MGI

MESTRADO
Gestão de Informação

Standardization of project management processes

An inside view on changes to achieve efficiency

Miguel Augusto Martins Pereira

Work Project presented as partial requirement to obtain the Master's degree in Information Management

UNIVERSIDADE NOVA
ISEGI Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

STANDARIZATION OF PROJECT MANAGEMENT PROCESSES

by

Miguel Augusto Martins Pereira

Work Project presented as partial requirement to obtain the Master's degree in Information Management, specialization in Management of Information Systems and Technologies

Coordinator:

Aleš Popovic

November 2013

DEDICATIONS

To the ones who are not among us anymore…

ACKNOWLEDGEMENTS

There are a number of persons that I would like to acknowledge for their help in writing this work. Without them, this work would not be possible.

Foremost, I would like to express my gratitude to my advisor, Aleš Popovic for his guidance and assistance in defining the guidelines of this work that helped steer it for a timely conclusion.

Secondly, I also would like to acknowledge my deepest and sincere appreciation to the support given by Susana Oliveira, not only for the endless nights spent proof-reading this work and the much appreciated input in every step of this work, but also for the backup given on every aspect of normal life, helping in making this work what it is today.

I also wish to thank my family, specifically my mother, father, grandparents and siblings, but also all the other family members that help me in this long period necessary to obtain the master's degree.

A special recognition to my co-workers at DRI that were kind to given their time in answering my questions, namely: Claúdio Calvão, Paulo Mártires and Dora Bailão. Also I would like to thank Diogo Cabral and the Melos, Salete and Ricardo for their help, proof-reading and valuable inputs. Likewise, I would like to give a special appreciation to my team-members that gave an important feedback by means of their daily work.

Lastly, I would like to send my appreciation to the friends that read my work in its several stages and gave an important contribute to it, but also to the friends that helped just by listening.

SUMMARY

The primary focus of this work is to provide a framework of standardized project management processes developed through the analysis of the processes of a Portuguese Small and Medium Enterprise (SME), in order to help address cost consuming issues that lead to inefficiency on a company.

After the identification of these issues occurring in the subject company, the objective is to determine the best practices of project management based on a theoretical background, in order to develop a standard in project management activities. To accomplish this, the project management workflow is re-modeled, while keeping measurement activities to ascertain its efficiency and validity.

The final part of this work is a proof of concept on how to integrate these workflows in an existing Information System structure and also it gives a series of guidelines on how to introduce these changes on a company.

KEYWORDS

Software development; Critical success factors (CSF); Project management; Agile; Scrum; Business process management (BPM) best practices; Management of information systems (MIS)

ABSTRACT

Software development is a discipline that is almost as old as the history of computers. With the advent of the Internet and all of its related technologies, software development has been on high demand. But, and especially in SME (small and medium enterprise), this was not accompanied with a comparable effort to develop a set of sustainable and standardized activities of project management, which lead to increasing inefficiencies and costs.

Given the actual economic situation, it makes sense to engage in an effort to reduce said inefficiencies and rising costs. For that end, this work will analyze the current state of software development's project management processes on a Portuguese SME, along with its problems and inefficiencies in an effort to create a standardized model to manage software development, with special attention given to critical success factors in an agile software development environment, while using the best practices in process modeling. This work also aims to create guidelines to correctly integrate these changes in the existing IS structure of a company.

KEYWORDS

Software development; Critical success factors (CSF); Project management; Agile; Scrum; Business process management (BPM) best practices; Management of information systems (MIS)

INDEX

IMAGE INDEX

TABLE INDEX

LIST OF ABBREVIATIONS

BPM: Business Process Management. An approach on managing the business processes of a company to improve their efficiency in order to deliver better value to the client.

BPMN: Business Process Model and Notation. A notation of graphical representation to model a workflow of a process.

CRM: Customer Relationship Management. A series of practices supported by an information technology infrastructure that manages the current and future customers (through the use of commercial leads).

CSF: Critical Success Factor. Necessary step to achieve success in the related endeavor.

EPM: Enterprise Project Management. An information platform to manage and monitor the company's projects.

IS: Information System. An aggregation term that encompasses information technology, people and their processes.

MIS: Management Information System. An information technology system to manage all the components of an information system.

OTE: Original Time Estimate. An estimation given by the PO for a functionality of a project.

PO: Product Owner. In Scrum it is the responsible to ensure that the project delivers the intended product to the client.

RTE: Remaining Time Estimate. An estimation given by the development team to an issue of a story.

SME: Small and Medium Enterprise. A definition of a company following certain parameters like number of employees and turnover or balance sheet total.

1. INTRODUCTION

Ideally, structured project management processes are essential to correctly manage a project, however the management of these processes is typically overlooked in favor of a quicker expedition of a project's product, especially on Small and Medium Enterprise (SME), where resources like time, knowledge and money are not as available as in larger companies.

Nevertheless, for companies in general and SMEs in particular, one way to resist today's economic crisis is trough efficient work, meaning that their internal processes should be properly structured and carefully aligned with their (strategic) goals.

The goal of this work is to provide a framework that standardizes software development project management processes. To do so, an analysis was conducted on a Portuguese SME which featured some problems on the level of project management in order to obtain findings that support the new proposed framework. A focus is given to SMEs since they are the bulk of not only Portugal's enterprise fabric (INE, 2010), but also in the EU (Wymenga, Spanikova, Barker, Konings, & Canton, 2012), and together they constitute the major employer of the global active population and are viewed as a major source of innovation on their fields.

1.1. CONTEXTUALIZATION AND PROBLEM IDENTIFICATION

In order to obtain the basis to formulate a consistent framework of standardized project management processes, an analysis of issues concerning project management was conducted in a Portuguese SME.

The company, DRI, works in the field of software development and its structure can be identified as a flat structure (Lim, Griffiths, & Sambrook, 2010) because there is a low formalization of the business processes, low level of centralization of decision making and there is not a culture of documentation, i.e. inserting the input on the status of a project, or the time spent on each of the tasks of the project. In the beginning, when the company had few workers, this was a manageable situation. As the workers were on the same project, all of the information was (mainly) known by everyone. But since the company is growing in size, the information tends to be scattered, so some adjustments in terms of business processes have to be made, as many of the processes are not modeled or even structured. Additionally, it is difficult to disseminate the information due to a lack of hierarchy on the distribution of information.

With the company's growth, the management decided to establish a common agile project management methodology using the Scrum framework and to use an Enterprise Project Management (EPM) to help with project management activities. Nevertheless, having the tools to do the job is only half the work, as it is still required to determine the correct application of these tools to better benefit from them.

In this way, the following issues were identified:

- There is not a common set of processes to manage the projects. Each project manager, or Product Owner (PO – which is an equivalent role of a project manager in the project management methodology adopted by the company) has its own methods, taking different steps in each project;
- There is not a standard procedure to gather and store the information related with the project. Presently, there are many programs available to be used for that purpose, but in some situations none is used. There is not a central system to store all the pertinent information of a project that is needed to support its development;
- The financial and commercial departments do not have a clear view of the actual status of the projects being worked on, preventing the financial department to issue on time payment requests and the commercial area to manage the expectations of upcoming customers;
- In the operational area, the developers also suffer with the lack of information about the project, because, since they do not have a clear picture they cannot be assertive and comply with deadlines;
- While there is some degree of tracking the time spent on the tasks performed each day against prior estimations, this information is not being used to improve the methods of planning and its estimations.

Due to the lack of an established set of project management processes, a company can suffer from the issues identified above which reflect on the company's performance and make the development process less effective in terms of time and resources.

Using the data obtained from DRI, it is the author's belief that, together with an assertive theoretical background, it is possible to devise a good foundation to build a standardized framework for managing software development projects and thus obtain a set of tools and processes to eliminate or mitigate issues occurring in a company, resulting in a better accountability for the clients and for the company itself.

1.2. Objectives Definition

Many of the issues previously pointed out may be (somewhat) addressed promptly, but that should not be the norm while implementing change. Taking the example of what happened at DRI, in the past, when a problem was found it was resolved straight away, while it was positive on a short term to patch a problem, not thinking and defining a strategy on the long term just postponed the resolution of the problem itself.

As such, the objectives for this work are to:

- Uncover the specific problems related with project management processes that occur in a company;
- Identify and model new project management processes to solve the problems identified;
- Provide methods to allow the managers, which are the main recipients of this work, to follow more closely the status of a project;
- Organize a common structure to store the information related to a project, so it may be available to whoever needs it.

Gathering from all of these points, the overall objective is to build a framework that allows a company to improve their project management processes and provide the tools and relevant information not only to the project managers, but also to all of the participants of a project thus allowing them to be more effective on their work.

Although this work focuses on the use of Scrum it is not its objective to uphold the use of a specific methodology, but to defend a broader approach on implementing agile methodologies.

1.3. The Importance of the Work

By employing the framework proposed in this work, a company can achieve:

- A unified set of project management processes to allow all of the company's managers to manage development project in a coherent fashion and store all the necessary information in a centralized location;
- A clear picture of the initial estimations and the real cost in terms of time used on the project;
- Validation of the work being done in the project through the use of metrics;
- Certainty that the development of products can be made with accountable and documentable quality.

1.4. WORK'S STRUCTURE

The structure of this work is as follow: In the introduction the problems that this work is aimed to solve are identified, as well as the work's objectives and goals. Afterwards the theoretical background section is presented, including the methodology employed and a thorough presentation on the research done for this work. This is followed by the presentation of issues found through a set of interviews to the POs and a group of proposals to solve the issues identified based on the interviews and independent observation. Next it is presented a discussion from a theoretical and practical point of view, together with the limitations of this work. Moreover, some additional recommendations and some notes of possible works in the future are made. At the end, conclusions about this work are drawn together with a deliberation of the path this work proposes for companies.

2. THEORETICAL BACKGROUND

2.1. METHODOLOGY

For this work, various methods of gathering information were used. Firstly observation, since the author works on the subject company of this study, day by day observations prompted several insights that lead to the next method: Interviews. Using these observations, a series of questions was devised (Table 1) and a series of interviews ensued with the POs of projects on the company.

Number	Questions	Objectives
1	Who supplies the information related to new projects?	To determine where the flow of information starts.
2	What is the typical workflow of a project, from the project management point of view?	To determine the current workflow of a project, from each of the POs in order to assess the common tasks.
3	Is every project treated differently, or do they share some common tasks / steps?	To determine if the PO has some structure in his / her process.
4	If yes, can you identify them?	Obtain the structure.
5	Can these common tasks / steps be grouped by type of project or by another category? In positive case, is that grouping currently working?	Determine if there are tasks that can be grouped and determine the criteria to do so.
6	Besides the communication that occurs with the developers, how is the communication with the other departments of the company made?	To determine how the communication, if any, is made.
7	What are the most common difficulties that you typically encounter?	To obtain a list of common problems already identified.
8	Do you have any suggestion to mitigate them?	Try to obtain some solutions that could be used in the final solution.
9	Do you currently compare the estimate times done previously with the actual time spent on project issues?	Try to determine if a comparison is made, and what are the results.

Number	Questions	Objectives
10	Why or why not?	It is expected that the PO does not compare the results. This will help to determine the cause and implement the solution.
11	Do you have any idea of which are the activities being constantly underestimated?	Try to determine if the PO has any idea which activities are more problematic.
12	Is there any parameter defined that allows determining if a project is a success or not?	To determine if the PO have any CSFs for projects.
13	If so, which are they?	Obtain the list of CSF of the PO.
14	If CSFs are defined, is there any idea which are the main problems that lead to a project not being considered a success?	Also to try to determine if there is information on which of the CSFs are the main reasons for a fail project.
15	And is it a problem that occurs frequently? Is anything being done to quench this problem and why?	Determine solutions to the problem, if there are problems in the current Enterprise Project Management (EPM).
16	Were there any CSFs defined for projects? Is it possible to group them?	Further information gathering to assess the status of CSFs of projects.
17	The new[1] EPM is already implemented. What do you make of it? Any suggestions to improve it? Have you given some input when it was implemented?	Gather information on the issues with the new EPM system. To determine if the POs where included in the rollout of the system.
18	In what stage of the implementation do you think the EPM is?	To determine the level of involvement of the PO in the implementation of the EPM system.
19	Of the tasks you perform regularly, do you think they could be done more efficiently using the EPM?	To determine if the EPM system is being used to help the PO, if it fits the PO's needs.
20	If so, is there any extra tool that you require to achieve a better efficiency?	Determine the needs of the PO in his / her daily tasks.

Table 1 – Questions for the PO

[1] Recently implemented at the time the questions were done.

Although the present list is long, it is believed that it was necessary to ascertain the information needed to tie up with the information gathered in the review of relevant literature. Also, the interviews on which these questions were used were estimated to take about thirty minutes, a period that is considered adequate.

With the information gathered from the interviews it was possible to assess the problems in project management, from the POs point of view. These results were juxtaposed and crossed examined with material gathered from another method: related literature analysis from the relevant works regarding best practices in project management, namely critical success factors to effectively implement agile software development projects like Scrum. By tying together this information, it was possible to determine which of the identified problems on the model company should be focused for this work.

Because the main problem identified in the company was not having a structured set of processes, the next logical step was the research of correct modeling of business processes through the use of a BPM framework guided by the critical success factors best suited for the company. Another point of focus would be to acquire the necessary information to measure (through the use of metrics) the effectiveness of the work being done on each project.

Finally, the best practices to implement and manage these changes on the Information System of the company were reviewed, making the information available to everyone who needs it, when it is needed and following a common structure that can be understandable for everyone in every department.

As such it is possible to divide this work in six separate phases:

Phase I – Analysis of relevant literature for the project management, business processes and management of information system topics;

Phase II – Formulation of questions, based on the observation on the daily activities of the company regarding project management;

Phase III – Interviews with the company's project managers (each at a time);

Phase IV – Analysis of the interviews and juxtaposition with relevant information already gathered;

Phase IV – Design the previously identified key business processes and determine the related metrics;

Phase V – Analysis on how to use the EPM to obtain the data of the designed metrics, how to structure the information outputted by the processes and if the EPM provides the necessary information out of the box or if some other implementations

are required. This will lead to a set of guidelines for the correct use of the IS to store and consult information related with a project, ranging from project status to project metrics.

2.2. PROJECT MANAGEMENT

One of the cornerstones of this work is the conjunction between an analysis of critical success factors of managing software development projects using an agile methodology and an analysis of the perceived problems occurring in the model company by the product owners. In the following chapter, this work will focus on understanding the role critical success factors can have when managing projects especially the ones relevant to the problems the said company presents.

2.2.1. Critical success factors

Chow and Cao (2008), citing Bullen and Rockart (1981) describe critical success factors as "the limited number of areas in which satisfactory results will ensure successful competitive performance for the individual, department, or organization". This work's goal is exactly such: to discover what the areas where more focus in resolving issues must be given, so the processes of project management can be more efficient.

Of the approaches that could have been done at this point, it was chosen to review a broad selection of relevant literature to identify several factors that are pivotal for a successful management of projects in order to support the issues reported in the interviews. It is worth noting that the identified factors are consistent with the principles behind the agile manifesto (Fowler & Highsmith, 2001).

2.2.1.1. Communication factors

Communication is one of the bases of the agile manifesto. This factor is present in many of the selected literature. In Sheffield and Lemetayer (2013), communication is presented as an important factor that leads to a successful implementation of agile projects. But in the study of Misra, Kumar, and Kumar (2009), when adopting an agile methodology on project management, communication was not determined as an important factor linked to success.

Nevertheless, for Chow and Cao (2008), communication plays an important role and was one of the attributes in the project management process factor that appears in their survey results. Communicating is a key for passing important information and

knowledge in a rich agile environment, granting a boost in the possibility of a successful implementation of the project.

As this factor is one of the problems identified in the interviews to the POs and has such high importance in the literature, a solution will be proposed to solve this issue.

2.2.1.2. Requirements' assessment

The assessment of the project's requirements is an important factor in every project management methodology. However, contrary to older project management methodologies, the involvement of everyone in the project is recommended in agile methodologies, as the shared experience can help the decision making on important steps of projects. For Moe, Aurum, and Dybå (2012), not involving the team prevents the usage of their knowledge in the early stages, leading the PO to make decisions unsupported, without comprehensive understanding, which may have severe repercussion in the development of the product.

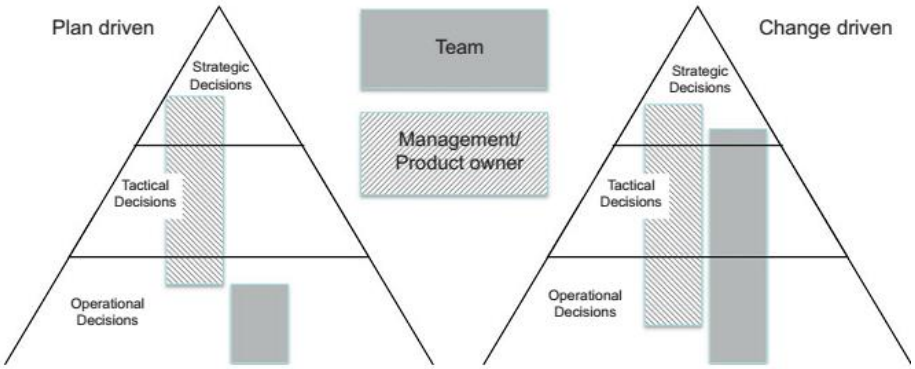

Image 1 – Requirements assessment from Moe et al. (2012)

As pointed out in the interviews, not assessing the requirements together as a whole (PO and the development team) is a risk that makes the project's success much more dependent on the experience and know-how of the PO, since he alone will be doing much of the strategic and tactical decisions.

2.2.1.3. Right amount of documentation

The agile manifesto (Fowler & Highsmith, 2001) postulates a "Working software over comprehensive documentation", relying on the tacit knowledge present on the team's members and their interactions to share the knowledge rather than on explicit

knowledge present on the massive documentation which is defended on the more traditional project management methodologies. Nevertheless, documentation is still very much necessary, as pointed out in the interviews, at the level of project management and also at the level of development. This last case will only be referred marginally, as the focus of this work is on a management level, but as an example, the use of documentation could be valuable to store information of development decisions or guiding material for the demonstration of the working prototype that is expected to be achieved at the end of each development cycle, as the project management framework chosen by the company (Scrum) advocates. Also, Chow and Cao (2008) found in their surveys done among Agile professionals that the right amount of documentation is an important factor and should be taken into account. The paper does not stipulate what that amount should be, which is correct, because the level of documentation will always be variable, depending on the type of client, type of project and also depending on the team and their experience. Documentation efforts should be focused on tasks like is postulated by Lindvall et al. (2004) and also Boehm (2002) which defends that "process procedures that involve tasks and milestone plans" should be documented.

