develop a method to implement QA activities in agile projects, as result was developed the QA4A method. The promotion information visibility while providing the project progress track and the quality assurance overview was achieved with the method developed. With this in mind, the study presented in this dissertation and all the developments undertaken contributed to this goal. The QA4A provided representation and a structured guide to select QA activities to be performed, encompassing all events, artifacts and roles of the Scrum framework adopted by the company. It helped to define the tasks, process and identify the deliverables required to the specific project assigned. The method provided a better organization to implement QA activities, promoting also clarification and adaptation of processes to be followed according to the needed mechanism to guarantee quality while promoting the continuous improvement of quality through ticket management, defect management and constant testing in due time when it was applicable. Once the most expected outcome was to perform QA tasks specifically to the needs of the assigned project. The approach of QA activity remained in continuous test reviews with focus on continuous improvement and the performance of a great work. The method aims help to evaluate the applicability and continuity of the defined activities and enable the analyses performed work. It provided a greater visibility and organization of the tasks, promoting agility and increasing the operational control of the project. In summary the table below give a brief overview how is the quality management is approached through the application of the developed method.

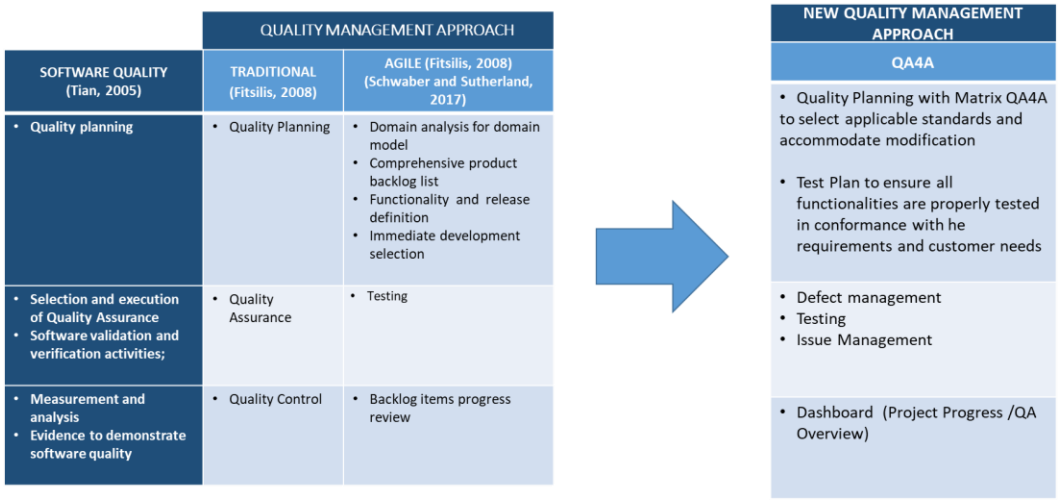


Figure 12: Quality Assurance for Agile Approach

It was possible to leverage QA4A tools to guarantee quality on process execution while promoting the common understand about the activities required to be performed, increasing team's vocabulary and awareness of the project system. The method was used to manage the variety and complexity of agile project that adopted scrum framework (Schwaber and Sutherland, 2017). The constant issue tracking allows the update in a routine basis of the backlog items promoting a constant review and testing culture. It resulted in the increase of the process visibility, dissemination of reports, and document organization in the project using shared networks.

References

AAMIR, M. & KHAN, M. N. A. J. S. 2017. Incorporating quality control activities in scrum in relation to the concept of test backlog. 42**,** 1051-1061.

ABRAHAMSSON, P., SALO, O., RONKAINEN, J. & WARSTA, J. J. A. P. A. 2017. Agile software development methods: Review and analysis.

BACCARINI, D. 1999. The Logical Framework Method for Defining Project Success. 30**,** 25-32.

BADIRU, A. B. 2011. *Project management: Systems, principles, and applications*, CRC Press.

BECK, K., BEEDLE, M., VAN BENNEKUM, A., COCKBURN, A., CUNNINGHAM, W., FOWLER, M., GRENNING, J., HIGHSMITH, J., HUNT, A. & JEFFRIES, R. 2001. Manifesto for agile software development.

BOEHM, B. & TURNER, R. 2003. Using risk to balance agile and plan- driven methods. *Computer*, 36**,** 57-66.

BOEHM, B. & TURNER, R. J. I. S. 2005. Management challenges to implementing agile processes in traditional development organizations. 22**,** 30-39.

BRHEL, M., METH, H., MAEDCHE, A., WERDER, K. J. I. & TECHNOLOGY, S. 2015. Exploring principles of user-centered agile software development: A literature review. 61**,** 163-181.

BRIOSO, X. 2015. Integrating ISO 21500 Guidance on Project Management, Lean Construction and PMBOK. *Procedia Engineering*, 123**,** 76-84.

CHO, J. J. I. I. I. S. 2008. Issues and Challenges of agile software development with SCRUM. 9**,** 188-195.

COCKBURN, A. & HIGHSMITH, J. 2001. Agile software development, the people factor. *Computer*, 34**,** 131-133.

COMES, S. & BERNIKER, L. 2008. Business model innovation. *From strategy to execution.* Springer.

CRISPIN, L. & GREGORY, J. 2009. *Agile testing: A practical guide for testers and agile teams*, Pearson Education.

DOVLEAC, R. & IONICĂ, A. Quality Management techniques embedded in Agile Project Development. MATEC Web of Conferences, 2017. EDP Sciences, 05003.

FITSILIS, P. 2008. Comparing PMBOK and Agile Project Management software development processes. *Advances in Computer and Information Sciences and Engineering.*

GARVIN, D. A. J. S. M. R. 1984. What Does “hltoduct Quality” Really Mean. 25.

GREGOR, S. & HEVNER, A. R. J. M. Q. 2013. Positioning and presenting design science research for maximum impact. 337-355.

HARVEY, L. J. E. B. H. M. B. W. 2006. Understanding quality. 1**,** 1-29.

HENDRICKSON, E. 2015. I Prefer This Over That - Test Obsessed.

HEVNER, A. & CHATTERJEE, S. 2010. Design science research in information systems. *Design research in information systems.* Springer.

HEVNER, A., R, A., MARCH, S., T, S., PARK, PARK, J., RAM & SUDHA 2004. *Design Science in Information Systems Research*.

HEVNER, A. R. J. S. J. O. I. S. 2007. A three cycle view of design science research. 19**,** 4.

HIGHSMITH, J. & COCKBURN, A. J. C. 2001. Agile software development: The business of innovation. 34**,** 120-127.

HIGHSMITH, J. J. S. H. W. A. O. H. H. 2001. History: The agile manifesto.

HUO, M., VERNER, J., ZHU, L. & BABAR, M. A. Software quality and agile methods. Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004. COMPSAC 2004., 2004. IEEE, 520-525.