2.3. BUSINESS PROCESS MANAGEMENT

The next focus of this work is an analysis of the processes in itself and the proposal of improvements. Additionally, the definition of metrics that will allow the analysis of the efficiency of said improvements will be sought. But to do that, theoretical information is needed to tie up with the aforementioned analysis, specifically information about BPM techniques. These techniques were chosen because, among all the approaches that could be implemented to improve the project management business processes, BPM is the most comprehensive, well-known, and widely used practice as Rohloff (2009) postulates and is cited by Škrinjar and Trkman (2013).

BPM can have a lot of definitions, as the one provided by Ravesteyn and Batenburg (2010) referencing Van Der Aalst, Ter Hofstede, and Weske (2003), in which BPM is "a field of knowledge at the intersection between Business and Information technology, encompassing methods, techniques and tools to analyze, improve, innovate, design, enact and control business processes involving customers, humans, organizations, applications, documents and other sources of information".

However, a definition more fitting to the scope of this work is that BPM represents the continuous efforts to analyze and improve the company's processes like the ones of project management, development, communication, among others (adapted from Trkman (2010)).

The analysis and redesign of the processes are done in conjuncture with BPM critical success factors and related best practices found in relevant literature. Afterwards, the processes can be modeled to improve them, respecting critical factors of success. Control metrics are also designed to control efficiency.

2.3.1. Implementation of the BPM framework

In this work the BPM framework referenced by Ariyachandra and Frolick (2008) is used to help develop the project management processes. This framework is composed by four core processes:

Strategize – In this step, it is determined what the strategy set out by the company is. With this information it can be possible (although it is said by Ariyachandra and Frolick (2008) citing Politano (2007) that this is a challenging task) to elaborate some metrics to measure the work being done on the processes.

Plan – It is on this step that the course of action is defined and the processes are designed as well as the supporting activities.

Monitor – It is at this stage that the importance of the BPM implementation is noticed, since results start to appear at this point. Based on the metrics, measurement results are available at the EPM system of the company. With this information, POs can plan additional actions, if necessary.

Take corrective action – At this final step (before the cycle starts over), the PO deploys corrective actions when a problem arises or when the data available indicate that problems could happen. An example related with this work could be the PO starts to notice that the team repeatedly takes more time than the estimated, while completing certain tasks. This could indicate that the estimations were under-budgeted or that there is a problem with the team and action should be taken.

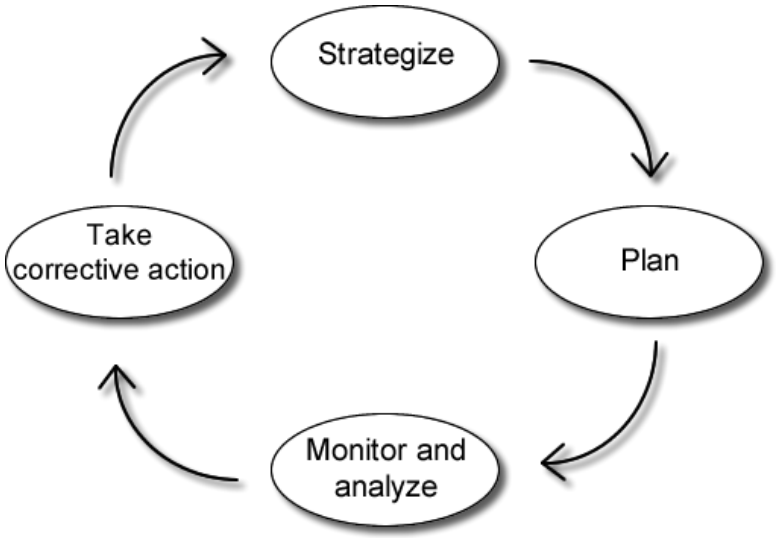

Image 2 – BPM framework

It is worth pointing out, that this looping iterative framework works in a similar fashion as in the Scrum framework, thus it makes sense putting these two working techniques together in use, to achieve the goals set out for this work.

2.3.2. Critical success factors and related best practice

While reviewing relevant literature, it became apparent that many of said literature presents a series of common CSFs to all of them, but still they are general purpose CSFs in their nature, like the ones purposed by Ariyachandra and Frolick (2008). Because of this, they could be used in other fields of research, like in the case of Information systems adoption research (in fact they are somewhat used in the chapter about Information Systems adoption). But while being general purposes in their nature, they should nevertheless be used, as they can form the building blocks for the BPM initiative. As such, the most relevant to this work are presented in the following table.

Critical Success Factor	Meaning
Champion	Central figure to support and promote the BPM initiative inside the company, giving information highlighting the gains of BPM.
Management of Resistance	Applying new work techniques always leads to resistance inside the company, out of fear or ignorance. By having the knowledge of who can have this behavior, additional support should be

Critical Success Factor	Meaning
	given to prevent eventual problems.
Management Support	The implementation of changes on how the company works must be supported by the management of the company.
Sufficient Resources	For changes to occur, it is necessary the initiative has enough monetary resources, people, and time.
Process Skills	For the success of the BPM initiative, it is necessary that people with extensive knowledge about the processes that will be worked on are involved in the process.
Technical Skills	Besides the importance of the knowledge about the process, it is also important to have people with good knowledge of technical skills to help translate the processes from a tactical level to a more operational level.
User Support	Since all or much of the company will be affected by the changes that an initiative of this nature brings, the support of the users is very important for its success.
Effective Communication	Only with an effective communication between all the participating agents of the initiative is it possible to transmit the correct information to achieve a successful implementation of the initiative.

Table 2 – Critical success factors for business process management.

Nevertheless, for this work, what was desired was to have a more concrete and in-depth CSFs that could be used to help implement the new design of the project management processes.

One identified CSF is the necessity to have the performance of the work being done constantly measured (Škrinjar & Trkman, 2013). The purpose of this CSF is to help determine the alignment level of the process with its goal, set by the management.

Likewise, Ravesteyn and Batenburg (2010) determined in their survey that taking measurements and controlling the work being done was also an important factor, and thus its monitoring should be a continuous effort. One example of such control happens at the end of each development cycle in a project (a sprint). This event is called a Sprint Review, where the data from the work being done is reviewed and conclusions can be drawn from it.

One of the objectives of BPM is to streamline the processes. Related to that, one of the CSF identified by Trkman (2010) is the need for organizational and structural changes, but also the company's capability to respond to that change. This critical factor helps "clean up" redundancies that exist in the company's structure by giving a proper definition of tasks, the competences expected from the workers and where responsibility lies in the processes. This is true when implementing changes where BPM is involved, like in the example of Jarrar, Al-Mudimigh, and Zairi (2000) where the change of management is also identified as a success vector, as well as the re-engineering of the business processes. Even further, one of the references of Rosemann and de Bruin (2005) also states that this process of change must rely not only on systems and structural change, but also on cultural change as well (Spanyi, 2003).

Referenced in many sources is the CSF of the need of IT/IS support of the BPM initiative. The usage of IT/IS should be an enabler and should be used to support the initiative, and not as the main driver for change (Terziovski, Fitzpatrick, & O'Neill, 2003). IT/IS can add its support to reduce the problems identified in the interviews that were related with the lack of documentation and communication.

But IT/IS support must go beyond the support for documentation, or else the system may become an information silo. Even more, Škrinjar and Trkman (2013) found that the use of a documentation system has little impact on the processes involved. An analysis of the existing system (EPM) will be done in a later chapter of this work to indicate tasks that could be done to support the BPM initiative. As reference to these tasks, some critical practices that were identified by Škrinjar and Trkman (2013), like the automation of several activities within the processes, to allow the workers to have more time to work on what really matters and not in tasks that while necessary (their necessity must also be object of attention in the re-design), are of a repetitive nature and can be achieved by automation.

Finally, a last CSF can be referenced from the work of Ravesteyn and Batenburg (2010). The involvement of the employees in the re-design of the process results in

their empowerment within the process and the company, which may contribute to the success of said process. This is because by participating in these actions, workers can see the changes and their benefits and act as champions, one of the most common CSF in relevant literature. This also goes along with what Scrum and the Agile Manifesto (Fowler & Highsmith, 2001) uphold, by having self-organizing teams and allowing business people and developers to work together daily throughout the project.

2.3.3. Metrics for measuring processes

The implementation of a BPM project is not a guarantee that the company will achieve better results. In fact, several papers show that BPM initiatives usually fail, like in Ariyachandra and Frolick (2008). The work of Al-Mashari and Zairi (1999) even states that "as many as 70 percent do not achieve the dramatic results they seek". One reason given for this occurrence was the lack of measurement to determine if the processes are aligned with the company's strategy. In response to this, measurements of the work being done can be achieved through the use of metrics.

The method of metrics for business processes is similar to the one depicted in the Image 2 – The BPM framework. In this case, as suggested by Kennerley and Neely (2003), there are three phases to the effective implementation of metrics that are also presented in a cycle. These are:

- **Reflection** – in this phase, an analysis is done on the existing metrics (if any) to determine which are effective or not (and thus should be pursued or abandoned) and what changes should be implemented in the metrics identified with such necessity;
- **Modification** – At this point, the changes that are required to be done in the metrics should be implemented on the IS;
- **Deployment** – Upon implementation, the IS should start collecting data to feed the metrics required to monitor the processes.

The work of Kennerley and Neely (2003) also presents a series of factors that can help with the constant evolution of metrics. These are the ones, based on observation, which would bring additional value to the metrics depicted on this work:

Factor	Category
Regular process to review measures with predetermined review dates and allocated resources.	Process
Availability of mechanisms to transfer best practice.	Process

Factor	Category
Integration of IT and operational objectives and resources.	Systems
Flexible IT systems. Enabling modification of data collection, analysis and reporting tools.	Systems
Availability of dedicated resources to facilitate review and modification of measures.	People
Development of a community of users of measures to transfer best practice (e-mail, user groups, benchmarking).	People
Culture conducive to measurement. Senior management driving measurement. Understanding of the benefit of measurement.	Culture
Acceptance of need for evolution.	Culture
Effective communication of measures and measurement issues using accepted media.	Culture
Use of measures to prompt actions, reflect on strategy and processes, etc.	Culture
Open and honest use of measures	Culture

Table 3 – Factors that help measuring processes

The categories of these factors represent the following:

- **People** – These factors provide the necessary human resources to implement the metrics;
- **Process** – Factors that guarantee that a process exists to manage the metrics;
- **Systems** – They are the factors that provide a structure that enables the collection of the measurements and outputs its results;
- **Culture** – The existence of a company culture that is focused on the gathering of information to feed the metrics.

Of all the factors presented by Kennerley and Neely (2003), only the cultural factors were all adopted for this work. While determining the necessary metrics for the processes can be a difficult task, dealing with cultural issues is also very challenging. Nevertheless, in the case of DRI, its culture is also what led not to include some factors presented on the referenced work, because DRI (through its workers) presents a strong and collaborative effort to constantly improve its methods of work.

When implementing this framework, a correct profiling of the target company should be obtained in order to consider the appropriate factors from the ones presented above: people, process, system and culture.

Another valuable point that should be addressed when creating the metrics is determining which purpose is behind each type of metrics. Behn (2003) presents a series of possible purposes in his paper, of those, some can be used on the processes that will be outlined for this framework.

Evaluate and control – By evaluating a process, a manager can have a better grasp on how well (or not) the process is being used and if it meets the goals that should be set when the process is implemented. For this purpose, Behn (2003) cites a report done in 1994, by the National Academy of Public Administration (NAPA, 1994) that states "Performance measurement of program outputs and outcomes provides important, if not vital, information on current program status and how much progress is being made toward important program goals". While the previous reason to measure had its focus on the process itself, control is aimed at the people that are involved in the process. Although DRI, by adhering to an agile oriented framework (Scrum), is looking to empower the team (Fowler & Highsmith, 2001), there is still the necessity to exert control to determine if the results are in line with the goals and strategies set by the company;

Motivate and celebrate – Measuring a process can also be used to motivate a team. Long (but not that long) goes the day managers followed the Frederick Taylor's process control (Behn, 2003), now shifting in favor of the agile empowerment. By measuring the positive results of a process, they can be used to properly motivate a team. If the results are not positive, they can be used to steer the team towards the goals defined in the sprints. By giving this opportunity to the team, it may induce the joy of reaching the goals on their own terms, increasing the motivation to achieve more, which is what Rising (2013) defended on her presentation. Celebration of the results of measuring a process is, contrary to the previous point, target to the whole company. By celebrating the reaching of goals of a process that were set in accordance with the strategic steer of a company is something that will induce a greater sense of team / company cohesion and may lead the individual to improve their self-worth by participation in such successful endeavor, which in turn may improve their performance output;

Learn and improve – Alongside with the evaluate purpose, the need to learn while measuring processes should be an imperative, since by doing so, the manager

can grasp the causes to those results (Behn, 2003). Specific metrics should be implemented to obtain such measures to persist with the continuous improvement. This leads to the final purpose: improvement. This purpose follows, to some extent, the point presented on the Agile Manifest (Fowler & Highsmith, 2001) which states "At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly". By knowing what happens on the process, actions can be taken to improve the process if necessary or to continue in the same fashion, if the indicators point to a satisfactory measurement.

2.4. INFORMATION SYSTEMS ADOPTION

The final theoretical focus of this work is on information system (IS) adoption and implementation practices. This information is gathered from the field of Management of Information Systems and its purpose is to act as the basis of a series of guidelines to allow a better adoption on the implementation of the redesign processes by the users of the EPM system.

The work of Silver, Markus, and Beath (1995) provides a common academic definition of Information Systems (IS), in which IS encompasses hardware, software, data, people, and procedures. With the exception of hardware, which does not have much relevance in this work, these components of IS are of high importance for a company and for the success of the implementation of the processes discussed in this work. These components can be described in the following:

- **Procedures** – The main component, which is the basis for the new implementation of this work – the processes. These must be coded in the form of procedures in the EPM system;
- **Data** – Data is what results from the activities of the procedures, produced by the people and will be stored in the EPM system;
- **People** – The workers of the company (and more if necessary) that will interact with the processes and will produce the data. While the procedures are of project management nature, not only the PO will interact with the software;
- **Software** – The EPM system in which the procedures will be coded, the data will be stored and accessible to the people that will interact in the future processes.

While it is not an extensive analysis for critical factors for the implementation of an IS, the work of Subramanian, Jiang, and Klein (2007), citing Alter (1979) purposes a

series of guidelines for implementation of an IS with a nature similar to what is in study in this work, and as such, it is important to take them into consideration. These guidelines are:

- **Keep it simple**. Implement / Deploy only which is really necessary and needed by the users of the IS;
- **Executive or top management participation commitment**. It is necessary for the top management to be involved with the development and support its implementation and its use throughout the company. As can be seen before, management commitment is a constant in all of the areas of study in this work;
- **Training**. Providing training to the users can even permit a further refinement of the IS and at the same time involve the users in its deployment to achieve a better acceptance within the company;
- **Prototyping / evolutionary deployment**. The implementation / deployment of the IS can and should start simple. But, with each iteration, new features can be added or more users should be brought in.

2.4.1. Adoption of changes

Changes are many times difficult to implement. Some even say that changes hurt on a psychological level like Koch (2006), citing medical sources. Resisting change is part of who we are, moving away from our comfort zone is something that we try to avoid on a conscious and sub-conscious level.

So there is a great necessity to pay attention on how change is managed within the company to ensure the success of the implementation of the changes suggested on this work.

First of all, and from an IS perceptive, the necessity of these changes is a by-product of the growth that the subject company has been sustaining for some time and also out of the necessity of innovation (Swanson & Wang, 2005) on their processes, added by the pressure of the markets in which the company is present to have services that deliver high value to their clients. Thus, there was the necessity to implement an Enterprise Project Management system.

Many of the relevant literature states that in the process of the adoption of such changes on the IS level of the company, the changes should be supported by one or more champions, who are ‘‘technological leaders’’ responsible for convincing the rest of the company of the need for change. The champions’ role is important, when they

support the introduction of new information technology unequivocally, firmly and decisively, so that the rest of the company is convinced of the need to adopt it. As such, it is necessary to identify them within the structure of the company. This is stated in the work of Bruque and Moyano (2007), which studies the determinants for adoption of information technologies by SMEs. While this work is focused on family and cooperative companies, a correspondence can be made to DRI, the subject company of this work, since as in a family company, everyone knows one another and many even have dealings outside the company. Nevertheless, like in the work of Bruque and Moyano (2007), it can be observed on DRI that this environment is starting to change and the company is becoming more structured and (even) more professionalized.

Like in the example above, these changes lead to a transformation on the culture that should be taken into account when managing change at a company. Corporate culture is an ever changing (and some aspects cannot even be controlled) property of a company, in which knowledge of more senior workers is passed to more junior and recent workers in the company (Jackson, 2011). Hynes (2009) citing Deshpandé and Webster Jr (1989) describes corporate culture as "the pattern of shared values and beliefs that help individuals understand organizational functioning and thus provides those norms for behavior in the organization". While implementing changes, the aspect of corporate culture is many times overlooked and is often cited as the primary reason for the failure to implement corporate changes (Linnenluecke & Griffiths, 2010).

It has been observed within the model company that its culture can be identified following an integration perspective, as a whole, in which it displays an informational culture, meaning that the IT and business goals are aligned (Jackson, 2011). It is also possible, however, to identify from the perspective of differentiation, a culture that each of the departments (systems, web development, applications development, etc.) has, comprising different perspectives on information systems and how to adopt them. With an unattended transformation caused by the implementation of changes, this behavior could even deepen, leading to a non-adoption of said changes.

This concludes the round-up of the research done for this work, which started with the analysis of the project management processes of the company, specifically the problems that plague them and a comparison with proven critical success factors in relevant literature. Since the analysis is on processes, it makes sense that as the necessity of changes was identified, a redesign should be made on said processes,

using BPM for that goal. To that end, an analysis was also done for practices on how to implement changes using BPM and employing identified CSFs. Because these changes need to be supported by an Information System (IS), the best practices on adoption changes on IS were also object of study for this work.

3. FINDINGS

With the theoretical background completed, the findings are now presented. Firstly, the subject company is contextualized in relevance with this work, regarding their strategic goals, but also regarding the chosen project management methodology.

3.1. ABOUT THE COMPANY

The company focused on this work is a global consultancy SME named DRI (http://www.dri-global.com), headquartered in Lisbon, Portugal.

DRI was established in 1999 as a result of an academic spin off. As of January 2013, it counts with 40+ employees, placed in 7 offices in Europe and North America. Its main service vectors are:

- Business & Social Intelligence;
- CRM & Social CRM;
- Mobile & Emergent Media;
- Web & Platforms.