IGAKI, H., FUKUYASU, N., SAIKI, S., MATSUMOTO, S. & KUSUMOTO, S. 2014. Quantitative assessment with using ticket driven development for teaching scrum framework. *Companion Proceedings of the 36th International Conference on Software Engineering - ICSE Companion 2014*.

ISO, I. J. I. O. F. S. 2012. 21500: 2012 Guidance on project management.

KAYES, I., SARKER, M., CHAKARESKI, J. J. I. I. S. & ENGINEERING, S. 2016. Product backlog rating: a case study on measuring test quality in scrum. 12**,** 303-317.

KNIBERG, H. & SKARIN, M. 2010. *Kanban and Scrum-making the most of both*, Lulu. com.

LACEY, M. 2012. *The scrum field guide: Practical advice for your first year*, Addison-Wesley Professional.

LADAS, C. 2009. *Scrumban-essays on kanban systems for lean software development*, Lulu. com.

LEE, S. & YONG, H.-S. 2009. Distributed agile: project management in a global environment. *Empirical Software Engineering*, 15**,** 204-217.

LEON, A. & KOCH, A. S. 2004. *Agile software development evaluating the methods for your organization*, Artech House, Inc.

LOSADA, B., URRETAVIZCAYA, M., LÓPEZ-GIL, J.-M. & FERNÁNDEZ-CASTRO, I. 2012. Combining InterMod agile methodology with usability engineering in a mobile application development. *Proceedings of the 13th International Conference on Interacción Persona-Ordenador - INTERACCION '12*.

LUCENA, P., BRAZ, A., CHICORIA, A. & TIZZEI, L. IBM Design Thinking Software Development Framework. Brazilian Workshop on Agile Methods, 2016. Springer, 98-109.

MEGAHED, S. & RAMADAN, N. 2016. Requirements Engineering in Scrum Framework. *International Journal of Computer Applications*, 149**,** 24-29.

MILLER, L. Case study of customer input for a successful product. Agile Development Conference (ADC'05), 2005. IEEE, 225-234.

MOHAMED, I. & RAMADAN, N. 2016. A Security Testing Framework for Scrum based Projects. *International Journal of Computer Applications*, 138**,** 12-17.

MOSSALAM, A. 2019. Projects' issue management. *HBRC Journal*, 14**,** 400-407.

MUNNS, A. K. & BJEIRMI, B. F. J. I. J. O. P. M. 1996. The role of project management in achieving project success. 14**,** 81-87.

NEAVE, H. R. J. J. O. T. R. S. S. S. D. 1987. Deming's 14 points for management: framework for success. 36**,** 561-570.

PEFFERS, K., TUUNANEN, T., ROTHENBERGER, M. A. & CHATTERJEE, S. J. J. O. M. I. S. 2007. A design science research methodology for information systems research. 24**,** 45-77.

PULEIO, M. How not to do agile testing. AGILE 2006 (AGILE'06), 23-28 July 2006 2006. 7 pp.-314.

RUBIN, K. S. 2012. *Essential Scrum: a practical guide to the most popular agile process*, Addison-Wesley.

SCHWABER, K. 2004. *Agile project management with Scrum*, Microsoft press.

SCHWABER, K. & SUTHERLAND, J. 2017. The Scrum GuideTM. The definitive guide to scrum: The rules of the game. November 2017.

SMITH, M. L., ERWIN, J. & DIAFERIO, S. Role & responsibility charting (RACI). Project Management Forum (PMForum), 2005. 5.

STOICA, M., MIRCEA, M. & GHILIC-MICU, B. 2013. Software Development: Agile vs. Traditional. *Informatica Economica,* 17**,** 64-76.

SUSSMAN, M. 2012. Metis Project Test Strategy. *Msg Life Group*.

SUTHERLAND, J., SCHOONHEIM, G., RUSTENBURG, E. & RIJK, M. Fully distributed scrum: The secret sauce for hyperproductive offshored development teams. Agile 2008 Conference, 2008. IEEE, 339-344.

SY, D. J. J. O. U. S. 2007. Adapting usability investigations for agile user-centered design. 2**,** 112-132.

TIAN, J. 2005. *Software quality engineering: testing, quality assurance, and quantifiable improvement*, John Wiley & Sons.

TOMANEK, M., CERMAK, R. & SMUTNY, Z. J. A. P. A. 2015. A conceptual framework for web development projects based on project management and agile development principles.

VON ALAN, R. H., MARCH, S. T., PARK, J. & RAM, S. J. M. Q. 2004. Design science in information systems research. 28**,** 75-105.

WILLIAMS, L. & COCKBURN, A. J. I. C. 2003. Agile software development: it's about feedback and change. 36**,** 39-43.

WINTER, R. 2017. Design science research in Europe. *European Journal of Information Systems,* 17**,** 470-475.

APPENDIX A: Online Survey Question Guide

Online Survey Question Guide

Question 1:

Please, give at least 3 examples of the most common problems you have when starting a new project?

Question 2:

Please, give at least 3 examples of the most common problems you have during the project's progress.

Question 3:

What do you understand per Software Quality?

Question 4:

How do you think you can control Software Quality?

Question 5:

How would you make sure that the same bug is not introduced again?

Question 6:

In your opinion why JIRA is a good tool?

Question 7:

Do you think you have the appropriate training to use JIRA?

Question 8:

What are the downsides using JIRA to track your work?

Question 9:

Please, define the success of a solution being delivered?

APPENDIX B: Online Survey Results

Question 1

Please, give at least 3 examples of the most common problems you have when starting a new project?

#	Job Title	Project Role	Answer
1	Business Analyst	Business Analyst	Identify the main stakeholders; have a common understanding of the process to follow; assure a validation of the requirements
2	Customer Service - Support	Business Analyst	Lack of concrete requirements, impossible requirements, difficulty/delay in the communication msg life and client. (It can take weeks to exchange emails to realize what the client wants, or even for them to realize what they really need. Besides the clear time of decision making between what is proposed and the time when it is effectively to move forward).
3	Software Developer	Developer	Don't know where to start, putting together all the information and trying to figure things out.
4	Software Engineer	Developer	Need of architectural changes, UI construction and uncertainty of requirements
5	Developer	Developer	new clear requirements, short timeline, miss communication
6	Software Engineer	Developer	Software requirements not defined in detail New Development Process leads to some confusion Team members communication
7	Data Analyst	Architecture guardian	incomplete information and incomplete explanations
8	Product Modeler, Customer Service Attendee, Mathematician (ordered)	Modeling, Configuration & Support, Data Intelligence Team Implementation (respectively)	Lack of information about costumer's expectations and needs (sometimes, missing customer info itself - field of work, business behavior, provenience, etc.) as well as business-functional specifications (instead of functional only). Uncertainty of requirements and nonexistence of entire process in which we will interfere - or start building - overview. Lack of structured and well-informed planning defined priory.
9	Software Engineer	Team Leader	Requirements uncertainty, project understanding, getting the team up to speed

Question 2

Please, give at least 3 examples of the most common problems you have during the project's progress.

#	Job Title	Project Role	Answer
1	Business Analyst	Business Analyst	Predict the remaining effort needed; manage outdated requirements; communication between all the stakeholders.
2	Customer Service – Support	Business Analyst	Difficulty in understanding the client's problems, especially in clients who have no minimum knowledge of IT. Delays in the tests that customers have to make and consequently delays in passing on the production of the developments. It takes time for customers to analyze proposals. Some lack of communication between the team and some developments take longer than they should.
3	Software Developer	Developer	Communication with the client, vague information, figuring out priorities
4	Software Engineer	Developer	Changes breaking existing features, feature development wrongly estimated, changes in team members
5	Developer	Developer	timeless gets shorter, requirements get bigger, miss communication
6	Software Engineer	Developer	Tight deadlines (deliveries)
7	Data Analyst	Architecture guardian	wait for answers
8	Product Modeler, Customer Service Attendee, Mathematician (ordered)	Modeling, Configuration & Support, Data Intelligence Team Implementation (respectively)	Change of planning due to lack of apriori planning. Too short delivery dates. Uncertainty of requirements inclusively during implementations.
9	Software Engineer	Team Leader	changing requirements, aggressive deadlines, communication between stakeholders

Question 3

What do you understand per Software Quality?

#	Job Title	Project Role	Answer
1	Business Analyst	Business Analyst	Proximity between the delivered solution and the agreed requirements
2	Customer Service – Support	Business Analyst	All the necessary processes to guarantee the quality of the product and its development.
3	Software Developer	Developer	Better Software Quality = Program with better efficiency
4	Software Engineer	Developer	If the software's features meet the goals of the application while having the least amount of bugs as possible while having a proportional level of performance according to what the software needs to do
5	Developer	Developer	represent the quality and performance of the solution overall
6	Software Engineer	Developer	According with the customer requirements test the software solution to validate if there are discrepancies between the software solution and the customer needs.
7	Data Analyst	Architecture guardian	quality of what it's implemented
8	Product Modeler, Customer Service Attendee, Mathematician (ordered)	Modeling, Configuration & Support, Data Intelligence Team Implementation (respectively)	Quick-responsive, with structured and commented scripting, with non-outdated technologies. A good-looking software with all the desired customer's features and some others we believe their business would need.
9	Software Engineer	Team Leader	fitness for purpose

Question 4

How do you think you can control Software Quality?

#	Job Title	Project Role	Answer
1	Business Analyst	Business Analyst	customer engagement in early stages; test automation.
2	Customer Service – Support	Business Analyst	Tests, metrics, metrics analysis and measurements to prevent repetition of errors.
3	Software Developer	Developer	By testing what is the better solution for a problem
4	Software Engineer	Developer	Having simple daily tasks like cross test and code review helps a lot
5	Developer	Developer	with qa process well defined and testing scenarios properly set
6	Software Engineer	Developer	Have a deep knowledge over the different software layers and testing the software solution before making it available for QAs.
7	Data Analyst	Architecture guardian	good results
8	Product Modeler, Customer Service Attendee, Mathematician (ordered)	Modeling, Configuration & Support, Data Intelligence Team Implementation (respectively)	By doing a lot of tests, by verifying with the BA team and requirements writers if the available features are as the customer said that wanted them, way before delivering.
9	Software Engineer	Team Leader	you cannot control anything. you can put some activities and process in place to minimize potential problems, mitigate risks and maximize project success probabilities

Question 5

How would you make sure that the same bug is not introduced again?

#	Job Title	Project Role	Answer
1	Business Analyst	Business Analyst	Run automatic tests.
2	Customer Service – Support	Business Analyst	organisation, rigour in development and documentation
3	Software Developer	Developer	Communicating with my team
4	Software Engineer	Developer	Code review and cross test help
5	Developer	Developer	create automated testing for that scenario when possible or document the scenario so it can be track over time
6	Software Engineer	Developer	It is impossible!
7	Data Analyst	Architecture guardian	if we resolve the problems with efficiency they doesn't appear anymore
8	Product Modeler, Customer Service Attendee, Mathematician (ordered)	Modeling, Configuration & Support, Data Intelligence Team Implementation (respectively)	With a good understanding on how the bug happened and in which circumstances something similar could happen, avoiding a fix of a bug solely due to a very specific situation. This approach would demand, of course, more time available to QA team, BA team and implementation team.
9	Software Engineer	Team Leader	you don't. You can minimize that probability by creating some kind of automated test.

Question 6

In your opinion why JIRA is a good tool?

#	Job Title	Project Role	Answer
1	Business Analyst	Business Analyst	Yes.
2	Customer Service Support	Business Analyst	JIRA allows you to make a ticket management that fits perfectly to the work in the company. It is also a useful tool in communication with the client.
3	Software Developer	Developer	We can see what anyone is doing / has done at anytime
4	Software Engineer	Developer	Besides being a database for all the tasks it helps managing the day-to-day tasks and allows the team to know the state of the project
5	Developer	Developer	yes
6	Software Engineer	Developer	N/A
7	Data Analyst	Architecture guardian	it's a good way to define tasks
8	Product Modeler, Customer Service Attendee, Mathematician (ordered)	Modeling, Configuration & Support, Data Intelligence Team Implementation (respectively)	I actually don't know another one and wouldn't say this is very good because each project has its own cycle and sometimes it is defined a very tricky and complex process for a very simple way to work. Anyhow, it was the chosen tool due to the company's Agile methodology.
9	Software Engineer	Team Leader	yes

Question 7

Do you think you have the appropriate training to use JIRA?

#	Job Title	Project Role	Answer
1	Business Analyst	Business Analyst	Yes.
2	Customer Service – Support	Business Analyst	No. I feel that for those who do not have experience with this type of issue management tools, it is not at all intuitive in the beginning.
3	Software Developer	Developer	Not really, but i do my best!
4	Software Engineer	Developer	Yes
5	Developer	Developer	No
6	Software Engineer	Developer	N/A
7	Data Analyst	Architecture guardian	no
8	Product Modeler, Customer Service Attendee, Mathematician (ordered)	Modeling, Configuration & Support, Data Intelligence Team Implementation (respectively)	No.
9	Software Engineer	Team Leader	it could be better

Question 8

What are the downsides using JIRA to track your work?