These services are mainly related with web development, done primarily with open source technology or services related with hosting and helpdesk support. The development is mainly done in the Lisbon offices, although there are projects developed by multi-country teams, where communication is an important topic.

Its current strategic plan aims (among others that fall outside of the scope of this work), on a customer level, to reach bigger clients that lead to more complex (and thus lengthier) projects. To support that, on an internal processes perspective, the company aims to improve the operating processes and also the product quality.

On an organization level, the majority of teams is focused in web-related development. But there is also a team of system administration, a design team, a financial team and also a commercial team. In addition to the development teams and besides the project managers that act as a PO for one or more projects, there are a number of team members that support the development of a project, as specialists. Team members like the CTO, the UX Lead or a Software Architect are commonly involved with the teams to deliver a product specially crafted for the client. There is a team member with the position of Controller that acts as a hub of information between the teams of different areas.

3.1.1. Project management methodology at DRI

For much of its still short life, DRI did not have an official project management methodology. Nevertheless, such importance was pointed out daily, as the company grew and matured. To achieve that goal, for some time now, several top management elements at DRI have been pushing for a complete adoption of Scrum, an agile software development framework.

Scrum is an "enhancement of the iterative and incremental approach to delivering object-oriented software" (Schwaber, 1997). Some of its ideals and methods were used in a meeting of representatives of several methodologies, like Extreme Programming, Feature-Driven Development, Scrum, etc., producing the Agile Manifesto, signed by all participants in said meeting (including Schwaber), in 2001. The manifesto defends the following:

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

Scrum (which takes its name from a formation in rugby) postulates that the software development should be done in small timed iterative steps, with the involvement of the stakeholder to improve the quality of the product. Stakeholders can be the customer, end-users, etc. They are the ones that receive added value with the development of the project.

Each of the development iterations is called a Sprint and can take from one to four weeks, depending on the stipulation given by the Project Owner (PO). In Scrum, the Project Manager figure does not exist, and it can be somewhat replaced by the PO figure. Since the sprints are fairly short, after evaluating what went right and wrong, changes can be implemented rapidly.

While analyzing a project, its requirements are broken down into stories by the PO. Each story represents a functionality of the project and they are further decomposed into issues by the development team. A component of the product can be composed by several stories. With a sorting order of the importance of said issues, these set of issues is what composes the product backlog. This backlog is where the development team, the Scrum Master and the PO of the project come before the start of every Sprint to build the Sprint backlog by choosing the first stories from the product backlog that fit the length of the Sprint chosen at the start of the project by the PO, to answer the goals for the Sprint, also set by the PO.

The implementation of this methodology is taken into account and is incorporated in the modeling of the processes described in the chapter 3.3. These processes (**Procedures**) are an important component of the Information System of the company, referred in the theoretical background and their modeling will address the problems found at the company. From these processes, **Data** will result and is to be stored in the **Software** made especially to manage projects, the Enterprise Project Management (EPM) system (both of these components are, as pointed out in the theoretical background, part of the IS of the company).

The final component of an Information System is the **People**. At this company, and for the case presented at this work, the bulk is the development teams. Each team can be comprised by two or more people, depending on the project at hand. Besides the already discussed PO figure, there is another element present on Scrum teams: the Scrum Master. This element is a person whose main focus is to overcome problematic situations, like missing information, missing resources, etc. The Scrum Master is present at all the Daily Meetings, a meeting that, like the name implies, is done on a daily basis and where one or more teams of a specific department gather to discuss the work done in the previous day, which problems occurred and what will be done in the present day. At these meetings everyone is encouraged to step forward with solutions to problems that their colleagues present. If no solution to the problem is found, it is up to the Scrum Master to find resources to help solve the problem.

3.1.2. Enterprise Project Management

The EPM system should be a fundamental part of the Information System of any company and as such, is one of the focus points of this work. In this system, all information of a project can be stored and consulted, by using the different components. The suite used at DRI is from Atlassian and the main components this work will concentrate its findings and discussions are:

- **JIRA** – A project management application, that helps to track, plan and analyze the data from a project. This application has several plug-ins installed, one of which is the **Tempo** plug-in that is used by the development team to log the time spent on each task of a project. Another plug-in is the **GreenHopper**, that is a visualization plug-in to display that tasks of the project in a Scrum board, with the several stages of the development, which typically is To-Do, In Progress and Done.

Another plug-in used is the **Zephyr**, used for storing information about the tests that should be done for the development being made;

- **Confluence** – A wiki like application that is focused on team collaboration. In this application, POs and the development team alike can store information (that should be in a common and structured way) about the project. This application also allows controlling / viewing several calendars, where much information about the teams (among other information) can be displayed to help every user plan its work.

Some EPM suites were analyzed and this was the one chosen to support the effort being made at DRI to improve the quality of its work. It is still a work in progress, with good acceptance from its users, but it requires some guidelines about its use, which are presented in this work.

When working with the EPM system, users should log not only the time spent on their work on issues of the project, but the time used on common tasks that exist in every project.

Name	Description	Audience
Daily Meeting	Each time a worker goes to a daily meeting, he / she should input the time spent in that meeting. If a worker is on two projects at the same time, the time should be logged in the project where the most time will be spent in that day.	Development Team and Scrum Master. Occasionally for PO.
Sprint Review	At the end of a Sprint there are several meetings that take place. The time spent on the sprint review, where the developed work is demoed to the PO, should be stored here.	Development Team, Scrum Master and PO.
Sprint Retrospective	At the end of a Sprint there are several meetings that take place. The time spent on the sprint retrospective, where everyone involved in the project gives an input on what went right or wrong with the sprint, should be stored here.	Development Team, Scrum Master and PO.
Sprint Planning	At the end of a Sprint there are several meetings that take place. The time spent on the planning for the next sprint should be stored here.	Development Team, Scrum Master and PO.
Backlog Grooming	The development team should know, at least, what they will do for the next month or so. Knowing this allows looking at the product backlog and doing some grooming, to see if any information is missing. The time spent doing this task should be inputted here.	Development Team.
Project Planning	When someone is involved in the initial phase of defining aspects of the project they should insert the time spent in this issue.	Development Team, Scrum Master and PO.

Name	Description	Audience
Analysis	The time spent on analyzing some new technology or development process should be logged here.	Development Team

Table 4 – Common project tasks

3.2. INTERVIEWS ANALYSIS

As stated above, one of the methods to gather information for this work consisted on a set of interviews. These were made at the starting point of this work (Phase III) to each of the three available POs at the time (the fourth was unavailable).

Upon conducting an analysis on the interviews, it became clear that the main problems in managing projects at DRI are: communication, documentation, requirements' assessment and the general lack of standardization of metrics and other status and control structure to evaluate projects and their success. As pointed out in the theoretical background, problems of communication, documentation (and at some extent requirements' assessment) are related most of the time, since much of the communication being done is not stored in a central repository and documentation is not structured in order to support communication. For complete transcriptions of the interviews, please consult the chapter 7.1 of the Appendix.

3.2.1. Communication, documentation and requirements' assessment problems

Project management communication is a problem that has multiple vectors: with the financial department, between the product owners and with the client. Naturally the POs also communicate with the development team, but in the interviews this vector was not referenced as problematic.

The communication between the PO and the financial department is mainly done in person and / or through email. The financial department mostly starts this communication by inquiring about the state of a project, because they have the information that in that period they were supposed to bill the client for the project. While this form of communication usually works most of the time and was only marginally referenced in the interviews, it is not as effective as it can be, since this situation leaves the financial department dependent on the initial information referring to when a client should be billed. If for some reason that information gets outdated, there is not a process to distribute that information.

The case of communication problems among product owners is due to the lack of a structured procedure of communication between them. One issue that was not pointed in the interviews but was commented afterwards was that the information of

the allocation of the teams is not easily accessible to every PO, although there is some progress in that field, with regular POs meetings, where the information about the status of projects is shared. The information is not available on the EPM system of the company, as it resides today on a shared online document.

These problems can lead, for example, to the loss of some of the members of the development team (or the whole team) by the Product Owner to another project that has a greater perceived importance, although there is not a distinct definition of what the importance of a project is.

Finally, the problems of communication with the client are a broader issue as they also involve documentation and requirements' assessment issues. While the main problem is when information is needed from the client during the development of the project and said information longs to arrive (something that eludes the purposes of this work), there are also problems with the initial communication with the client, when the project is in the requirements' assessment stage. Since there is not a structured path on how to proceed at this stage, the information is gathered and stored in different manner by each PO, depending on their experience. This also leads to poor estimations, as the project objectives are not clearly known. Also, in the interviews, it was possible to determine that there is not a method to classify a project regarding its importance for the company and also to categorize each of the requirements in term of their importance, to determine the critical success factors for the project.

Resolving these issues can also help reduce the necessity of information from the client in the later stages of the project, which causes waiting times and hinders the normal flow of the project.

3.2.2. Lack of control on project status and further analysis problems

Another major issue raised on the interviews was that there are not established structures to control the status of the project yet. This means that from the start, there is information that is missing, namely which are the critical functionalities that should be implemented for the project to be considered a success.

Likewise, there is not a standard way for the PO to determine if the project is on track and on budget. A situation that is also true regarding the development team, which has a large importance in the organization of work in the agile methodologies. It is desirable that they have this information to give their input relating the development that should be done. As for when the project is concluded, there is not a

predetermined set of rules to declare if the project was a success or not, since there are no metrics to help determine it, alike there is no metric to determine if the estimations made for the tasks were in fact the time spent on them.

3.3. MODELING THE PROCESSES

With the information obtained from the interviews, unscripted talks with peers and through work experience, it is possible to identify the five business processes related to project management that represent the workflow since the beginning of the project to its end. The end may represent the actual end of the implementation of the product or the end of a phase of the project if the project has the necessity to be divided into phases. Each of these processes will include, whenever possible, metrics to measure the quality of the work being done. With this data it will be possible to determine if the methodology employed is working as expected or if it is necessary to introduce changes onto it. This analysis should be done yearly. This time period is best suited since a shorter time frame would not generate enough data to allow an accurate perception of the effectiveness of the process, and a longer time frame, while valid, could present a risk, by perpetuating inefficiency.

To increase the changes of a successful implementation of the BPM framework on the processes of project management, the usage of the critical factors discovered on the theoretical background is pivotal. As such, the champion of this project is long identified (it was the first subject being interviewed), but from day one, all the management supports the initiative. Moreover, with a correct usage of the IS supporting those changes will come a good distribution of the global knowledge about the processes as well as the technical skills required for these tasks.

Regarding the steps pointed out in the theoretical background for modeling a process using a BPM framework, the stages of Strategy and Plan are conducted in the scope of this work. First by aligning the modeling with the strategic goals of the company, then by refining the processes and employing a structured documentation to support the improvement of quality of the work, by providing the information when needed with a common structure across projects. Finally, the Plan phase will focus on the actual modeling of the processes.

However, firstly it is necessary to model the processes, as the processes discussed on this chapter had not been previously modeled, since there was not a formal implementation of them. Because of this, each PO would implement the project

in their own way, making the modeling of the processes AS IS impracticable for this work. Because of this, only the TO BE modeling will be presented.

All the processes will be presented as a graphical representation using the Business Process Model and Notation (BPMN). The usage of the BPMN is based on the easiness of implementation and the ample set of artifacts that can be used for modeling, demonstrated on related subjects on the masters' course. Also, the analysis of Wohed, Aalst, Dumas, Hofstede, and Russell (2006) which evaluates "On the Suitability of BPMN for Business Process Modeling", sustained the decision to use this notation.

These next chapters describe each of the processes, by displaying the TO BE model, a complete description of it, the usage of the EPM system on how to obtain the information gathered in the processes and if there are any, a complete description of the metrics used on the processes.

In the scope of this work, each project should have the following processes' workflow:

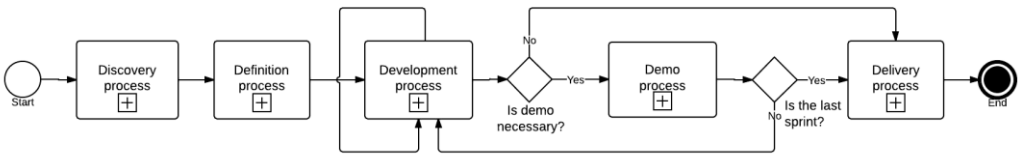

Image 3 – Complete process workflow for a project

3.3.1. Discovery process

The first process is called **Discovery**, and it is started when the project is awarded to the company. Since the initial contact is typically made with the commercial department (although, depending on the type of client and project, some management staff can also be involved early on), many aspects of the project have to be discovered. A representation in BPMN notation is presented below:

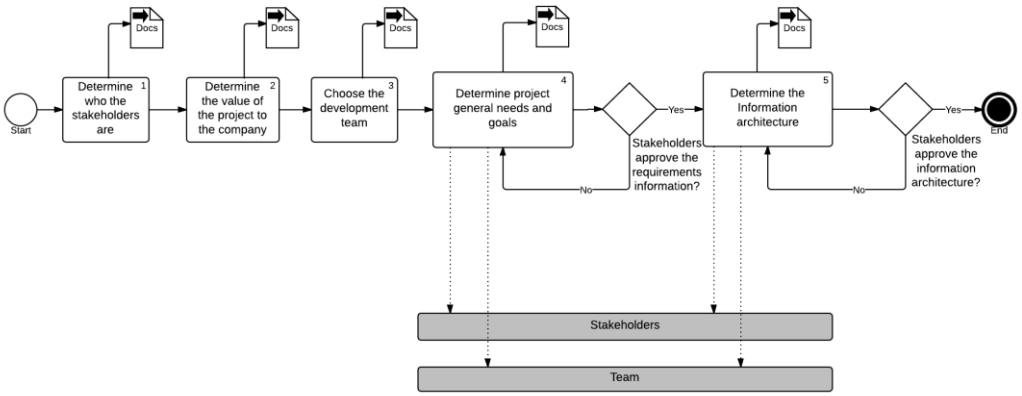

Image 4 – Discovery process

3.3.1.1. Process details

For this process, the focus is to determine the project needs, which range from the business' needs, the client's needs and the final user's needs. The project's goals and related constraints that may exist are also gathered at this stage. All of this information is discovered in collaboration with the project's stakeholders, but also with the development team, as the team can add a valuable insight based on their experience gained in previous projects.

So the first task (#1) will be the identification of said stakeholders, since they are the ones that should have all the required information indicated above, thus this task should be done as early as possible.

In the next task (#2) the project rating will be determined. This rating is an average of several indicators to define the level of importance of the project for the company. This can lead to decide which team is assigned to the project, depending of its complexity, if the project is of high priority for the company's interests or the order on which the projects will be developed. Below are the indicators in a table format:

Indicator	**Description**
Value of the project	Information related with the overall price or time equivalent agreed with the client.
Importance for the company	Determine if the completion of this project will bring future projects with the client or if the project is of high visibility.
Complexity of the project	High-level evaluation of the complexity in order to

Indicator	Description
	complete the project.

Table 5 – Project rating indicators

Each of the indicators should take a numeric value from 1 to 10 and their average will indicate the project's importance for the company. These indicators should be revised in the yearly evaluation of the processes and more can be added or removed if their practicability is not proven.

With the rating of the project it will be easier to determine (on task #3) which team should take on the project or even decide to use several teams, based on the knowledge of their development capability to tackle the complexity of the project, but also if that team if available or not in the expected period of development.

To determine when a team is available, the PO will have a tool on the EPM system. The Confluence application in the EPM has a series of calendars where it is possible to obtain information of various types. For this case, there is the Team Assignment Calendar depicting that information. This information is typically collected at a meeting that occurs weekly, where all the POs convene to report on their projects. The company's Controller gathers that information and updates it on Confluence. It is important to determine which team will be assigned to the project as early as possible to introduce them to the project, but also to have their input based on their tacit knowledge (gained in previous projects) at this critical phase of the project, where the requirements are being identified.

The next task (#4) is the pivotal one and is where all of the initial information will be gathered. During this collection, the information can be structurally stored in Confluence for reference and communication with the rest of the team (the structure is presented in the next chapter). This task is important to determine the project needs, goals (such as the implementation of the most important components for the stakeholders - CSFs), and functionalities (also known as functional analysis). These topics are discussed in close collaboration with the team, as pointed out as a critical success factor in the theoretical background. The information gathered is also a safeguard, because the stakeholder must acknowledge and accept it before moving to the next task of the process.

Afterwards, on the last task of this process (#5), the information architecture will be determined. The information architecture is where it is determined how the data is

organized and how it is processed. Also, the stakeholders must formally accept this information to close this process.

At each of the process steps, a series of information is gathered and should be stored in the company's EPM. With this documentation, the problems identified in the interviews should be quenched, such as the problems in getting the correct requirements from the client and project's documentation. The development team will have access to this documentation, and a validation of information must be obtained from the stakeholders by means of electronic communication.

3.3.1.2. Usage of the EPM

All of the information gathered in this process should be inserted onto Confluence for future reference. Regarding this process, the following diagram depicts how the information should be inserted. Each element in the diagram represents a page in Confluence:

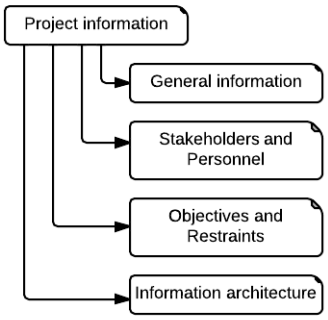

Image 5 – Information structure of the Discovery process

The first page, Project Information, works just as a placeholder for all other pages and will be used in other processes.

- **General Information** – A general description of the project. Also present in this page will be the information of the project rating (from task #2);
- **Stakeholders and Personnel** – This page stores the indication of the stakeholders and their contacts for reference, as well as the personnel of DRI that will be involved in the project, saving the information gathered in tasks #1 and #3;

- **Objectives and Restraints** – All of the information gathered on task #4 should be stored at this page;
- **Information architecture** – The information gathered from this topic on task #5 is stored in this page.

After this information is "discovered", the project can move on to the next process.

3.3.2. Definition process

The next (second) process is **Definition**. At this stage, what is important is to analyze the information gathered from the previous process, break the information down into manageable bits and categorize it, making it possible to obtain more accurate estimations of the work to be done. The main agents involved in this process, besides the PO, are the stakeholders and the development team.