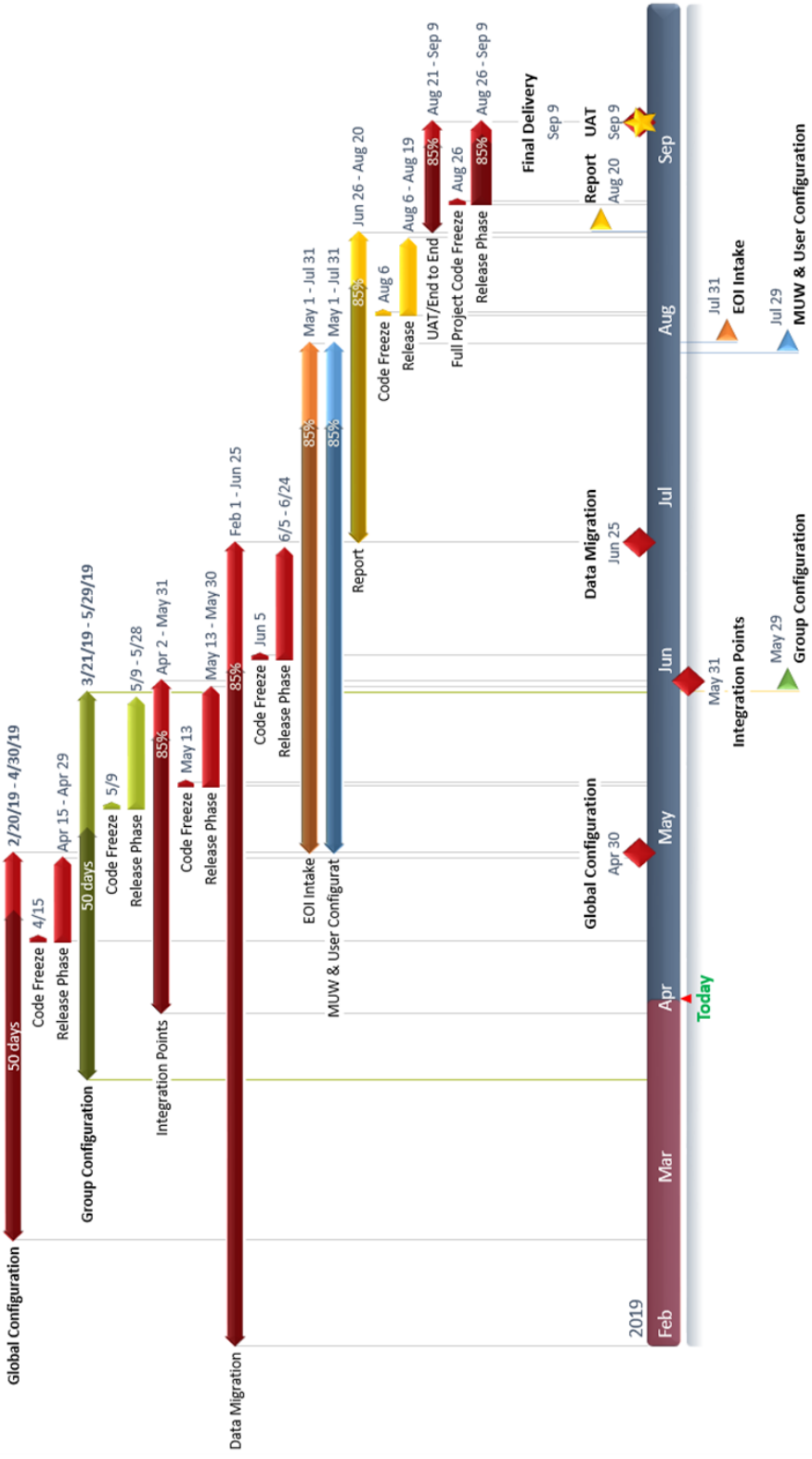
#	Job Title	Project Role	Answer
1	Business Analyst	Business Analyst	To be helpful for a project it must be very well structures; sometimes is not used in the same way by everybody; there are not standards defined in order to allow different projects to be compared.
2	Customer Service – Support	Business Analyst	I use a more "simple" JIRA with customer users included in the workflow. It is very difficult to "teach/educate" a client to use. However, when customers use it, they are usually able to fill in the information well. However, it is always necessary to have a super vision.
3	Software Developer	Developer	People may feel controlled
4	Software Engineer	Developer	Having to log work on JIRA besides logging it to the ERP
5	Developer	Developer	the ticket workflow need to be well set otherwise it will cause bottleneck
6	Software Engineer	Developer	N/A
7	Data Analyst	Architecture guardian	when people don't know how use JIRA
8	Product Modeler, Customer Service Attendee, Mathematician (ordered)	Modeling, Configuration & Support, Data Intelligence Team Implementation (respectively)	Not viable for certain types of development (investigation projects with largest user stories). It is a Project Management tool and it is not friendly regarding Product Management Does not support multi-product backlogs, capacity planning, trade offs, road maps. Each project has features and functionalities impacting on the main Product of the company. Each project is handled regardless of the product, which is shared by almost all projects (in our company situation). This leads, in a long term, to many different and parallel ways to develop similar functionalities.
9	Software Engineer	Team Leader	we are using a jira instance managed by the US, so we can't configure it/tailor it to our specific needs.

Question 9

Please, define the success of a solution being delivered?

#	Job Title	Project Role	Answer
1	Business Analyst	Business Analyst	The delivered solution meets the customer requirements; parts of the solution can be reused in other projects; it is possible to analyze the project (effort, added value, areas to improve);
2	Customer Service – Support	Business Analyst	Sucesso na entrega basicamente é definido como zero problemas e tudo a funcionar. Normalmente acontece quando o desenvolvimento é rigoroso, organizado e os testes são massivos.
3	Software Developer	Developer	If the solutions is what the client had in mind
4	Software Engineer	Developer	The same as project success
5	Developer	Developer	a good solution that satisfies the client long term
6	Software Engineer	Developer	N/A
7	Data Analyst	Architecture guardian	complete the plan
8	Product Modeler, Customer Service Attendee, Mathematician (ordered)	Modeling, Configuration & Support, Data Intelligence Team Implementation (respectively)	The successful delivery of a solution is defined by the accurate, useful, not complicated and expected behavior of all functionalities and features integrated along a business process, helping that process to be more efficient.
9	Software Engineer	Team Leader	All stakeholders satisfaction

APPENDIX C: Project Timeline



APPENDIX D: Msg Life Iberia Implementation Model Standard

VOYA PROJECT ROLES		
Project Role	**Description**	**Msg Life Group Units**
Project Manager	• Guarantees project scope, timelines and budget are aligned with service level agreement (SLAs)	FJA
Business Analyst	• Transforms Business Requirements into Project focused functional Specifications according to costumer needs	FJA
Project Facilitator	• Manages the project internally in Iberia, when the customer facing responsibilities are performed by other unit/company • Responsible for negotiating budget, plans, work process, mitigate risks and protect the implementation team while guaranteeing successful project delivery	FJA & Iberia

Implementation Lead	• In charge of the whole implementation:
	  • Project setup, customer specific developments, process parametrization, catalog enrichment
	  • Integration development (not impacting core msg.Sales)
	  • Builds the bridge between Business Analyst, Development Team, QA, Project Facilitators and customer project manager.
		FJA & Iberia

Customization Team	• Product parametrization and modeling • UX/UI (core msg.Sales) • Documents and reports	Iberia
msg.Sales Project Development Team	• Architecture Guardian Identifies topics which might impact other projects, S&S product vision and best practices • Representative per area: Front-End, Back-End, DevOPS sharing knowledge and solutions between the several teams • QA Guarantees whatever is produced is bullet proof, and "Done criteria" is respected	Iberia
Product Management Team	• Works with Project Facilitator and Business Analyst transforming project requirements into product requirements; • Core Framework road mapping, governance, prioritization and integration between project and product • Guarantees proper transfer of product requirements to each project development team • Manages msg.Sales product backlog and priorities • UX/UI specifications according customer needs	FJA & Iberia
Agile Coach	• Facilitates the work process in each project development team • Removes roadblocks • Helps to identify areas to improve • Pursues continuous improvement mindset	FJA & Iberia

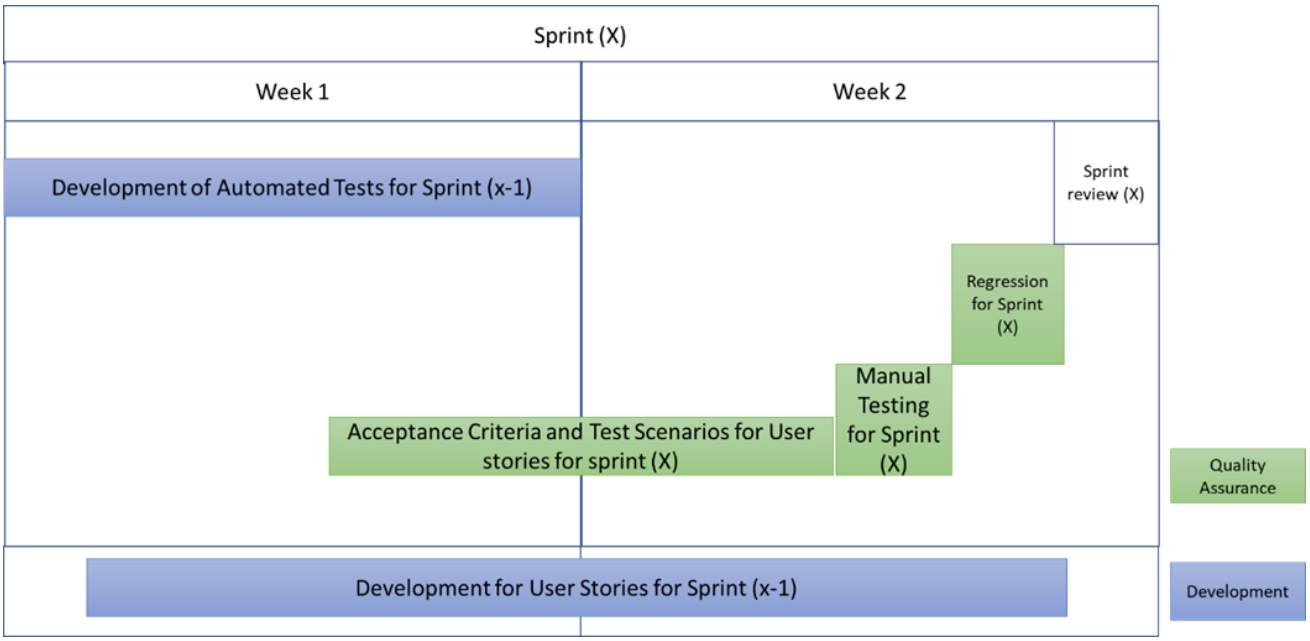
APPENDIX E: Quality Assurance for Agile Matrix

QA Activities

Project Configuration:	Iberia +FJA
QA Link :	BA
QA Objectives:	Voya is responsible for all testing activities. FJA will provide guidance to Voya Quality team how best perform the test per FSD sign-off. FJA's QA will use this this specification for regression testing on each deliverable
Test type (Cross, smoke, Stress, QA):	QA: The tests will be built not on use Cases, but rather the more granular user stories, which allows to complete testing on a few user stories per Sprint.
Nightly Enviroment:	http://ny-vm-23:7080/voya/#4015 **User:** stark **Password:** Password 1
QA Enviroment:	

Phase	Deliverable / Task	QA Iberia	Dev Iberia	Auto (Y/N)	Team Iberia	FJA	VOYA	Comments
Project Reports								
	Timeline	RA	N/A	N/A	RA	R	ACI	Reviewed as project update by FJA
	Progress Report	RA	N/A	N/A	RA	N/A	N/A	Project tracking will take place in JIRA with excell support
	Status Report	RA	N/A	N/A	RA	R	ACI	The completion of these will be reviewed weekly in Stand Up meetings and bi-weekly in Sprint Showcase/Planning meetings
Test Levels								
	Unit Test	N/A	N/A	N/A		I	RA	
	Manual User Story Test	RA	N/A	N/A	RA	N/A	N/A	
	System Integration Test (SIT)	N/A	N/A	N/A		I	RA	
	Regression Test	N/A	N/A	N/A		IC	RA	1. FJA will develop regression tests to ensure functionality previously delivered to Voya is working as designed after feature enhancements are complete. 2. These tests will be executed by FJA in FJA's QA environment.
	User Acceptance Test (UAT)	N/A	N/A	N/A		I	RA	1. Voya will develop User Acceptance Test (UAT) Cases designed to ensure delivered functionality meets the Voya's functional needs. 2. These tests will be executed by Voya in Voya's UAT environment
	Nonfunctional (Performance)	RA	N/A	N/A	RA	RA	I	
Deliverables								
	Test Planning	N/A	N/A	N/A		IC	RA	1. All testing will be based on functionality defined in approved Functional Specification Documentation. 2. General Entry/Exit criteria will be agreed on 3. FJA and Voya will work together to define the test management plan.
	Test Scenarios	N/A	N/A	N/A		N/A	N/A	
	Test Cases/Test Scripts	N/A	N/A	N/A		N/A	N/A	
	Requirements Traceability (to Test Cases)	N/A	N/A	N/A		N/A	N/A	
	Test Execution; Metrics, Status, Defect	RA	N/A	N/A	RA	N/A	N/A	
	Test Evaluation Summary Document	RA	N/A	N/A	RA	N/A	N/A	
Manage/Control								
	Deployment	N/A	N/A	N/A		CI	RA	1. Deployment to Test, Training and Production
	Enviroment	N/A	N/A	N/A		N/A	N/A	1. FJA will create and maintain a Quality Assurance (QA) Test environment within FJA's infrastructure for testing before delivery to Voya. 2. Voya will create and maintain a UAT environment for integrated testing of each delivery from FJA 3. This environment must be prior to the start of UAT and maintained throughout the duration of the project.
	Test Cycles	N/A	N/A	N/A		N/A	N/A	1. Functional Testing 1.1 FJA will execute predefined functional tests for new features before each delivery. 1.2 Duration: 1 week 2. Regression Testing 2.1 FJA will execute regression tests before each delivery. 2.2 Duration: 1 week (in parallel with Functional Testing) 2.3 The regression tests executed for each release will include: 2.3.1 A based set of regression testing defined for every release 2.3.2 Additional regression test(s) defined by FJA focused on functional areas impacted by code changes in the release. 3. UAT 3.1 Voya will execute UAT tests upon delivery of each release by FJA 3.2 Duration: 1 week 3.3 UAT will sign off on delivery upon completion of UAT test cycle. 3.4 Sign off document must be signed and provided to FJA
	Change Request Log	N/A	N/A	N/A		R	ACI	If defects are identified Voya and FJA will determine if it is part of the original scope or a change control is required.
	Test Acceptance Criteria	N/A	N/A	N/A		CI	RA	
	Defect logs	RA	N/A	N/A	RA	R	ACI	**Iberia:** **FJA:** For each defect raised by Voya a classification (Blocker/Critical, High, Medium, Trivial etc.) will be assigned as mutually agreed by Voya and FJA. Voya will have 30 days from final delivery to validate functionality. 1. Review defects 2. Ensure reported defects contain appropriate information for FJA to triage and correct 3. FJA and Voya review of defects and mutually agree on priority and target release delivery for each defect