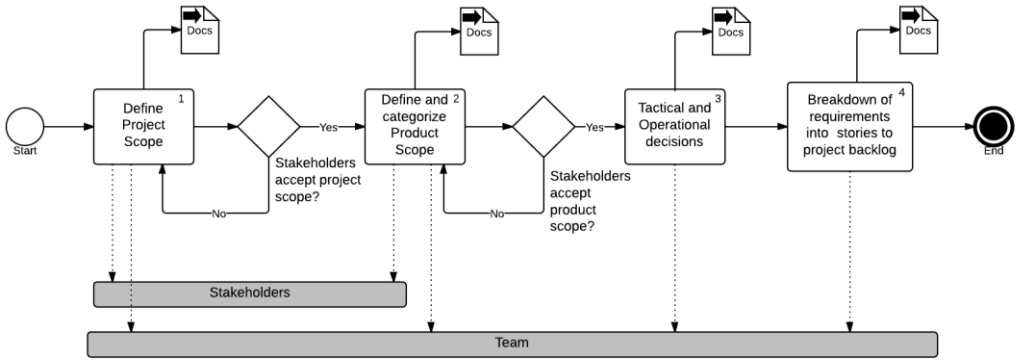

Image 6 – Definition process

3.3.2.1. Process details

In this process, the main task is the definition of the scope of the project, but also the scope of the product. The difference of these two scopes is described in the PMBOK Guide (Project Management Institute Inc, 2008):

- The **project scope** relates with the goals of the project, meaning that the project should result in an answer (typically a product) to the needs of the client;
- The **product scope** is related with the product that is produced in the project and represents the features that it must deliver.

While developing the project, the lack of a clear definition of scopes has been identified as the cause of delay in some projects. As such, the first task of this process (#1) is where the definition of the first scope will be addressed. The PO can consult with various elements of the development team to assist in this task, if necessary.

Afterwards, the stakeholders should approve this scope. If the stakeholders later require changes of scope in the project, the information gathered here will help the PO to mediate any conflict that may arise from this request. It is not that change cannot be done. In fact, by using an agile methodology, the company embraces change. The point here is that by defining a scope, a commitment is done by the PO in which that project with said scope will be done in X time. By adding changes, the stakeholders must drop some other functionalities that are not as necessary as the changes proposed by the stakeholders. Another option is that the changes are incorporated in the scope by renegotiating further payment and time.

Next in line (#2) will be the definition of the product scope and the categorization of its components in accordance with their importance, in order to obtain the critical success functionalities for the project. This may be seen as a breaking-down of the goals of the project that were previously defined in the **Discovery** process, with further refinement. Again a close collaboration with elements of the development team and the stakeholders is important to obtain the maximum amount of information. After all of the information is collected and stored in the EPM system, the stakeholders must formally accept these requirements to advance the process.

Activity #3 is where tactical and operational options are discussed and decided. Together with the development team and other in-house experts like the CTO or the Senior Software Architect, it is discussed in detail what is the best way to implement the project, which technologies should be used, among other decisions relevant to the project, for instance if the project should be done in whole, or in phases to deliver additional value faster to the final client. If this happens, the next processes repeat as many times as there are phases in the project. Based on the information gathered of the scope of the project / product and also with the project's goals gathered on the **Discovery** process, it is possible to obtain a list of tests that should be done to determine if the product is ready for delivery (acceptance tests).

The final activity (#4) is the further breakdown of the requirements into stories that form the product backlog, which is an artifact of Scrum. For each of the stories created, the PO must indicate an estimate for its cost of development, the Original

Time Estimate (OTE). To reach this estimate the PO can consult with the development team. Additionally the PO must define what the acceptance tests are for each of the stories. At this point, the PO should have enough information to decide the length of the Sprints, and although Scrum does not postulate the change of said length, as it may cause a misrepresentation of the metrics, the PO must have room to change it when the project status requires it, since Scrum also upholds the embracing of change.

3.3.2.2. Usage of the EPM

Unlike the previous process, the information gathered in each of the tasks of this process will not only be stored in Confluence, but also in JIRA because the final task (#4) of the process outputs stories. To create stories on the EPM, the PO should create issues of the type "Story" with the following information:

Name	Description
Description	The general description of the story. What is expected from the story and other relevant information.
Where to Demo	The location on where to demonstrate the functionality of the story.
How to Demo	How to demonstrate the functionality of the story to the PO.
Acceptance Tests	The definition of the acceptance by the PO, which generally corresponds to the how to demo information. Additionally, indication of tests necessary to achieve the status of accepted story should be inserted through the use of the Zephyr plugin.
Definition of Done	The general definition to determine if the story is done.
Original Time Estimate	The estimate done by the PO. This is the time that, given all the inputs provided by the development team (if any), the PO expects to spend on this story.
Sub-tasks	This data is introduced by the development team, with all of the necessary tasks (actions) to be performed to complete the story. All existing sub-tasks must have an estimation done by the development team, and are entitled Remaining Time Estimate (RTE). Creating sub-tasks is not always necessary as the story may be atomic in nature, meaning that the story describes a unique action to be taken. In this case the development team inserts their estimation (RTE) in the story itself.

Table 6 – Information needed for a story

The stories must have all the information needed for the development team to analyze and to create sub-tasks to develop the story. It will be on these issues that the team will work on and log their work time (also in JIRA). But as this task is the responsibility of the development team, it is not depicted on this process' model. This activity is called backlog grooming.

The story can only be considered ready to be included in a sprint backlog after the team inserts the information described above and marks the story as "Ready".

When analyzing the project at the end of a sprint (done in the **Development** process) or even at the end of the project (done at the **Delivery** process), the PO can look at the discrepancies obtained (through the analysis of metrics) between the estimation given to the story and the sum of the issues the development team gave. That information can help the PO make more accurate estimations on other projects.

The following diagram displays how the information that this process outputs is stored on Confluence:

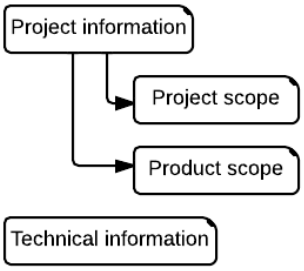

Image 7 – Information structure of the Definition process

The first page, Project Information, is the same depicted on the previous process that also holds the general information related to the project.

- **Project scope** – Like pointed out in the previous chapter, this information is gathered from the stakeholders in task #1 and should be formally approved by them. Any changes that are done afterwards, should be inserted here along with the rationale of that change, what has been decided and the formal approval of the stakeholders;
- **Product scope** – The information of the scope of the project will be lodged at this page. Like the previous scope, all changes in this scope

should also be stored in this page, with the justification of the change and the formal agreement of the stakeholders;

- **Technical Information** – This page is on the same level as the page Project Information and it gathers all the information that results from task #3. The information is, as the name implies, of a more technical nature contrary to the previous pages. The information ranges from access data to applications and services used on the project. Since much of this information varies from project to project, it is impossible to easily categorize the information to differentiate in diverse pages. Also, the organization of the subpages is done by the development team and not by the PO of the project, which is the main recipient of the findings of this work.

With all the information worked on and accessible by all members of the team, the project can evolve to the next process.

3.3.3. Development process

The **Development** process, the next process in line, is where most of the development of the product is effectively done.

This process is highly iterative, meaning that it will be processed as many times as there are sprints for the current phase of the project or for the entire development stage. The main development stage is where the product is being development based on the information gathered on the previous processes. Nevertheless, new development iterations could happen in the **Delivery** process, if small corrections are proven to be necessary and if they can be done. For these corrections (and for the delivery of the product itself), another set of sprints will take place.

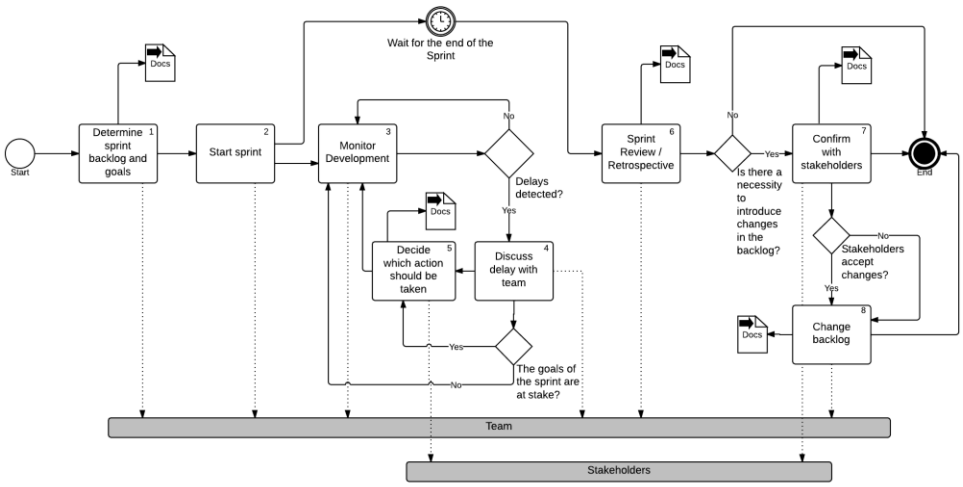

Image 8 – Development process

In terms of project management and following the actions proposed by the Scrum framework, the project owner has, with this process, the main mission of monitoring and controlling the development of the product.

3.3.3.1. Process details

The initial task (#1) in the process is the definition of the sprint by the PO in close collaboration with the team. Like previously discussed, if necessity arises, the duration of the Sprint can change. The reasons could range from the availability of the team, the complexity of the development of the upcoming sprints or the status of development requires more time to produce a presentable prototype.

Still in this task, and with the information of the duration of the sprint, the team proposes a sprint backlog, based on the sorting order the PO gave to the product backlog. As for the PO, he should set out the goals for the sprint, based on the project planning. So, to complete this task, an understanding between the PO and the team must be reached on what the PO accepts as the result of the sprint and how far the team commits to develop in said sprint.

With all this information defined, the sprint is started by the PO at the task #2, by indicating it on the JIRA application.

Afterwards, task #3 occurs during the length of the sprint, in which the PO will monitor the time spent on the development being made by the team (against what was previously estimated). This monitoring is achieved by using the EPM system. For

each project, JIRA provides a visual aid, the burn-down chart. This tool will be further detailed in chapter 3.3.3.3.

Another tool to control the state of the project can be the attendance of the daily meetings. Over these meetings, every member of a team or even of an area of the company (the web development department, for instance) participates in a daily basis, it has a duration of no more than fifteen minutes. In these meetings, in which the PO can be present but just in a watcher capacity, the team members discuss which tasks were done in the previous day, the difficulties that appeared and other issues that are thought relevant. Every other co-worker can give their input to solve a problem. This way, the PO can have a good grasp of the status and problems of the project and can discuss them with the team after the meeting, if necessary.

If a problem (delay) is detected, the PO must determine its cause with the team (#4). If the problems cannot be quenched within the sprint, some items of the sprint backlog may have to be postponed for a future sprint. If this happens, it may even lead to some functionalities with least importance to be put aside in order to conclude the most important functionalities of the project. That decision must be done in collaboration with the stakeholders (#5), and that information is stored in Confluence for future reference.

At the end of every Sprint, the PO gathers with the team and the Scrum Master (at task #6) to review the project's progress and the team demonstrates the new functionalities to the PO. By doing the meeting, the PO has another tool to control the progress of the project and make management decisions upon the outcome of the meeting. These results should be stored in Confluence.

It may be possible that not all of the stories of the sprint are dealt with according with the objectives of each story, for instance implementing something on an estimated time, or even with the goals of the sprint, resulting in a reimplementation of the story. This may lead to more time being consumed so other stories may have to be dropped. In this situation it is necessary to check the acceptance of the stakeholders (#7). In the positive case, changes are made in the product backlog (#8), sorted according to the importance the PO gives to the story. In the negative case, changes are done nevertheless, but no other stories are dropped, since it was the liability of the company to reach the set goal in the estimated time. It is worth pointing out that this situation should not occur with the implementation of these new processes, as the data that supports the estimates is more complete.

3.3.3.2. Metrics

The information gathered in the EPM can be showed in metrics that can be of two categories: **informative**, a simpler metric that indicates a value that can be compared with other metrics to reach a conclusion; and **comparative**, a more complete metric that already compares two or more values to help the PO reach some conclusion. For this process, the following metrics are available:

- **Selected Work** – This informative metric presents the sum of the estimations made by the PO (OTE) for the issues selected for a sprint;
- **Committed Work** – This informative metric is a sum of the estimations the team gave (RTE) to the sub-tasks of each of the stories selected for a sprint. This may differ from the estimation given by the PO to the story these sub-tasks belong to, since these were given with more fine detail. The PO can also examine the discrepancies (if any) and acquire important knowledge to apply in future projects;
- **Adopted Work** – This informative metric outputs the sum of the estimates of stories that were inserted by the development team into the sprint backlog (after obtaining clearance by the PO) after the sprint started, meaning that some work was not properly defined and additional issues were used to complete the goals of the sprint. But this is not the only case, the PO can also insert issues on the sprint backlog, after discussing it with the development team. It may happen for various reasons that result in the necessity to do additional work, although this will lead to the dropping of some other issues;
- **Total Time Spent** – In this informative metric the PO can have a complete cumulative sum of the work done by the development team in the sprint. In a perfect world it would be the sum of the **Committed Work** with the **Adopted Work** metrics, but that is very rarely the case. This metric will assist not only the PO, but also the development team to help determine what went wrong with their estimations;
- **Time Estimate vs. Time Spent** – With this comparative metric, the PO can assess the difference between the sum of estimates given by the team and the sum of what was actually spent in the development. This control metric allows the PO to determine which issues are typically under-budgeted (or over-budgeted) and act accordingly, by discussing this with

the team. The PO also obtains important information that can be used when giving the Original Time Estimate to future projects that have similar issues;

While most of these metrics can be used for controlling purposes to ascertain the status of the sprint, they serve mostly as a learning tool, since they are mainly obtained at the end of the sprint, when conclusions can be drawn. For actual and "real-time" control the PO has the dailies and specially the burn-down chart, described in the next chapter.

With the results obtained from these metrics, the PO can motivate the development team to perfect their work or to celebrate what was achieved.

3.3.3.3. Usage of the EPM

Regarding the monitoring of the sprint's progress, the PO has the burn-down chart in JIRA which is a tool that displays the working hours being logged by the development team on the stories of each sprint.

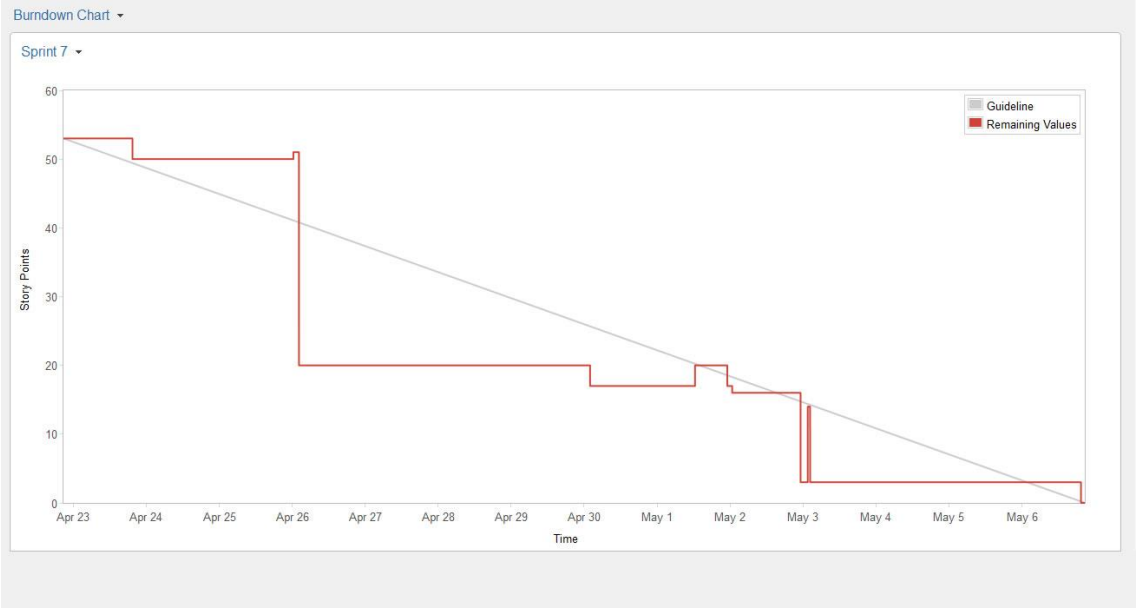

Image 9 – Burn-down chart

If the indicative red line goes up the grey diagonal, it means that the work is getting delayed and the goals of the sprint may be in jeopardy. Here the PO has two options: either go directly to the team and talk about the delay, or attend the daily meeting to try to understand the reason of the delay. Afterwards, if necessary, the PO can act upon that information, as described on the process model.

The information collected in this process will also be stored in Confluence and has the following structure:

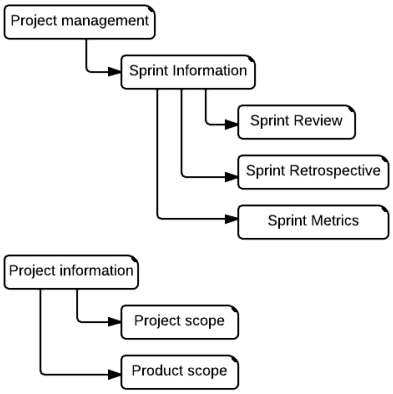

Image 10 – Information structure of the Development process

The first page, Project management holds the project information regarding the sprint.

- **Sprint Information** – On this page, the PO should insert in a tabular form the general information about each sprint, gathered along this process – its goals, start and end date and other relevant information. The financial department can use this information to bill based on the terms of the project's contract;
- **Sprint Review** – This page stores the information about all the sprint reviews that are made at the end of the sprint. At the sprint review, the team members discuss how the sprint went and point out issues (if any) that could affect future sprints or even the final product, that need to be addressed and by whom should they be addressed;
- **Sprint Retrospective** – The information on this page is gathered on the sprint retrospective, which takes place after the review, in which the team members reflect on events that happened in the sprint, how to improve the development of the project and what was learn in the sprint, for instance what was correctly or wrongly executed, other ideas to test and even praise someone that did a really great job;
- **Sprint Metrics** – This page displays all the information the EPM system collected in the sprint. This information is presented by the Scrum Master or by the PO who curates the data into perspective with other sprints to

allow a critical outlook of the team's performance to give room for improvements;

- **Project and Product Scope** – These pages are stored under the umbrella page **Project information** created on previous processes. Each time there is a change in the scope, project or product, the information should be inserted on this page, with the reply given by the stakeholders accepting the scope change. Although a project scope change could happen, that would mean that the requirement assessment failed considerably and may lead to a major redesign of the project.

After the sprint is done, the PO can decide if the current stage of the project is significant to demo to the stakeholders. If so, another process is set in motion, the **Demo** process.

3.3.4. Demo process

When a sprint is finished, the PO needs to determine if a demo with the project's stakeholders is necessary, meaning there are functionalities or information that the stakeholder can view and give additional inputs to improve the product.