APPENDIX F: Msg Test Cycles



APPENDIX G: Project Report Scope

Voya Report Scope

It is necessary to gather information weekly from

- FJA/Iberia Project Timeline
- JIRA: Export Issues Cvs format

What information I want to see?

1. Times estimated vs remaining time for the whole project
2. Work progress
   a. Tickets
      i. Total of closed tickets / month
      ii. Total of tickets/ area
      iii. Total/Status
   b. Team
      i. Remaining work/ month
      ii. Available development team time/month
      iii. Planned development time / month

What information should be available?

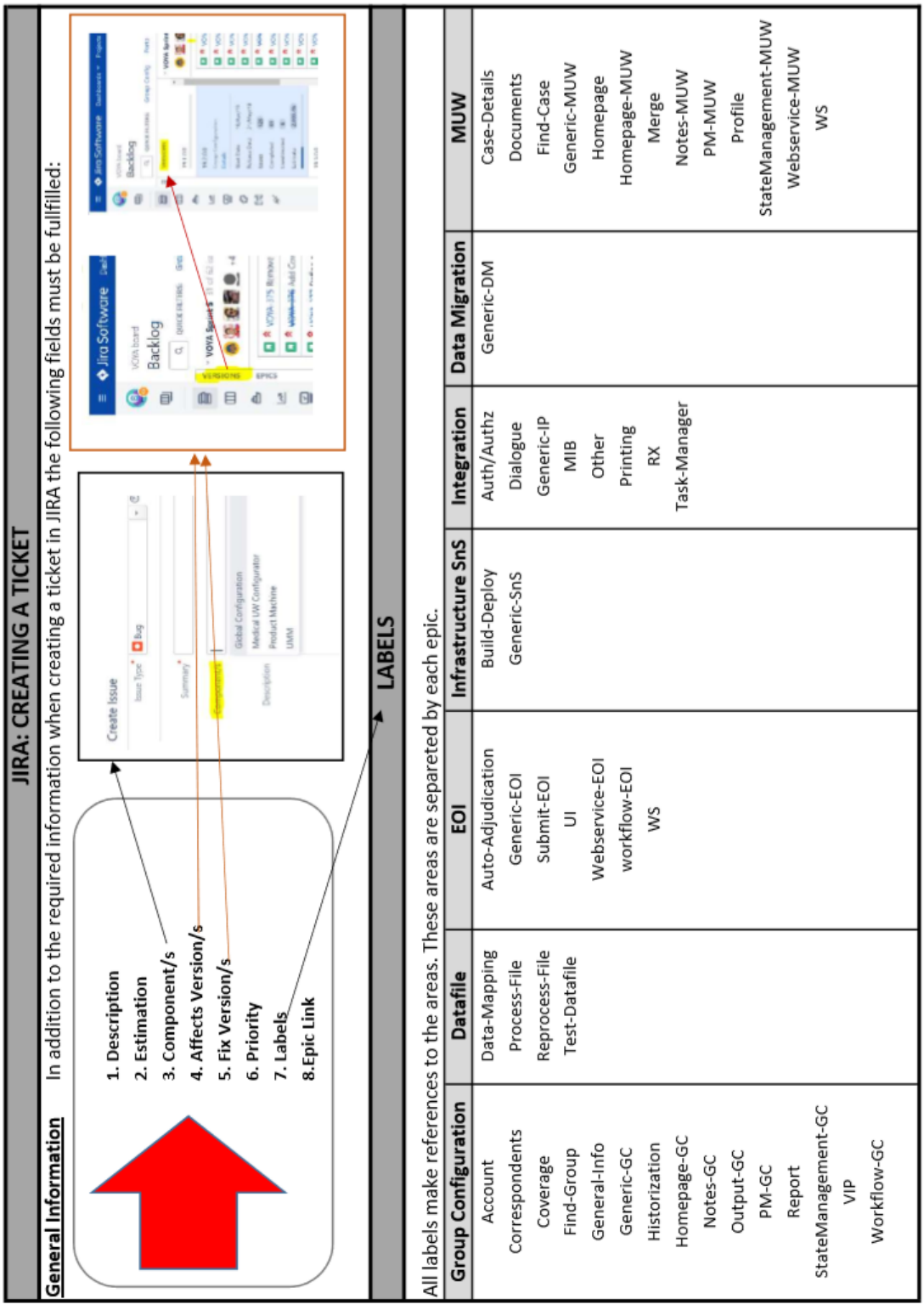
How to create the Graphs

3. **JIRA**
   1. Search Issue with the desired columns
   2. Export CVS file with selected Fields
4. **Excel**
   1. Import CVS file to an empty worksheet
   2. Confirm if the field “remaining time” is equal to zero for all issues that have status “closed” or “done”