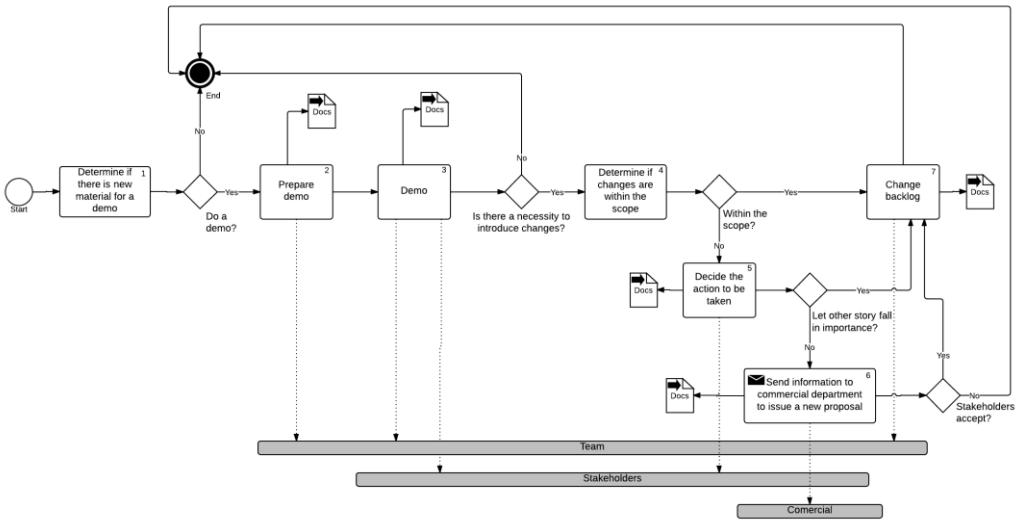

Image 11 – Demo process

3.3.4.1. Process details

Up front, on the #1 task the necessity or not of the demo is established. The PO should look into the goals of the sprint, as well as the goals and results of the previous

sprints (if any) to help determine that necessity. In case there is something constructive to show to the stakeholders, the PO, together with elements of the development team, prepares the demo (#2). This demo is an extension of the demo the team had done for the PO, which here as the role of presenter. By doing the demo, the PO will gain a holistic view of the project: its status, any difficulties that may not have been previously discussed and determine the necessity of change. These insights occur just at this stage, because in the sprint demo, the PO only had a feature oriented view of the project.

The demo of the product must (if it is technically possible or feasible) be prepared to be presented in the same server used at the end of the sprint demo. Ideally this should be an external server because the demo with the stakeholder usually takes place outside the company's office, or by means of remote session. This way, it should be possible for the stakeholders to interact with the product being demoed. Any information that is gathered at this task should be stored in Confluence.

At the next task (#3), the demo takes place with the stakeholders. Elements of the development team can be called in to help with the demo. With it, the stakeholders have a better grasp on the progress of their product, and at this point, may require some changes, improvements or new features. If that is the case, the PO must analyze whether the changes are within the scope of the project (task #4) elaborated in the **Definition** process or not. If they fall within the scope of the project, the PO should input those changes onto the project's backlog at task #7, make the necessary adjustments to the sorting order of said backlog and inform the development team of those changes, to allow them to analyze the changes and add their estimations (RTE).

But if the changes are beyond the scope of the project, it is necessary for the PO to decide (#5) along with the stakeholders what action should be taken. One option will be the insertion of changes on the project's backlog resulting on the drop of some other story instead, allowing the process to proceed to task #7. If not, the commercial department (#6) must issue an addendum to the cost of the project to accommodate the additional change. Only if approved, are the changes added to the product backlog on the task #7.

3.3.4.2. Metrics

- **Number of Scope Changes in the Project** – After the demo, it is valid that stakeholders determine the necessity of change. The information of these

changes is recorded in Confluence with a correct contextualization, making this informative metric useful to help improve the requirements' assessment phase in future projects. The information gathered for this metric must be inputted manually.

The metric described above can be used for controlling purposes for the project at hand, but also for learning and improvement, since the PO can ascertain from the changes in the scope what were his / her mistakes in gathering that information in earlier processes and improve that in future projects.

3.3.4.3. Usage of the EPM

In this process the information collection is mostly done on Confluence. If some changes are approved, they are added to the project's backlog on JIRA.

The structure of the information that results from this process should be in the following manner:

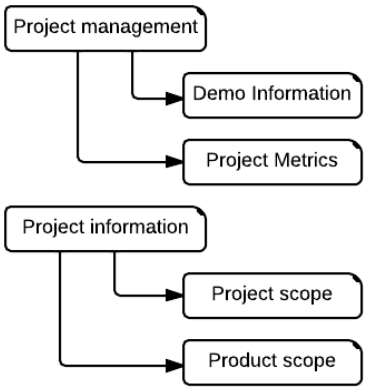

Image 12 – Information structure of the Demo process

The first page, Project management, holds the project information regarding the demo information.

- **Demo Information** – On this page, the PO should insert in a tabular form the general information about each demo that is performed for the stakeholder. The information stored can be the date of the demo, access data to the platform, etc.;

- **Project Metrics** – This page displays the project metrics information for this process. This page also holds the information of the metrics described in the previous chapter;

The third page, Project Information is the same holder page referenced on the **Discovery** and **Definition** processes. For this process, the important pages are the **Project scope** and **Product scope** pages (see chapter 3.3.2.1 for the differences between them), that have been defined in the **Definition** process. These are only relevant if there are changes approved by the stakeholders and the PO regarding the budget. If such is the case, the information regarding the change should be inserted in the appropriate page (depending if the change is on Project or Product scope).

When this process is completed, and no more development sprints exist (that lead to the completion of the **Development** process), the project comes to the last process, the **Delivery** process.

3.3.5. Delivery process

The last process, called **Delivery** is the final process of the totality of the project or of a current phase if the project has been divided in phases. It occurs when the product is ready to be delivered (deployed). In this process it also occurs: the elaboration of technical documentation and the training session (if such components were sold); the project (or phase) review to wrap-up the development in which an analysis is done on project evolution and outcome and what could be done to improve the efficiency of the processes (either of project management or development in nature). The process is depicted below:

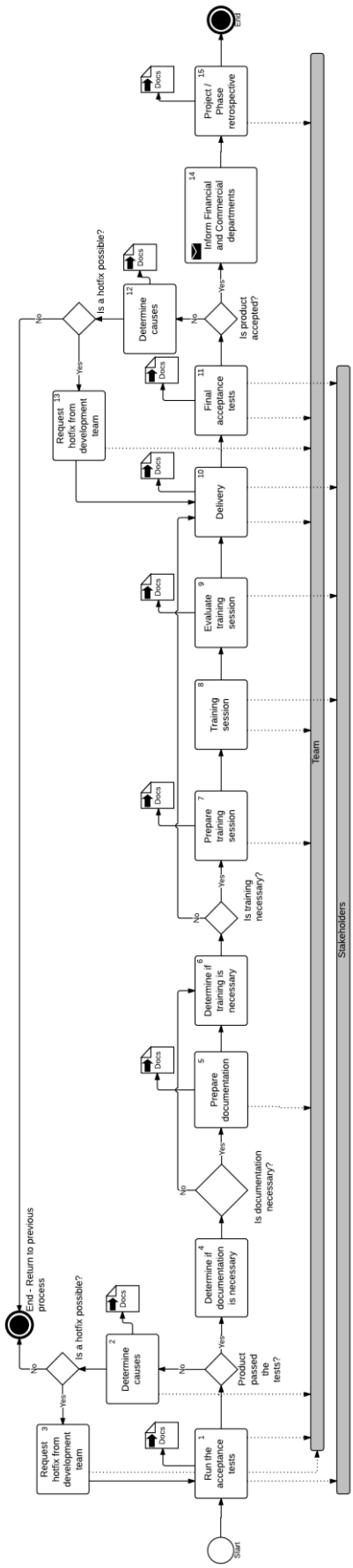

Image 13 – Delivery process

3.3.5.1. Process details

In this final process, firstly it is necessary to perform a final check (task #1) of the product on the pre-production server used for the demos, using the acceptance tests defined in the **Definition** process. These tests are created in a JIRA component, the Zephyr, which can store the information of the test status and provides a dashboard with the relevant information of the test cycle. If any of the tests failed on task #2, the justifications are collected on the EPM system. In the case a hotfix (quick fix done by the development team) can be implemented without the necessity of a new sprint, it is asked of the development team (#3) to do so. Otherwise the process ends and the project returns to the **Development** process, since another sprint / other sprints of development must be done. On these tasks, the development team and stakeholders are involved to validate the results of the tests and possible errors.

If the tests have all been successfully passed, the next task (#4) is to determine if it is necessary to write documentation for the product (if that deliverable was sold). In positive case, said documentation is elaborated in task #5 with the collaboration of the development team, as they possess much of the knowledge of the inner workings of the product.

If no documentation is necessary, the flow passes to task #6 where it is determined if one or more training sessions were sold. In positive case, they are prepared (#7) in conjunction with the development team. The training session(s) (#8) can involve one or more stakeholders, as well as elements of the development team, which can even be tasked to present the training session themselves. Afterwards the stakeholders evaluate (#9) the session to rate the quality of the training, in order to improve it.

After the training session, or if the session was not necessary, the actual delivery (#10) of the product on the production server takes place.

Afterwards, an additional testing session is employed (#11), using the same acceptance tests, this time in the production server. If the tests are not met with success, the causes must be determined along with the development team (#12), an analysis must be done about the possibility to develop and deploy a hotfix. If this is not possible, then the process ends and those problems must be quenched in a return to the **Development** process. If a hotfix is possible, then its deployment (#10) is requested to the development team (#13) and the process goes through to task #11 of the final acceptance tests.

When all of the final tests are met with success, the PO should inform (#14) the financial and commercial department of the delivery of the product so they are able to continue their own processes.

The final issue (#15) represents a final project / phase retrospective involving the development team and the Scrum Master, to reflect on the progression and results of the project / phase. This way it may be possible to transform the tacit knowledge gained on the project / phase to a more explicit knowledge in the form of improvements in the processes, whether they are of management or development nature.

As depicted by the diagram, documentation is produced on every activity to quench the problem identified in the interviews: lack of documentation and communication. While the agile manifesto upholds that documentation should not be a focus, it is still much necessary to guarantee the safekeeping of the information.

3.3.5.2. Metrics

- **Number of times the acceptance tests failed** – With this informative metric the PO as well as the development team can have a better grasp on how many times the pre-deployment acceptance tests failed. This matters to improve not only the development methods (of their internal tests), but also the acceptance tests the PO does at the end of each sprint to ensure the success of the delivery of the product;
- **Most typical causes of acceptance tests failure** – This informative metric will gather all the reasons for failing in the initial tests, making it possible to determine which are the most typical causes to prevent them from happening in future projects;
- **Number of times the final acceptance tests failed** – After the delivery, another batch of tests must be done by the stakeholders. By using this informative metric, it is possible for all the parties involved to determine the quality of the product. The measurements for this metric must be inserted manually on Confluence, as the tests are performed outside of the EPM system;
- **Most typical causes of final acceptance tests failure** – As in the previous metric, this informative metric can be used to unsure that repeating causes that occur after the product is delivered can be quenched and

prevented in the future. Similar to the previous metric, the measurements must be inserted manually on Confluence;

- **Ratio of times the acceptance tests failed versus the times the final acceptance tests failed** – This comparative metric will allow determining if the final tests are having a superior number of failures than the initial tests. With this information the development team and the PO can ascertain the cause and try to mitigate the problem. One cause could be that a technical requirement was not gathered in the appropriate process and, without that information, the error occurred. Because the measurements of the final tests are inserted manually, the measurements for this metric must also be inputted in the same manner on Confluence;
- **Statistical analysis of the evaluation of the training session** – With this information the PO can have a better understanding of the effectiveness of the training session and can use this information for future improvements.

Like in Development process, these metrics are transversal in their nature, since it is possible to use them to control the level of errors being done in the process, but also they can be used for motivation purposes based on the results obtained. These metrics are moreover capable of permitting improvements on future projects.

3.3.5.3. Usage of the EPM

The information collected on this process will be stored in the EPM system in the following structure:

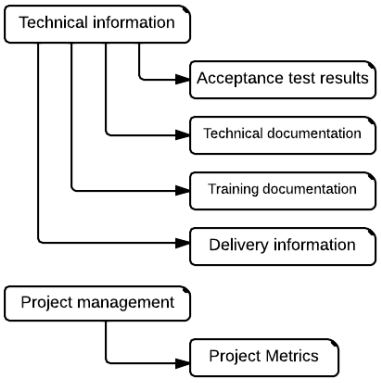

Image 14 – Information structure of the Delivery process

The page Technical Information, also mentioned on the **Definition** process, holds the gathered information on this process. The following pages should be inserted in Confluence.

- **Acceptance tests results** – This page should be used to store all the results of the acceptance tests done in this process, the initial and the final ones, in a tabular form. In case of failure in one or more tests, the causes should be pointed out.
- **Technical documentation** – This page only exists if it is determined that the technical documentation is a deliverable that was agreed to be produced. After the documentation is concluded on this page, Confluence allows exporting sets of pages as a PDF to deliver to the final client;
- **Training documentation** – This page will exist only if the training was an option sold in the project and should hold all the information and materials regarding the training;
- **Delivery information** – All the relevant information about the deployment of the product on the production environment should be stored on this page.

The Project management page, described in the previous processes, holds the **Project Metrics** page that stores the information about some metrics. Since the initial tests are done using a plugin of JIRA, Zephyr, the information is displayed in the Zephyr dashboard.

Also, it is worth pointing out in case the acceptance tests outputted errors, given that they were the results of an error of development, a new story should be created on JIRA with the relevant information to correct them.

With the conclusion of this process and the delivery of the product, the project or a phase of the project can be considered complete from the company's point of view.

3.4. Obtaining the Measurements

When starting a project, the PO should build the structure described in the previous chapters in the EPM. Nevertheless, the pages related with metrics information can only be built at the time the actual development starts, when the measurements begin and the metrics take meaning. Each of the metrics is presented on the Confluence application through the use of a special macro, entitled **JIRA Issues**. With those tables in place, metrics will be available as the data is inputted.

It is apparent that the EPM system, through JIRA and Confluence provides good resources to obtain the measurements of some metrics, but not all, especially the ones from the **Delivery** process that do not rely on JIRA issues, creating the necessity of manual input for the measurements. This issue will be further analyzed in the Limitations sub-chapter of the Discussion.

Also important to point out is that, while the implementation of the processes described above could be done somewhat easily, since the system that will support it is already in use, obtaining the data (measurements) necessary to fuel the metrics to retrieve some useful information is more difficult. It is necessary to change the working habits of the development teams for them to start inputting valid working times. To do this, verbal incentive must be present at all times by the already identified technological leader – one of the Managing Partners. But this role of champion can also be played by the PO of a project or even the Scrum Master assigned to the team, since they are in constant contact with the team and can encourage the acceptance of these changes.

On a conclusion note, by obtaining these metrics, the PO has a good starting point that enables him / her to determine if the project is a success or not. That determination should be done bearing in mind the strategic goals of the company, but mainly the goals of the projects, which vary from project to project and should be determined at the beginning of the project.

4. DISCUSSION

Upon completing the findings of this work, it is necessary to determine the implications it has on a theoretical level but also on a more practical scope.

For the theoretical angle, the linkages between the fields of project management, business process management and management of information systems are done. As for the practical discussion, considerations are done regarding the five processes that were modeled, their use in the projects and the validity proven by the use of the metrics.

Concluding this chapter, some recommendations, some limitations of this work and possible follow up works are presented.

4.1. THEORETICAL IMPLICATIONS

From a theoretical perspective, the implications can be divided in the three fields of this work. For the field of project management, this work presents several critical success factors that must be observed as a means to resolve the issues pointed out: **communication**, **requirements' assessment** and the **right amount of documentation**.

Given the agile methodologies depend on the tacit knowledge to be passed on face-to-face communication, many of the relevant information regarding a project can be lost. In order to obtain a coherent communication, this framework presents a set of rules on how to structurally store the information, while keeping in mind another critical factor, the right amount of documentation. These information structures are meant to support the PO of a project to make project related decisions, but also the development team.

Still regarding the critical factors, the requirements' assessment, typically done by the PO of a project, is presented in an agile way, a paradigm shift if it may, by involving the development team right at the beginning of the project, instead of just on the development stage. With this earlier involvement, the PO can use their tacit knowledge to obtain a more accurate assessment on what is required of the project and its cost.

A BPM framework was used as a starting point on the modeling of the processes presented on the **Findings** chapter. This highly iteratively framework emulates, to a certain point, the mechanisms upheld by Scrum. The BPM framework revolves around the notion that every process has room to evolve after an analysis of its results. After a new plan is drafted, an iteration of the process can be implemented.

But the analysis of the results of the processes is only possible if there are metrics feed by measurements to ascertain the validity of the work being done.

For that purpose, a series of metrics was devised, as well as the method on how to obtain its data. But given that the metrics are only usable if data is inserted (on the EPM system), a cultural change must occur in the recipient company to encourage the introduction of work-logs by the development teams. Since introducing this type of information will give a great deal of scrutiny to the daily activities of the development teams, some elements of the teams may feel suspicious of the purpose of such activities and might not insert the information accurately. To counteract the resistance to change, this work also recommends the employment of one or more champions. These roles should be assigned to people that are perceived not only as technological oriented people, but also leaders that can motivate others to understand that the necessity of change is also beneficial for everybody, being a step to improve their work and themselves.

4.2. PRACTICAL IMPLICATION

The framework presented on this work can act as a stepping stone for a company working on software development to improve the efficiency of their project management processes. While this framework was built using Scrum as a reference, it is just that, a reference and any other agile methodologies (such as extreme programing among others) could be used with little modification to the processes.

Also important to point out is that the information gathered to help build this framework was based on a company which had problems that were deemed common on most companies. However, before embarking on the implementation of this framework, an analysis of the issues occurring on the recipient company must be undertaken in order to adapt the framework and its processes. Likewise, this work presents the usage of a type of specific software, the EPM system by Atlassian. But this system should only be viewed as a reference, since there are several other suites that can accomplish the same result.

In the interviews done with the POs, at phase III, several issues regarding the management of projects were identified: Communication among all the participants of a project (ranging from the client to the development team), documentation scattered and unorganized, requirements' assessment problems and the difficulty to identify the status of a project.

To address such problems, five processes for project management were modeled: Discovery, Definition, Development, Demo and Delivery, already described in detail in the **Findings** chapter. For each process, it was devised a structure on how to store the data outputted from every process, for easy access of the documentation from all the participants that require it. Having a common structure for all projects will allow a more efficient flow of information, since everybody will know where to look for it. Also, making the information easily available on an EPM will allow a better communication, since the EPM should have the necessary mechanisms to alert others about the availability of relevant information.

It was identified that the EPM does not always respond to every necessity of a company, as such, some adaptation must be done to either the EPM or the company. This decision must be based on the difficulty of changing either of the components of the company's IS (**Processes** and **People** vs. **Software**). The same conclusion could be drawn regarding the usage of Scrum. Being a framework, a company could use only some of its components and "bend" some premises if it benefits them.

Additionally, metrics were also identified and described to allow the POs to have a complete picture of the status of the project. Nevertheless, since these metrics are targeted for measuring the development being made, it is not possible to have them in all the processes, specifically the first two: Discovery and Definition. This is because at these stages, no actual development is done, but only analysis and establishment of goals which are not suitable to be measured.

But the inexistence of metrics does not mean that these processes do not answer the issues identified. As pointed out earlier, the development team should be brought in to the project as soon as possible to help the PO obtain a proper requirements' assessment, based on their experience (tacit knowledge).

With these guidelines a company has additional tools to counter the lack of resources such as time, knowledge and money, which is a common situation among SMEs and thus can permit them to strive in their market.

4.3. LIMITATIONS AND RECOMMENDATIONS

While researching for the topics of this work, it became apparent the lack of a thorough research on the impact agile methodologies could have in SMEs and how they are being used by those companies, a topic that could have helped in the making of this work. Being SMEs the bulk of the companies operating worldwide,

implementing this study in the future could bring a great insight on the current state of the art in software development in SMEs.

In terms of the premise of this work, to standardize the project management processes, the scope could be expanded in an effort to implement a standardization of other processes, as the actual development. Nevertheless, this is a greater effort (but not impossible) since the implementation of software has many variables and each technology has particular aspects to it, requiring a great deal of analysis before embarking on the modeling of these processes.

Taking an example from the subject company of this work, the choice of the IT system that will support these changes is of great importance, to check if the information needed to manage the projects is easily obtained from the system. It became apparent that to obtain and use the metrics to assess the validity of the work, the DRI's EPM system alone was not enough, since some of the metrics are only obtained through manual labor. The data is all there, but there are not automatisms available to deliver them how they are intended. Furthermore, this is not a case where the metrics and related processes can be changed as perceived on the EPM system, since the system does not provide by default many metrics to evaluate the work that is being done (which lead to the development of metrics described in this work). Using the system as is will force the managers to continue to use excel worksheets, a lengthy and manual operation that involves collecting all the required information leading to much time being consumed and sacrificing availability of said information.

A solution for this problem, on DRI, was the development of another layer deployed on top of the EPM. This new layer can communicate with the EPM system though its API to automatically gather all the required information to feed the metrics and thus provide valuable and timely information to the project owners and other managers. This solution is best suited for the company since said API was already used to integrate the EPM system on other tools used by the company. Another solution could be to create a plug-in to the EPM system, but since this system if written in JAVA and the main programming language used on the company is PHP, it was not a viable option.

These shortcomings further highlight the necessity to perform a careful analysis of which system to implement to support the changes on the company. If there is already an EPM system in use and issues of the same nature as the ones presented in this work arise, it will be necessary to determine the course of action: can the EPM be adapted or should it be substituted by a new one. It is almost certain that no EPM (or

similar system) will fit perfectly with the processes of a company, so an analysis must be conducted comparing the cost of changing the existing EPM and the cost of adapting a new one. In the case of DRI, a great amount of time was already spent with this system, its users are used to it and making further customizations is a far more cost effective solution for the company. This is a specific solution, specially indicated for the subject company, DRI.

While this work presents several metrics to help quench the problems identified in the interviews and respond to the critical factors pointed out in the theoretical background, more metrics could be implemented. One specific metric could be created to help the PO determine what the rate of completion of a given component is. But since the EPM cannot make calculations of a specific group of stories, this prevents the collection of measurements to feed this metric, rendering it useless. This situation can be resolved with additional development as already discussed.

Armed with this framework, the management should draw its own conclusions and implement the changes it considers aligned with the company's goals and strategy. One option is to embark on a pilot testing with one PO and its projects to validate the implementation of the guidelines in close collaboration with the top management. Afterwards, with lessons learned the method can be expanded to all POs and their projects.

Finally, it is worth pointing out one topic that was proposed in a reference for this work, authored by Bruque and Moyano (2007). This work proposes the implementation of a quality system to further help the information technology adoption, to work as a facilitator in the adoption of change but also to provide other goals. For the DRI example, it could represent another tool to achieve a long-term goal that is to have the company's project management processes certified in an ISO standard.

4.4. FUTURE WORK

After the conclusion of this work, the next step could be the analysis of the results the changes brought to the subject company, DRI. With this analysis, a refinement could be done not only on the processes to further improve them, but on the information structure and the metrics being used. The analysis could also answer if the new developments solved the limitations present in the previous chapter.

Another analysis that could be done is the role agile methodologies have on larger companies, whether they are being used at all and if the processes proposed by this work could be used on such companies.

5. CONCLUSIONS

Collaboration! This work set out with a different idea, efficiency. But what was learned from the analysis of all the references collected is that for any effort of this type to succeed, in fact to achieve efficiency, collaboration is paramount. One of the topics discussed on the theoretical background revolves around this, to get the input of everyone, to have champions to invoke that idea among the workers. One example is with the case of BPM, in many of the references it is stated that BPM cannot guarantee success on itself. Changes are necessary, to transform the mindset of the company, but not only the workers, the top management as well.

Communication is also a key factor in all the fields discussed in this work. Without it, the information does not flow and goals are not reached in its fullest. Moreover, information is of great importance for every company and having it promptly available contributes to improve communication, but also permits a better sense of collaboration between everyone.

Of course a company does not strife on collaboration alone, and through the use of the guidelines discussed on this work, it has an essential tool in the effort to increase its productivity and reduce unnecessary costs, maximizing its revenue.

Just by realizing that change is necessary, the transformation is already happening. It may be a long term project, since changing mentalities and processes is a time consuming endeavor, but it is a necessary step for the path to achieve the so much referenced efficiency.

6. BIBLIOGRAPHY

Al-Mashari, M., & Zairi, M. (1999). BPR implementation process: an analysis of key success and failure factors. *Business Process Management Journal, 5*(1), 87-112. Retrieved from http://www.bus.iastate.edu/nilakant/MIS538/Readings/BPR%20success%20and%20failure%20factors.pdf

Ariyachandra, T. R., & Frolick, M. N. (2008). Critical success factors in business performance management—striving for success. *Information systems management, 25*(2), 113-120. doi: 10.1080/10580530801941504

Behn, R. D. (2003). Why measure performance? Different purposes require different measures. *Public administration review, 63*(5), 586-606. doi: 10.1111/1540-6210.00322

Boehm, B. (2002). Get ready for agile methods, with care. *Computer, 35*(1), 64-69. Retrieved from http://academic.regis.edu/dmmiller/courses/mscc630/r1064.pdf

Bruque, S., & Moyano, J. (2007). Organisational determinants of information technology adoption and implementation in SMEs: The case of family and cooperative firms. *Technovation, 27*(5), 241-253. doi: 10.1016/j.technovation.2006.12.003

Chow, T., & Cao, D. B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software, 81*(6), 961-971. doi: 10.1016/j.jss.2007.08.020

Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software Development, 9*(8), 28-35. Retrieved from http://www.pmp-projects.org/Agile-Manifesto.pdf

Hynes, N. (2009). Corporate culture, strategic orientation, and business performance: New approaches to modeling complex relationships. *Technological Forecasting and Social Change, 76*(5), 644-651. doi: 10.1016/j.techfore.2008.08.014

INE. (2010). Micro, Pequenas e Médias Empresas em Portugal - 2008. Retrieved from http://www.ine.pt/ngt_server/attachfileu.jsp?look_parentBoui=91762214&att_display=n&att_download=y

Jackson, S. (2011). Organizational culture and information systems adoption: a three-perspective approach. *Information and Organization, 21*(2), 57-83. doi: 10.1016/j.infoandorg.2011.03.003

Jarrar, Y. F., Al-Mudimigh, A., & Zairi, M. (2000). *ERP implementation critical success factors-the role and impact of business process management.* Paper presented at the IEEE International Conference on Management of Innovation and Technology

Kennerley, M., & Neely, A. (2003). Measuring performance in a changing business environment. *International Journal of Operations & Production Management, 23*(2), 213-229. doi: 10.1108/01443570310458465

Koch, C. (2006). Change Management - Understanding the Science of Change. *CIO.com*. Retrieved from http://www.cio.com/article/24975/Change_Management_Understanding_the_Science_of_Change

Lim, M., Griffiths, G., & Sambrook, S. (2010). *Organizational structure for the twenty-first century.* Paper presented at the annual meeting of The Institute for Operations Research and The Management Sciences.

Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., Kahkonen, T. (2004). Agile software development in large organizations. *Computer, 37*(12), 26-34. Retrieved from http://cs.txstate.edu/~js236/201205/cs3398/agile_large_organizations.pdf

Linnenluecke, M. K., & Griffiths, A. (2010). Corporate sustainability and organizational culture. *Journal of World Business, 45*(4), 357-366. doi: 10.1016/j.jwb.2009.08.006

Misra, S. C., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software, 82*(11), 1869-1890. doi: 10.1016/j.jss.2009.05.052

Moe, N. B., Aurum, A., & Dybå, T. (2012). Challenges of shared decision-making: A multiple case study of agile software development. *Information and Software Technology, 54*(8), 853-865. doi: 10.1016/j.infsof.2011.11.006

Project Management Institute Inc. (2008). *A Guide to the Project Management Body of Knowledge (PMBOK Guide) - Fourth Edition* (Vol. 1). Newtown Square, PA: Project Management Institute, Inc.

Ravesteyn, P., & Batenburg, R. (2010). Surveying the critical success factors of BPM-systems implementation. *Business Process Management Journal, 16*(3), 492-507. doi: 10.1108/1463715101104967

Rising, L. (2013). *Carrots and Sticks: What Incentives Really Work?* [PowerPoint slides].

Rosemann, M., & de Bruin, T. (2005). *Towards a Business Process Management Maturity Model.* Paper presented at the ECIS 2005 Proceedings.

Schwaber, K. (1997). SCRUM Development Process. In J. Sutherland, C. Casanave, J. Miller, P. Patel & G. Hollowell (Eds.), *Business Object Design and Implementation* (pp. 117-134): Springer London.

Sheffield, J., & Lemetayer, J. (2013). Factors associated with the software development agility of successful projects. [Article]. *International Journal of Project Management, 31*(3), 459-472. doi: 10.1016/j.ijproman.2012.09.011

Silver, M. S., Markus, M. L., & Beath, C. M. (1995). The information technology interaction model: a foundation for the MBA core course. *MIS quarterly*, 361-390.

Škrinjar, R., & Trkman, P. (2013). Increasing process orientation with business process management: Critical practices'. *International Journal of Information Management, 33*(1), 48-60. doi: 10.1016/j.ijinfomgt.2012.05.011

Subramanian, G. H., Jiang, J. J., & Klein, G. (2007). Software quality and IS project performance improvements from software development process maturity and IS implementation strategies. *Journal of Systems and Software, 80*(4), 616-627. doi: 10.1016/j.jss.2006.06.014

Swanson, E. B., & Wang, P. (2005). Knowing why and how to innovate with packaged business software. *Journal of Information Technology, 20*(1), 20-31. Retrieved from http://terpconnect.umd.edu/~pwang/Swanson%26Wang2005.pdf

Terziovski, M., Fitzpatrick, P., & O'Neill, P. (2003). Successful predictors of business process reengineering (BPR) in financial services. *International Journal of Production Economics, 84*(1), 35-50. doi: 10.1016/S0925-5273(02)00378-X

Trkman, P. (2010). The critical success factors of business process management. *International Journal of Information Management, 30*(2), 125-134. doi: 10.1016/j.ijinfomgt.2009.07.003

Wohed, P., Aalst, W. M. P., Dumas, M., Hofstede, A. H. M., & Russell, N. (2006). On the Suitability of BPMN for Business Process Modelling. In S. Dustdar, J. Fiadeiro & A. Sheth (Eds.), *Business Process Management* (Vol. 4102, pp. 161-176): Springer Berlin Heidelberg.

Wymenga, P., Spanikova, V., Barker, A., Konings, J., & Canton, E. (2012). EU SMEs in 2012: at the crossroads. Retrieved from http://ec.europa.eu/enterprise/policies/sme/facts-figures-analysis/performance-review/files/supporting-documents/2012/annual-report_en.pdf

7. APPENDIX

7.1. INTERVIEW TRANSCRIPTS

7.1.1. Interview 1

- Interviewer: Miguel Pereira
- Interviewed: Cláudio Calvão
- Date: Thursday, 8th of November of 2012

MP: When a project is starting, although these days, the processes are constantly in mutation, but normally when a project is delivered to you, from which areas of the company did it come? Is it from the commercial area?

CC: That's right, from the commercial area.

MP: But how is the information passed, is it a meeting with the commercial area where the information is passed, or there is directly a meeting with the client to determine the requisites or both or none of this?

CC: Ok, so normally there is the following. The projects can arrive to us by two means, or because the commercial area tells us that there is a necessity to do a pilot project, that will enter as a project that we must do in order to demonstrate to the client (as a proof of concept), or because we've done a commercial proposition and we must implement a project based on that commercial proposition.

MP: Like in a Requirement or something?

CC: That's right. The client delivered a list of requisites in a proposition. We do a commercial proposition. Obviously we have at that moment a series of meetings to get a better grasp of what it is intended and to give a rough estimate in order to have an idea of the cost of the project for the client. And if the client accepts, the commercial proposition is won and it transforms in a proper project. These are the two ways in which projects came to us. Afterwards, what happens? When the project is won, normally there is the phase of project managing, and that phase usually encompasses five stages, the one described by PMI (Project Management Institute), which are the beginning of the project, planning of the project, execution of the project, control of the project and closing of the project. And we are going to manage the projects in this manner. We are already doing this, although not clearly defined in this manner, as a process, but we are in the stage of defined them clearly to understand them. What tasks I have to do in the beginning, in the planning, in the execution, in the control and

in the closing of the project. Basically it is to determine a set of activities to be done, who is responsible by doing them to implement the project. Normally in the beginning of the project what it is done? Normally a meeting with the client is conducted where the scope of the project that has been adjudicated is presented, its restrictions in terms of time, a high-level planning, which is a planning that was already given to the client in the commercial proposition, that states that the project will take two to three months (for example) and it will have this sequence of activities, it can be done by sprints, it will be done in 4 sprints (for example) and the designation of the team that will work on the project and the identification of the stakeholders on the side of the client that we will need to interact. From there we will go to the stage of analysis, of discovery, right? Of course, we have an idea of what it is, but we have meetings, interviews and conversations to understand with more detail the problem ok? In order to afterwards build the backlog. A backlog gives us a broader scope of the project. In the proposition stage the client indicates that he wants this and that, and them when we talk with the persons it's not exactly that, it is a bit to the side and we need to register all that to understand that ok, from all that we gathered is the beginning (stage of the project).

MP: And at that stage can be the situation (the client says) that isn't exactly what we want?

CC: Yes. In this phase of beginning we have a kick-off meeting and we have a high-level planning and we have an identification of stakeholders. Then in the following stage, which is a planning stage, we are going to discover / analyze further, and in these reunions we try to grasp the actual scope of what we have to do and we create the backlog, and while doing so we can arrive to a conclusion that what the client says he actually wants in the interviews is bigger or different from the first proposal. For instance in the proposal we contemplated four epics: we need modules A, B, C, and integration with system Z. But then we are going to refine the assessment: in module A which are the features, the views, the validation and other requirements, and we can find we said ten, ten, fifteen and twenty days, for a total of fifty five days in the proposal but in that refined assessment we conclude that it is going to actually take seventy days, and at this time the assessment for an estimate is just made by the product owner, because when the team gets involved the estimates can widen or shorten. But it is essential at this stage that the product owner acts accordingly to the difference which was detected. He needs to say which backlog items aren't going to be done to comply with the budget or suggest and additional proposal to comprise the

additional items. This would be ideal, but these tasks aren´t done immediately, it takes time and there are several things that we can't assess at this stage, just after a couple of months

MP: Sometimes just after a couple of sprints, or even at the third …

CC: That's when we realize there are some tasks out of the initial scope, evidently as time goes by, it gets harder to negotiate with the client, if in the initial proposal there are seventy days he may or may not accept it. When he buys fifty five days and it's going to take seventy in the first reunions there's still a margin to negotiate or abandon the project, but after some time he is certain that the fifty five days comprise all his requirements, in the long run he won't accept the seventy days. This might happen, but it needs to be mitigated. At the planning stage, after these reunions we can have a more concrete planning, well if there's all this backlog, we start managing the time and cost of the project.

MP: At this stage it's not the responsibility of the commercial area…

CC: The commercial area after the moment of the creation of the project, disappears: It's the project area: the project owner and its team

MP: The commercial area is no longer involved with the project…

CC: No, not directly with the activities, obviously they might get called to reunions to discuss differences in scope, the can also be called to some reunions in order to diplomatically help the client when the relationship gets tense.

MP: They were the first contact…

CC: The proposal was theirs, and the clients trust them so they can mitigate problems. They can also be called just to maintain a courteous relationship, and in the end they can be called to access whether the client liked the team or not, to evaluate how the project was developed, in order to get more positive feedback for the team to gain trust for future projects.

MP: At what stages does the financial area get involved?

CC: When the commercial proposal is done usually there are predetermined stages of billing. These billing stages are set in the proposal it might be 30% in the adjudication, 30% when A is made or delivered, and 40% in the final delivery. It means the financial area will emit invoices of the percentage values at predetermined dates; obviously the first billing is made as soon as the project is adjudicated, and the information is passed along to the financial area

MP: and how is this information passed on? It is by e-mail or verbally? Isn't there an automatic response at the moment, to determine in which stage the project is?

CC: No, it's always communication, the team, in this case the project owner, who is responsible for the project management know we've achieved a determined stage and he communicates that to the controller or directly to the financial area so as to bill at a certain stage because we achieved a milestone or delivered something. Sometimes it is the financial area that sets out and makes enquires about the status of the project, so they can issue a bill to the client.

MP: So, do they have access to determine the state of the project or is the timetable set initially their guideline?

CC: They make a prediction because in the beginning of the project they try to plan when the different steps of billing will occur. When the project is adjudicated, for instance in January they ask for a prediction on the following stages, and we give them an estimate based on the number of days set to reach the milestones and they plan to bill accordingly, for instance two months to complete the first two tasks so they plan to bill in March, and in the end of the project estimated in four months, so in May we can bill this. Then they confirm at the different stages, and we have to explain to the controller if the times are being met or not. We should be able to explain every week to the controller the moving forward of the project at the estimate time or if it's sliding so the billing can be done accordingly.