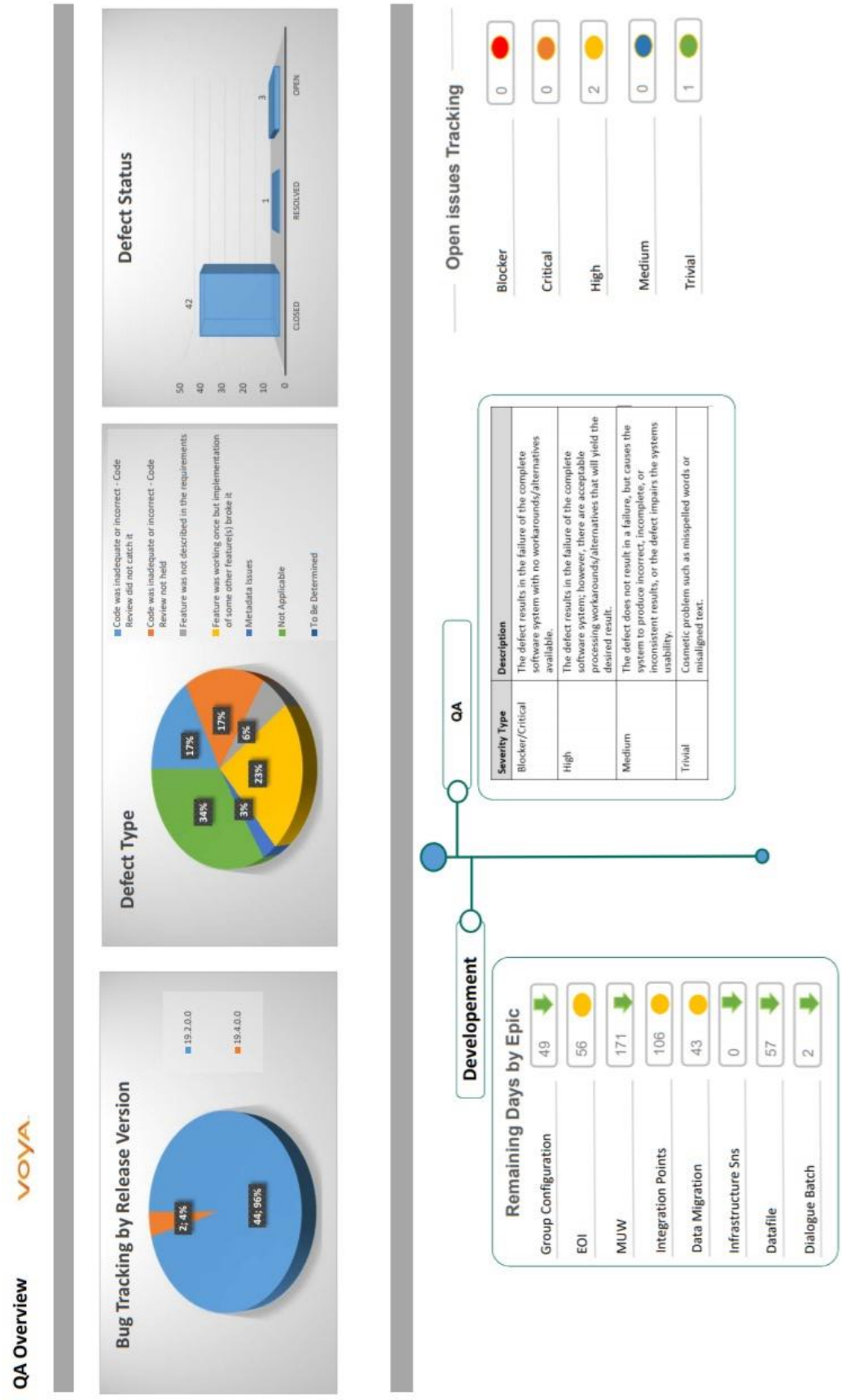
      If yes» proceed to pivot table creation

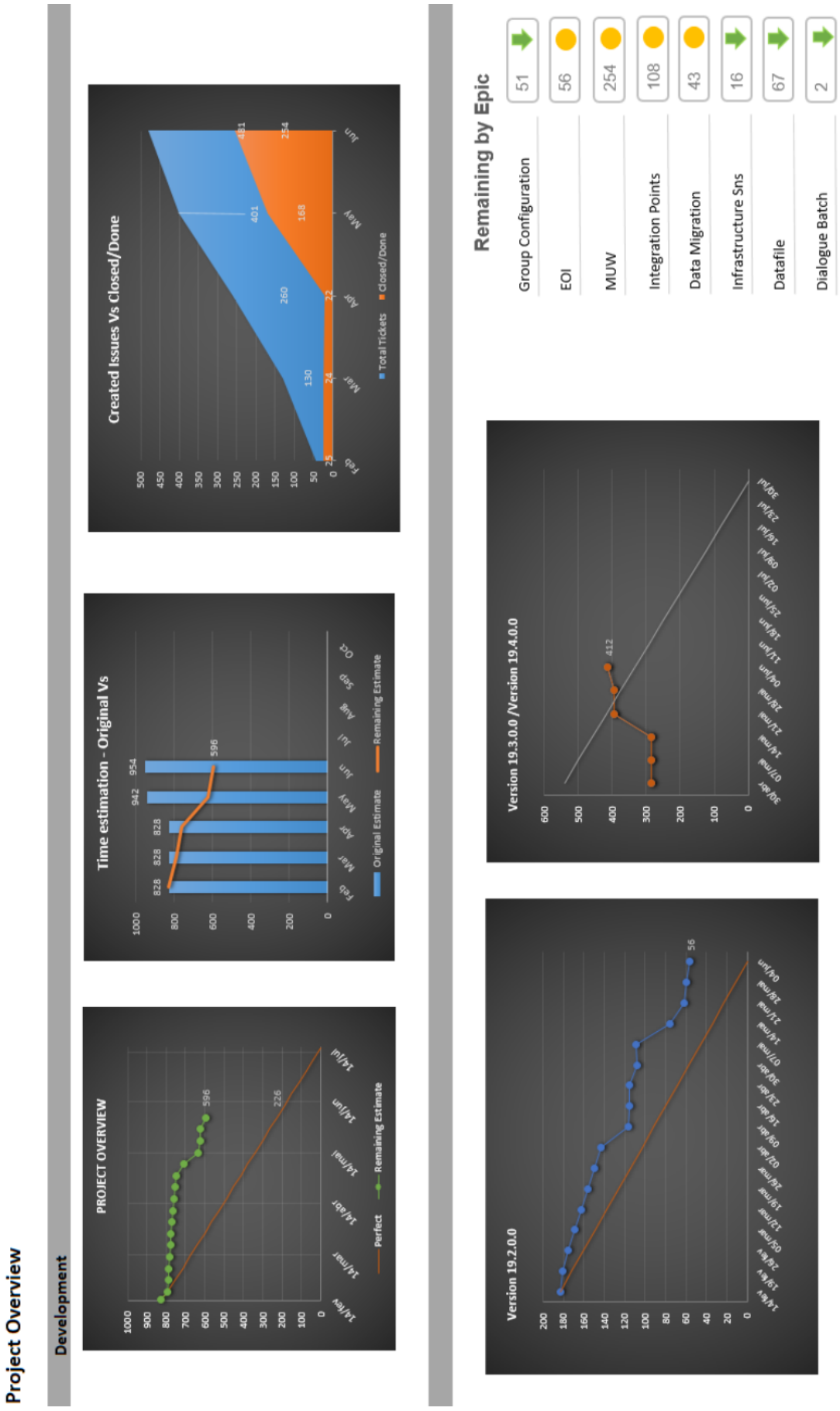
      If no» replace all field values to zero
   3. Create Pivot Table
   4. Update Graphs