MP: Is it expected or desirable the existence of a tool in the new platform to allow a better communication, so the financial area can perceive the flowing of the project?

CC: We aren't focusing in the tool as much as in the process to communicate. Whether or not we're going to have a tool that solves this problem I don't know yet. This could be done like this: inside Sugar CRM we have the opportunities module where we create our commercial proposals, there's the project module, which generates a project when the commercial proposal is won. And inside the projects, we could have, lets imagine, another module that corresponds to the billing stages where we would determine this first billing stage corresponds to 30% of the total, this is 30% and this is 40% and I could relate it to the timeframe, thus the financial area would only have to list the billing stages which were expected to occur at a given time. The financial area now has this in their financial maps and that is the reason they don't ask us about it. It would be important to have this, because then the project manager would be able to look at his/her projects and create pressure accordingly, and focus on those projects to overcome the setbacks which won't allow the billing. Besides, when the client asks about the project, the project manager could inform the stages of the

process and the corresponding billing. Many times the client asks the project manager how much is there to bill or what was already billed?

MP: They don't have that information, because there are several stakeholders involved…

CC: The product owner / project manager should be informed. There are other possible scenarios, imagine that we are behind schedule because the client isn't answering, then we can negotiate the billing accordingly; billing it 30%, 15%, 15%, 20%, 20%, 10% in order to achieve a more detailed project and to negotiate other milestones to prevent not billing 40% because the project isn´t reaching conclusion. Imagine that because of the client we are two years waiting for the billing of 40%.

MP: Indefinitely…

CC: But this is still just an idea, it would be interesting to have this information, there would be much more ….

MP: This platform will be integrated with Sugar…

CC: Sugar has the possibility that the commercial project and the project management from the view of… why do we put the project here? We put the project here to determine that the project has a budget of 18K, but it was sold by 20K, taking into account that in terms of development cost it takes 18K, but there was the commercial ability to sell it above, although that doesn't give us the right to take 20K, we have to do the development with the estimated 18K, but as it was sold by 20K there is a margin from the commercial area, but to manage the project it's 18K.

Then, there should be possible to create priorities in order to determine the most important projects from the many ongoing projects, and make them surface and focus on them as they could be essential. For instance they were low prioritized initially because you could end them early, but because of this they can bring more projects or take another project with other costumers or they are identical to others made. Thus we have to determine this degree of importance, so it's here on Sugar where we can define this degree of importance, in a commercial or marketing perspective.

MP: The time planning will no longer be done here on Sugar like we use to, it will be used just for financial analysis...

CC: Exactly. Budget and priority would be the main items but another important feature would be to determine the stage of the project whether it is active or not, for instance the project we have today on the board (PMO) whether it's in an analytical stage or in development

MP: The stage will be updated in Sugar…

CC: It will all be managed and updated here so that in the future we can have a digital board which shows the projects and their stages and that a log (a PMO log) can be associated to these projects. If someone from the commercial department wants to know how the project is going along, he goes to the board and looks at the log. He views the information that is collected on the PMO meetings and he then know the stage of the project. He can see indicators that indicate if the project is over-budgeted, under-budgeted, what is the percentage of the conclusion of the project, that should be feed (the information) through the size of the backlog, information that is on another platform, using JIRA and Greenhopper.

MP: You would have the conclusion….

CC: You have the entire backlog and the timeframes; you can collect the information about the amount of time remaining to conclude. Then I know the scope of the project and I can promptly understand the percentage of conclusion in the project. And this can be feed automatically if I have it updated, if there is a time change it would generate an alarm. It would be good to have deltas if from one week to another we change the percentage, the backlog increases, naturally. Did we negotiate this? The project manager needs to negotiate for more days or not. So we wanted to have all these indicators that allow management to understand how the projects are going, the deltas on a weekly basis informing if there are more or less things. Knowing that the artifacts of the tasks to perform, the times which are registered against the tasks that exist, and the amount of things to do (scope, time remaining and time elapsed).These are all managed here.

MP: This answer was very comprising and you answered a lot of questions, we have also covered the different stages of the project. The idea of the stages is they are always the same for all the different projects.

CC: Yes, the stages are the same because any project a beginning and an end, any project needed planning, big or small it needs planning. All projects have an execution and all projects need to be controlled. As far as project management goes this is true for all, but how much documentation, how much information you need to develop to have and show to the client I think we should devise a matrix. It's what we are trying to do next, that is to have a matrix that tells us if the project has less than 30 days, low priority and with a new technology… then what applies to this project is this set of activities.

MP: And the priorities?

CC: I don't know if these are going to be the criteria, there will be a matrix that shows several hypotheses, if the project is of this type, then we need to do this, it still has all the stages set earlier, but it states here you should have a kick off meeting and a team presentation, here you don't, you need very rigorous planning and you need to do such and such. The closure of these projects, for instance big emblematic projects, always has the presence of the commercial, the company's CEO and whatever, it needs some protocol, if it's a small project it doesn't: the team finishes the project, sends a mail to the client upon completion, they receive and e-mail stating it's all ok and the project is closed, for instance it can have some further costumer survey to the client to get feedback on the project, the quality, the team, the interactions…

MP: There is no idea on the metrics and clear divisions of the matrix items?

CC: It's being developed, what Ana Paula is doing is, as she knows the metrics that should be analyzed "One time value" and other indicators that normally are important to project management on the PMI methodology and the objective is that she tells us which are the indicators you should have, if this indicator is above this value then you should worry about this project because it's not in control, these are indicators of execution and control of the project. These indicators are, if the project as such a type you should finish like this, if there is another, then in a different way, because there is a lot of documentation you aren't going to make for a five day project and deliver it.

MP: It's about its value…

CC: It's not justifiable, only a big project… All the projects that surpass a certain value or have a certain degree of importance we have to deliver a checklist of documentation. These deliverables are a service we provide and that service is paid for that's why they are more expensive, we need to deliver them to justify the value. Another thing that is made, we have an estimated backlog, the people executing the tasks, all the team is called upon to do the planning, who creates this backlog is the product owner, saying my client wants this and I'm willing to pay this amount of time

MP: So each task is initially planned…

CC: Yes, he has already planned, then comes the team and in the sprint planning is going to detail, making a new planning, doing a breakdown, much more thorough and we can reach the conclusion about time estimates to each task and then, based on these estimates, if there are differences he needs to renegotiate with the client and he has to manage all these differences. While executing the teams logs the time and says how much time still remains, meaning the execution and control are always together.

When we perform a task we are executing control, because we are saying what's still missing, then everyday there is evidence…

MP: This control is made by the team itself.

CC: It's the team that says what's missing. The team has the notion that this should take fifteen days and today I spent one day, but there are still fifteen remaining, but after one day's work, there are only ten days left, because we advanced and the complexity diminished. Every day control indicators are available that say it's reducing or rising. It's in the burn-down (charts) that we see if we are destroying backlog or filling backlog. So every day there is control and execution, we plan in the beginning of each sprint, every day we execute, every day we have meetings to mitigate risks: obstacles, what were we working on yesterday, and what we are working on today. Every day we have control, because as we say what we have already spent in an activity, we are also saying what´s still missing these indicators are evidenced. So execution and control are hand in hand in each sprint, with this at the end of each sprint, the product owner is able to communicate to the client, my velocity on this sprint was such, so it's predictable that the project instead of two weeks is going to take three. Imagine that you said sprint one - ten days, two people with weekly sprints are ten days per person, this is what the team committed to do, but then during execution the team could only do eight days, this permits to do this: if I could only make eight, what am I going to do? Then I assume I can only do eight in each… and then I know, in the end, I need eight more.

MP: An additional sprint…

CC: Five sprints and I'm going to communicate with the client, based on the information people reported what they could do, what they weren't able to do, others, that weren't on the scope of that sprint, initially I committed to this, but only this is done

MP: And as it's communicated, it's renegotiated…

CC: As it's communicated and then all about managing the expectations. The client can say no, it's nonnegotiable, so we need to tell the team to make up for the lost time, the client is pissed, annoyed. The idea is this, the product owner discusses with the client and there's no problem, we go along at the same pace, but the team knows, we had planned to do in 10+10+10+10, but we are losing. Next the team, imagining the client agrees, and the team is committed to eight and does twelve, the average is ten so in the next sprint we say ten and we are going to be on time again. It can be done on a daily basis. The following day, someone is sick, we can only do four

what's the average on this? 24/3 – eight again. Always with this, we look at the velocity chart and say all we can accomplish we split by the number of sprints we can obtain an average velocity, and at each sprint closure we can say if the velocity was such, then it takes x time. Imagine there is a sprint when nothing is done, we started the sprint but there was no traction it gets <ero, when the sprint ends and we are going to be penalized because if we haven't done anything we had committed to do, then it is going to take a much longer time. Then the idea around this control is to give us indictors we can look at and tell client if it's going to be delayed or not. The product owner needs to understand how long will it take and negotiate with the team and if there is no margin renegotiate with the client, give him an estimate on how long it will take. Execution and control (repeats 3 times). At the end of the project, it's the closure, the ceremonies of closure: deploys to production, closure of all, finishing up the documentation and delivery of it, run the check list to see if all we said we'd do is done, have a final meeting with the client's team or send an e-mail if the type of project is such, wait for feedback, make a costumer's survey and then it's closed.

Another thing I forgot was, while managing this, inside these applications we can also do resource planning. We'll manage to have teams which are stable during a time period and we can say this team is doing two, three or four or one project, depending on the dimension, in the near future and so we can plan this team, this week is doing this project, the next week, they'll be doing another, then two weeks, this, the following two, that. So you can plan, and based on that planning, understand that if more projects are available but there's no one to hand them to, you need to speak to human resources.

MP: They need to get more contracts…

CC: There are more projects; we are missing resources, the teams are smaller, at the speed they perform and at the speed the projects are coming in, they are going to have projects on the backlog, they are only going to be able to deal with three months from now, because they need to close everything at hand.

MP: Then it's also difficult to manage that with the client…

CC: The project comes in, there is a first meeting to present the team responsible for the project, to collect and have an analytical stage, because then the team can continue to develop, if there is someone responsible to do this first part of analysis, it manages to work, because this stage needs to be done to build the backlog to give to the teams and work on it to break it down into sprints, but it can start going forward with something, maybe fifteen days or a month, but not three months and then it

needs to tell the client that, based on the planning of our resources, we will only be able to start this project three months from now. We are very sorry but…

If we have deadlines, we sometimes need to alter prioritizing. We have the order of projects, we can delay a project, renegotiate with that other client, to fit this one in the middle.

MP: There is all that management…

CC: Up till now we have included commercial department, financial, human resources and …

MP: Everyone involved. Now, for another part: the difficulties encountered in the projects, is there any idea of the nature of the problems. Are they all of the same type?

CC: There are several and from all sources. For instance, there are problems related to the client, for example the client does not give information, or which is complicated, whose objective is not to contribute to the success of the project, inside the companies we are extraneous to this, but have to deal with it; relationship problems between client and company, problems about internal inefficiencies: not being able to have estimated and ready backlogs, poor analysis and estimates, loss of team (leaving the company, becoming ill, producing below of the expected, because they're not motivated; there is people and motivation management…

MP: This people management is always done by the Project manager or is there any other way?

CC: Not necessarily, who worries about the team wellbeing is the scrum master, the role of the scrum master, his objective is in first place guarantee that the team has sufficient information and all it needs to develop their activities, if I have a problem, whichever it may be it is to be reported to the scrum master. Because what the scrum master wants is … I feel you can give some more, what is missing? I'm missing a machine, it's too old it's delaying me. Ok, that's a problem, we can make it better; let me see what I can do. The Scrum Master is there to help the team to overcome any problem that may arise, to guarantee the team delivers, in the maximum performance and tuned in. Thus keeping levels of speed stable (not irregular) and high. Better than high, in a first stage, stable.

Predictability, in order to be able to know we always keep this velocity, I like to know Christmas comes on December 25th, not be in doubt when Christmas is, this time… That occurs with movable holydays, when is Easter? We never know. And this type of predictability is good. You can count on that. People like to know what they can

count on, at the end of the month receiving a pay, sometimes not at the end of the month, at the middle, sometimes two months after.

Predictability is only that, if we are comfortable with it. Then it can be more velocity, let's work on that, as the teams become more experienced, we are going to have components we can reuse in other projects, our speed increases.

Communication problems, not knowing that there is something going wrong with a project until it blows on our hands. So indicators, information on differences of deltas on a daily basis, while people justify time losses and I'm still missing this and it's more than what I thought it was. If this doesn't flux information that tells you, attention we have a problem; then you can't manage efficiently, because you don't know, you think everything is alright. There is a need to get people responsible to concentrate all this and unblock, through communication, the fact that Sá (a team member) has entered to perform this control and to be scrum master at the same time is to try to understand what the team needs, what they want or need in the project or other things, then I'm going to pass that on, to the proper responsible, at this time he may not be able to be on that meeting but I am the keeper of that information, and that's my function. And if we can solve it, very well, we are all glad, everything is flowing. Guarantee speed. Guarantee the control and that this information is being properly recorded, because this information gives us these indicators and they allow us to act. It's her that needs to tell us, we have to bill this, we need this or that, we need more resources, because of the projects that are entering and we have the team completely allocated for the following two months. Controlling…

MP: all the variables

CC: It's not only on the project, but globally all the activity of the company.

MP: And specifically on the estimates, is there any analysis to the sprint planning and reviews over the estimated versus the real time spent?

CC: Every day. It's the control, monitoring and control. Every day we can understand if there are deltas comparing the time we thought would take and the time that we are executing and what's missing. There is a table here at JIRA…

MP: In Plan tab?

CC: Yes, I can't tell you the name exactly but this report is really nice, because it informs you that the sprint initiated with all these items on the backlog and these times included, remaining estimate, this one starts baldly because there are many at zero, all should have a value, and then every day as long as there is a…

for instance on the following day of the sprint start, the sprint started at 12.48 (It was its creation), at 5pm there was also an item being added, it doesn't have time associated, but it should have, because if there is this item still included, what's missing, and that would make a difference, a delta here at the estimated time.

MP: Then you can configure the sending of warnings?

CC: The objective is to look here and be able to understand by the charts… the project owner looks at this and understands we are working and the backlog is being destroyed, if the backlog increases, he says damn! We needed a day to understand there was still a lot missing. Then it wasn't a good planning.

MP: That one dropped right after that…

CC: Then he said false alarm, but anyways, it is information that needs to be properly controlled. Scrum is extremely demanding on that point. Every day you say I spent this, this is still remaining, every day you get information, every minute you log on time and saying what's missing you're exerting control. Because you're permanently saying, yes, we are better now because the time spent advanced the project or no I've spent this time but there are still so much more to do. Every time you log on time, you do this management. It's very demanding…

MP: But it's not evaluated if there is a certain type of activity which is usually underestimated.

CC: Not at this time, but these are metrics that may be inferred in the future. Because then we can make component analysis and conclude, normally these components it stretches, we estimated such and it slides. After the project, we can ask: what were the things that essentially slid? It was this user story, this and this.

Then we look and ask if this user story is always the same, why did it slide? A new component, we had never done anything like this, and there was a spike to investigate, we grasped something that we had never understood, we learnt and we hope in the future we can do this. Start to know, even which person, because each one estimates in a different way, understands the time he thinks he's going to take, but only after a while does the person start to understand exactly how long he takes to do things normally, and starts being much more assertive. For example, two days to do this, tomorrow you are going to do the same and say two days to do it, and in fact you only take one day after some 4 or 5 iterations you can say one day to do this. Because you have analyzed and understood, that it always takes a day. As it becomes all the same, you can be much more assertive on your estimates, this is brought by experience, time and the way the team works, the working process of the team,

several things. We can identify that people may need help or training, because they're working on components they shouldn't, because they don't have enough experience on that technology, we could have other people working on those components because they are more able, and we can schedule that components for two weeks from now, when we have the most expert person to do this available and if it is done at this time, it is going to be done by someone who will take a lot longer and it isn't worth it. And then you can negotiate with the client saying, our expert is only available two weeks from now, so we're doing this in two weeks, maybe it's preferable to let the project slide completely.

MP: And what about critical factors to the project's success? Normally when it starts, are they defined?

CC: Yes, that's one of the things that should be done over the first assessment; we have already found a company that does that. When we win a project, over the kick off meeting, give a questionnaire to the client to fill out in the beginning of the project, with questions such as: What are the criteria you consider more valuable on this project? Like: the estimated time, the quality and robustness of the final product, finishing the project within the predictable cost. Several indicators! The client manages its own expectations. He says "I value this and that" and then you can communicate to all the team, the PO to the team and to the Scrum Master, all understand what's important to the client. On the other day we had a meeting with Frederick, and Correia was saying 'I'm worried about this, because we can do this, in this way but we are going to have problems with performance; if we do it like that we are going to take the longest time, the problem of synchronizing data bases it's going to take a very long time to do, on the meeting we talked to the client which was Frederick and explained we can have performance problems here, how many clients are there? And other questions… this is going to hit Sugar(CRM) a lot and we can have problems. Frederick answered "No problem. If that happens: more machines, more processing capability, I don't think it is going to happen. For me that's not a current problem." We defined the expectation he said "I'm not worried about the performance, let's do it that way…"

MP: That information is inserted, it's…

CC: More than inserted, that information was communicated. Scrum doesn't predict we document everything, obviously it may be registered; send an e-mail with that information, that's enough and I save that e-mail attached somewhere in the issue or in the description of the issue itself it says "performance is not a issue" there it must be defined that over the acceptance tests a performance is not considered. But

there is not a formal way, the main aspect is communication, the registering of the evidence is a default.

MP: Ok, parameters. The main success factors isn´t still very developed, is it? The critical factors?

CC: No, that's something we are going to develop, all these systems you are talking about are going to be deepened, with Ana Paula, and they are going to be clearly defined and documented. What for? In order to have a process where clearly we know we had a commercial propositional that was closed originating a project, the first thing we are going to do is understand what type of project this is, consulting the matrix, let's create the project, while doing so we define the billing indexes, we create the project on this side: first stage the product owner needs to do this; second stage he needs to do that and if there are any differences he needs to communicate with the client… We clearly understand every stage and all the activities, the people who need to do what; when there are differences, which is perfectly normal, what should be done; in such a way we can manage the whole thing with all its variables, as we seen…

MP: They are many. Now you don't have a way to ascertain if the project was successful or not. There aren't metrics.