APPENDIX H: Ticket Creation Guide



APPENDIX I: Project Progress and QA Dashboards




Project Overview
Development
PROJECT OVERVIEW
Perfect
Remaining Estimate
Time estimation - Original Vs
Original Estimate
Remaining Estimate
Created Issues Vs Closed/Done
Total Tickets
Closed/Done
Version 19.2.0.0
Version 19.3.0.0 /Version 19.4.0.0
Remaining by Epic
Group Configuration 51
EOI 56
MUW 254
Integration Points 108
Data Migration 43
Infrastructure Sns 16
Datafile 67
Dialogue Batch 2

APPENDIX J: Test Documentation

Note: Precondition and test expected results columns were deleted from this table due to project confidentiality

TOPIC	TEST	DEVELOPED	TEST DATE	ISSUES
	Authentication & Authorization	YES		
My Queue Homepage	Verify the Homepage defaulted to "My Queue" view	YES	03/06/2019	
My Queue Homepage	Verify the row fields in "My Queue"	YES	03/06/2019	
My Queue Homepage	Verify the Group status	YES	03/06/2019	
My Queue Homepage	Search Bar	YES	03/06/2019	
My Queue Homepage	Last modified date column	YES	03/06/2019	
My Queue Homepage	Last Modified date column for submitted groups		03/06/2019	
My Queue Homepage	VIP Groups	NO		
My Queue Homepage	Click “+Add Group” link to add a new Group	YES	04/06/2019	
My Queue Homepage	Clicks the Gear (Edit) icon on a row	YES	04/06/2019	
My Queue Homepage	Clicks the ‘+’ sign in the Accounts column on a specific row	YES	03/06/2019	
My Queue Homepage	Verify user with inappropriate rights cannot view or edit VIP group	NO		
Find group	User clicks on Find Group quick link	YES	03/06/2019	
Find group	The user enters Group number search criteria in the Find box	YES	03/06/2019	
Find group	The user enters Group Name search criteria in the Find box	YES	03/06/2019	
Find group	User Clicks the Magnifying Glass icon to Find Group	YES	03/06/2019	
Find group	User checks “Show Group History” Checkbox	YES	04/06/2019	
Find group	Verify Actions buttons on Archived Status	YES	04/06/2019	
Common Actions	Settings button in every screen	NO		
Common Actions	Reload button in every screen	NO	04/06/2019	
Find group	User clicks the ‘+’ sign in the Actions column for a specific row in the table to display the Account(s) associated with the group	YES	04/06/2019	
Find group	Select VIP Groups for Visualization	NO		
Find group	Clicks the “View” link in the table row		03/06/2019	
Common Action	Submit	YES	03/06/2019	**VOYA-471**
Common Action	Clicks “Update” Command button	YES	04/06/2019	
Common Action	“Cancel” on the bottom of any screen	YES	04/06/2019	
Common Action	"Delete” on the bottom of any screen	YES	04/06/2019	
General Information	General Information tab is displayed as landing screen when adding a group	YES	04/06/2019	
General Information	Enters the Group Number	YES	03/06/2019	
General information	verify general information fields	YES		VOYA-617
General information	Verify Invite hold values set up		04/06/2019	

TOPIC	TEST	IMPL.	TEST DATE	ISSUES
General information	Verify termination Date Setup	NO		
General information	Change the flag of the EOI Options to YES	NO		
General information	Select EOI options	YES	04/06/2019	
General information	Verify Data file type	YES	04/06/2019	
General information	Verify Email Invite	YES	04/06/2019	
General information	Verify Send URL	YES	04/06/2019	
General Information	Invite Hold Message	NO		
General Information	Situs state edition Message	NO		
Notes	Clicks the “Add Notes” command button	YES	04/06/2019	
Notes	Enters form text verification	YES	04/06/2019	
Notes	Associate note to the group	YES	04/06/2019	
Notes	Verify group name and account name (header)	YES	04/06/2019	
Notes	Cancel the note by clicking the “X” on the popup form	YES	04/06/2019	
Form Notes	Add Form Notes	YES	04/06/2019	
Group Account	Click “+Add Account”	YES	04/06/2019	
Group Account	Verify Validation of the Group Account. Click Add button on the popup	YES	04/06/2019	
Group Account	Verify Generic Web URL	NO		VOYA-471
Group Account	Reactivate group account	YES	04/06/2019	
Group Account	Edit group account	YES	04/06/2019	
Group Account	Group Status- State Management	YES	04/06/2019	
Notes	Clicks the “+Note” command on the row for the specific Account	YES	04/06/2019	
Notes	User “check the box" to set the archive status of the note	YES	04/06/2019	
Coverage	Clicks “+Add Coverage” link	YES	04/06/2019	
Coverage	Selects an Account from the Dropdown	YES	04/06/2019	
Coverage	Narrow the list of coverages by entering text into the Find field at the top of the table	YES	04/06/2019	
Coverage	Selects a Coverage by checking the box the “Sel” column next to the Coverage	YES	04/06/2019	
Coverage	Selects value for “Applies To” columns	YES	04/06/2019	
Coverage	Selects Amount Type	YES	04/06/2019	
Coverage	Clicks “Add” command button on coverage popup	YES	04/06/2019	
Coverage	Edit coverage - Select the row with the appropriate Account and Coverage to Edit and click the Gear Icon to edit	YES	04/06/2019	
Coverage	Inactivate/Remove Coverage from Account	YES	04/06/2019	
Coverage	Clicks the “Update” command	YES	04/06/2019	
Coverage	Verify the existence of at least one Coverage for each Account for the Group.	NO	04/06/2019	

TOPIC	TEST	IMPL.	TEST DATE	ISSUES
Correspondents	Selects the Account to which the Correspondent will be added	YES	04/06/2019	
Correspondents	Search correspondent in the popup	YES	04/06/2019	
Correspondents	Selects and add a name from the correspondent list clicking on the row	YES	04/06/2019	
Correspondents	Selects the values for the Account-Specific dropdowns	YES	04/06/2019	
Correspondents	Remove correspondent - Selects the Account from which the Correspondent will be inactivated and clicks the Trashcan icon	YES	04/06/2019	
Correspondents	Verify at least one correspondent per account	YES	04/06/2019	

APPENDIX L: Test Workflow defined with QA4A