CC: No, that's a thing that doing the costumer survey in the beginning and in the end we can start to assess that. We want to put these metrics to the costumer survey and send it to the client "Did you like the project? Yes, No. What did you like the most about the project? Which are the strongest features among all the components of the project? The people, the team, the technology, this, that, what did you like the most? The time frame, the product's quality?" We can evaluate a global perspective in line with the initial expectation survey, in order to understand if the expectations were fulfilled or not. In metrics views we can say we have 100 projects a year and from those 80% have an acceptance level from the client's view as good, the relationship with the team is spectacular, then we can divulge these indicators and evaluate more than x% of our clients like our team, more than x% are glad with our work, we can manage to give that feedback to the market, that's what brings even more clients…

MP: To the market and internally…

CC: Internally, the team can understand the work… the recognition the clients are having of the work they do, externally the clients, or other potential clients will say 'Let´s work with this company because they are good'.

MP: These criteria, these factors aren't defined yet…

CC: They are going to be all defined…

MP: because there was no time, is it basically a problem of time?

CC: Yes, It is going to happen now…

MP: soon

CC: soon. These systems that we have been talking about are all linked. The way we are changing this is not let's just start here, because there are some themes we need to understand today and overcome today because they'll bring more productivity. Now we are going to focus on planning, here and here. We want to have estimated backlogs, completely estimated and that this value matches this value, what we sell to the client is the same as the backlog, if there is any difference let's immediately speak to the client, when we come here and we sit to plan the sprints we want to understand, are there any differences, if so let's speak to the client, and the team needs to pick this and work on a known thing, and every day they register time spent / remaining. This is our main worry. If this is in gear, then yes. Then how projects are brought to us? They are registered here. Example: let's add fields to a Sugar(CRM). This field is feed by a service, this field is feed by another and so on. Then at a certain time we are able to say: "we have the project here being feed by these indicators, based on time remaining, spent, and the deltas of the differences from one week to the other" and then we are managing to tackle the main problem which is the teams are producing, if we worry about this and the teams are here at the place, we don't have metrics, we have nothing, we don't know how things occurred. Then the questionnaire we give the client in the beginning and the one in the end they are going to be the last things to do, they are not the first. We basically already know the client likes to get things on time, he likes quality and the budgets are met, these premises we know that the client always wants them. Are there a lot to evaluate we don't evaluate now? Yes. We want to evaluate them in the future, but the priority is to steer the boat.

MP: Alright. Now, about the system we are using JIRA and the other components, we don't have a name…

CC: EPM - Enterprise Project Management

MP: Yes, I've just seen that name… Are the tests being successful? Do you think so? Is it a bet for the future?

CC: I believe so. The Nordics love it, they have given us feedback, they are loving it. The system is incredibly powerful; Ricardo and I are well informed about its abilities, we still haven't found the time to inform the people we are starting to do things this way.

MP: Is there a way?

CC: No.

MP: For example, yesterday we were asking where to load time and, this option or that.

CC: He told me this morning, and it is going to be stipulated, there'll be documentation on how to deposit time, after stipulating that, which is something that is important for us, part of the daily activity of everyone is the time registration and that plan needs to be rapidly clarified so that people start registering the time on the correct places. And the way we are doing it is… If we were going to define the whole process to load on to system it would take six months, and then it is preferable to load this, it happened in the same way with activecollab until we were on gear, you need to give training to people. There was that whole process but we were already using it, we have already sent it forward. I think that is an agile corporation, a company that puts itself through challenges. Let's start doing this this way, it trembles a bit, we are a bit lost, but six months from now, we are going to be better and one year from now, even better, two years, better even. The perspective needs to be this 'continuum improvement, what are we going to attack now? What are the main problems?' Then one of our main objectives we are trying to obtain is backlogs, all projects having an estimated backlog, even if it on high level: ten, ten, ten, fifteen, twenty, fifteen, twenty… There! It's estimated… Does it match with the number of days on the proposition? No. Then re-scheme to match.

MP: But it can't be from such a height….

CC: But it is. It may be. There's no problem, you don't have any commitment. What do you know today? If you do five, ten, five, ten, five, ten and it matches the number of days, perfect.

MP: But there is a risk of trying to match the estimates with the proposition…

CC: Yes, because I know, if I sold fifty days, it's because I need to do this in fifty days, now when we start breaking it down, we evaluate if it can be done in fifty days. If not, we renegotiate with the client. The client bought it for fifty, to say fifty we had some assumptions, then we need to present all the ifs. Sometimes the client says "I need a system that does such and such." This is his request, how do you know? You answer, telling him thirty days, but then he wants that and that. You answer: "What we had in mind for thirty days was such and such, not all you are asking now, what you are saying is different, if you had told me that in the beginning we would have proposed seventy or hundred or one thousand".

MP: Do you think…, well I assume this project is always ongoing?

CC: Yes, first stage: have a baseline.

MP: Are we already on the first stage? The second stage? What's the status?

CC: We are on … that's a bit hard to answer. We are well on the way, we already have all the applications installed and are using them, I could explain all this, because they are things we are already doing. It is not tuned; we are at a stage of testing and tuning…

MP: discovery?

CC: That I thing is always there, the same way we are working. Every time you do a planning for a set of functionalities in a sprint you are doing discovery. When you are going to break it down… You have a backlog and it is estimated. Yes! I have a backlog and I know this is what I want to do, did I go to every detail and refined it? No. There are some parts where I'm more advanced and there are others where I'm more distant, but I know what I want, and how much time do I have to do it all. That's why I can't tell you if we are on a first, second or third stage, because it is iterative. How many passages have we done? Several, we started by installing all the applications, we started reviewing the methodologies we are going to use, we've already done a previous study on project management methodologies and software development and now we need to tune it. We need to ask: "In what way does this adapt to our current model?" Assuming that our model isn't the correct one. There is still a lot to improve.

MP: OK. We have spoken about the tool, is there anything the tool doesn't…

CC: The tool needs to manage requirements, backlog and background of the tasks. The tool needs to allow communication and by registering things you are already communicating and by defining estimates and times (by the team and PO), you are already communicating expectations, you have the Confluence that allows you to describe and document e everything, you have the possibility to make project's documentation and analysis, glossaries inside Confluence, the possibility to make mock ups and diagrams. The communication platform is still missing, we have pretty much chosen one, but we haven't gone forward yet. But we are moving on, because for the reviews item, for the sprint reviews it is very important. As we do many remote reviews, for instance for the Nordics, we are doing a remote presentation we need to have a Webex where we can show, where we can record and receive those feedbacks after.

MP: Is that something that may be integrated?

CC: I don't know yet. We are trying to explore in what way it can be integrated. The process is integrated; I can say all sprint reviews done with remote clients, or with

near-shoring teams are done through an application; I can do a presentation and recording so I can pick the feedbacks and feed the backlog based on them. The testing tool: Two - "Zephyr and Bonfire". Zephyr permits to create test cases, test scripts and test cycles to allow building: functional tests, a cycle test for cross browser for IE, Firefox Chrome, to have a test cycle of regression tests which are when we change something, at least these ones we need to test, a test cycle of unitary tests of client acceptance or user acceptance tests. Then Bonfire ad hoc tests you can surf the application, the product you have just conceived, capture screen shoots and import them directly inside JIRA / GreenHopper. Make shoots with boxes and arrows and stuff around…

MP: Saying what?

CC: Saying what the problem is, it automatically captures operating system, browser version and a series of things…The testing part is covered. We covered requirements, developments, tests, documentation, diagrams, mockups, communication. I think we have all the tools and conditions to do the best work possible. I don't see this happening at the companies around us. I look at the companies around us, and see them working in the model in which DRI worked in the past, where the information was exchanged orally, there were no records and where people didn't have the notion if they are performing tasks that are really productive or being…

MP: They don't know the status quo…

CC: Exactly, what we ultimately want is a room full of screen across the office where we have all the information about our activity, all we do on these screens, it's something you look while you pass, and if you see a red screen you see that project isn't ok, something is up, can I help? We are having a problem in the support area, how many tickets arrived today? Can I help? More projects are arriving, like in the airport: the plane A, B, C are landing…

MP: It's important to have the notion of what's coming…All that at 'Torres do Oriente' right?

CC: Yes…

MP: (Laughter).

CC: Not there, there is a limitation which is you can only have one person by each 10 square meters, renting an office of 400 square meters you could only have 40 people there, that's what we have here. There's a limitation about emergency exits, so there's the limitation that each office only has one person for each 10 square meters

and it's a brutal cost… If anyone here wins the lotto and wants to rent the offices there I won't mind.

MP: Ok. It's all.

CC: Are you enlightened? Do you think my vision…?

MP: I already had an idea, but I wanted to have these themes to discuss with the professor and see, based on my research, but these weeks I arrive home a bit tired…

CC: You have the idea of what's happening and will happen. Ana came here to give these trainings and last week Ricardo and I had a reunion with her to try to understand what she thought about the teams and she said she thought the teams had a lot more knowledge than what she has observed elsewhere, when she gives Scrum training the people don't know, don't understand and never heard of it, she has arrived at places where they say "we do Scrum, but nothing like this".

MP: (Laughter)

CC: And she understood we have a predisposition, we are all opened up, one of the main difficulties on implementing these sorts of things is resisting changing, but she understood we are really orienting on improving and we all want something much more organized, more logical and predictive. Then what we agreed with her is that we will continue to do this training process, but on the job, tackling the critical points. We told her "we were going to try to guarantee we have backlogs prioritized and estimated and we want you on the planning meetings to observe how do the guys break down the problems and registers this information so that you can promote improvement on how we do that. We weren't used to it, to breaking down sometimes we think but we don't write I need to do this, and this, and this…" And it works because when you write everything, you only need to…

MP: Pass on?

CC: You have seen the documents you have being making for support. To me it's logical. If I need to pass the instruction of deployment to someone, I write all of them, and tell them where they are, you share…

MP: I even liked, I hadn't discovered yet, that I could put the checkboxes…

CC: This is done and this too and so is this… And I already have a plug-in that is 'Talk' where you can put comments in line like you do on google docs, for instance "I don't quite know this" or "attention to this". You can communicate with everyone and our real objective is to find a paraphernalia of things, not to substitute verbal communication, that is always implicit, but to register and save evidences about the things we do. Under that a secret project is running that is the collage of our process to

CMMI, to be compliant with maturity level II at least and one day be able to propose ourselves to certification. Why is it secretly running? Because it has terminology a bit different from Scrum and we don't need to know. You do the activity XPTO of whatever which is take a photo and put it inside the system. The value it brings even if we don't know how it is called on CMMI it's totally irrelevant. All we need to know is that we have a screenshot and we put the photos of our designs and stuff saved either on the issue on JIRA or inside the tree of the project and so on…

I think with this we are leaping forward in quality, I'm clearly understanding that six months from now over to a year we are going to be on a completely different register. You are here longer and you have observed, how we were a few years back, how we have been progressing and I think this what it does for us is to believe, this isn't being done by one or two, this is being done because everybody wants this and is moving there. We have accomplished to understand and believe this works. Then look, put things ongoing and continue to fight and wait. Essentially we need to wait, wait knowing we are doing something. Just waiting, knowing that nothing is happening, that we are doing nothing, no, that is demotivating. Sometimes it's not so fast, we need a longer break, then we go forward quicker.

MP: The way it's always been: one step at a time.

CC: It needs to be solid and sustainable, otherwise…

MP: OK.

CC: I hope I've been clear.

MP: Perfectly…

CC: I'm sorry about the delay on the conversation, but these have been crazy weeks…

MP: Ok, let me just…

7.1.2. Interview 2

- Interviewer: Miguel Pereira
- Interviewed: Dora Bailão
- Date: Friday, 29 of November of 2012

MP: The first question is: who gives you the information you need for the new projects? From the commercial area, is it you…

DB: From Claúdio, (laughter) basically…

MP: It doesn't come from Ryder, is it always Cláudio?

DB: No, 90% comes from Cláudio, existing business or new business that arrived through him and then he passes it on to me. Then if it is a project I am already working on like Tecnidelta, it's me who does the requirements assessment with the client, then I validate it with the assigned commercial, in this case Rebelo or Ryder, Diogo (Rebelo) on the knowledge and Ryder on the bureaucratic process of proposals. When it's really new, normally it's always Claúdio. I'm going to tell you about the cases, Tecnidelta is the one I've told you about, Verti I was presented the project first, this case was Cláudio, SIASP was Cláudio, the Fifty-Six I don't know if it was Claúdio or Paulo Mártires because they already had prior knowledge…

MP: Yes, yes…

DB: They present it to me, pass on the information… no from SIASP it was Jorge Teixeira da Silva who pass it on. They pass on to me all the pre-analysis work they had done and I validate it, ask for some more components and it starts in this manner.

MP: And then?

DB: When it's with the client, it's the client that gives me that information.

MP: In the ones you were already a part of?

DB: Exactly.

MP: While setting up the project, do you always start by asking that information to Claúdio? But eventually you contact the client and do your own analysis or do you base yourself in the one they get you?

DB: There is always doubt, then you have to talk to the other contact, it may not be the client directly, but someone responsible… how is it called, it's not the business…

MP: From the client's side?

DB: yes.

MP: Stakeholder.

DB: Stakeholder, exactly. It may be someone external and in the SIASP case it's Santana, then he speaks to Moutinho and they see internally with the client Engineer what's is name… So there are a lot of people. I also validate it internally with people who already knew the project inside. But the initial documentation is never on the level of detail we need, so we always need to pose more questions, your own team asks you questions and you need to get…

MP: the information, yes. And do the projects always follow those steps? Or is each project different?

DB: In everything in life, each project is different. There isn't a project 100% like other.

MP: You can never identify some steps that…

DB: No, the steps are global, but they always have different contours. Because of the people involved, for instance in the case of Verti there was a first requirements assessment done by Guerra, I think, and then I spoke to the client and did another requirements assessment myself.

MP: You did that requirements assessment yes. But then how Cláudio said there should be those phases: control and execution, not that much to you, that's more for the development crew, the control, the initiation and the delivery stages… inside those stages I don't know if you have that notion… If you think like that…

DB: Right now I was just focusing on what happens in the beginning…

MP: But what about along the whole project?

DB: The whole project…

MP: Are there any stages you identify?

DB: I identify, I identify myself with this way we are working now, in the sense there are several iterations and we are doing that on all he projects now. As I did in other projects, where the methodology was to show in each three months or right at the end of the development it isn't going to work, not to say bad words

MP: That's ok, Then I

DB: You filter the bad words… The following part on how we are doing it know obliges that, every two weeks or at the maximum every three weeks, you show something to the client. It's a way of having a routine…

MP: Also monitoring

DB: Monitoring what is being developed. And that, once again, is different from client to client, for instance Verti in some subproject we can do that really every two weeks, because we had the demos and we passed them onto their environment. With Tecnidelta there is every 4 or 6 months that we do small new features. In the SIASP case, every time we develop something we show them to validate it, just now I've received and e-mail giving me feedback on what I had sent them, that's why I took a little longer.

MP: You do the monitoring and at the end is it always the same way? How do you… You give it to the client and…

DB: Once more, it depends on the client, from project to project. I'll give you all the examples of the projects I've worked on. On Verti's case, we had an environment that we made available for them to test; we had different people with whom we spoke to; the demos we made throughout development were with key users no, the main

stakeholder, he validated everything, did the tests, etc. when he accepted, then we had a formation with the key users and they did tests and then we entered production and finished. They internally did more validation and tests.

In Tecnidelta's case, for example we do our test here; we make available a test environment, the client (Cláudia) tests it, sometimes she doesn't give the correct feedback and then we put it on production. Then who gives feedback are the operators, who give her feedback that Cláudia sends to us.

In SIASP's case, I still haven't put anything in production, but it's the stakeholder.

MP: It is going to be like that…

DB: We try to have at least one testing environment / quality and then the final – production.

MP: Beyond the communication you perform with the development team, besides that with what other departments do you communicate during the development stage, how do you do it?

DB: Internally, here…

MP: Yes.

DB: There's only one thing that periodically we are questioned about, the financial department normally when a preposition is made for the projects that are billed at the start, in other words they have fixed price there is a clause that says when the project is billed, x% in the beginning, x% in x months and x% sometime after and x % in the end. When that time comes, the financial department asks if they can bill the client or not. That's what I speak to them.

MP: Is it done verbally or…

DB: Mail and verbally both. With the commercial department always when there is a requirement that changes the conditions, for example the other day there was a problem with Tecnidelta's database, it crashed and the client spoke to me and to the technical (support) department and I also spoke to the commercial area to validate that that was contractually established and included on the service, so that I wouldn't offer the client a service that they weren't paying for. And thus I think I spoke about all the DRI's departments: Systems support, commercial and financial. Marketing, they speak more to me than I to them. Not the one you were thinking about, but for instance I rarely asked them something for the projects, they ask us more for their projects not the personal part you ought to be thinking about like newsletter and what not …

MP: Yes, but it wasn't all… that and for them to publicize the projects that were done.

DB: Yes, but they can also ask, as they already did, when they want a case test approved by the department.

MP: Now, maybe the most difficult question: What are the main difficulties you encounter on managing today?

DB: The projects?

MP: Yes.

DB: Be able to document, now with Confluence it is a precious help. But it was to be able to document conveniently the projects and I'll explain what I mean with conveniently document projects. Why is this? I have something that I need to describe, something I want to alert to, something I just want to point out, something to keep to show the client in the future and each time I had these needs I wrote them in a doc or excel or google share doc something like that and

Two things: The first: I'd lose time knowing how to organize my own laptop, what kind of information that was; they should be inside a folder called whatever for me to be able to find it later, then I'd lose time over the format, I need it to be pretty, Can I make it into a table, if I come here after three months am I going to recall how this was written? So, there were always two thorns in the side that prevented me to reach the goal more rapidly which was to effectively describe what I wanted to describe. With Confluence, that helps a lot because I can do basic formatting a la wiki, it doesn't matter it is always available to everyone, so I don't need to worry about where it is, I can search for it in any way that I will find it and latter I can export it to pdf or put onto a mail I can do whatever I want. I also can reuse that information to everything I want. One of the greater difficulties that I had since my other company and now here in the beginning was, the initial process in every project is always a meeting, minutes of the meeting, then a draft of the proposal, then the effective proposal and them the requirements detail and this is all information that before was only on docs, then the requirement was not that one, it had the number of the other requirement and them I had to write over many times the same thing, sometimes the text wasn't exactly the same thing, when I only wanted to add detail or readjust. That got in the way, because it was time consuming and it perverted the time I had to profound the requirement detail in each deal, now it is much better. I can reuse everything I write once written it may be linked to various places, linked Confluence to JIRA and it has been brutal. I

make better use of my time and the documentation is always available to everyone. I'm speaking most about the tools because they have been the greatest help…

MP: But your problem was documentation…

DB: In the sense that, I'd do it, spend time doing it and then what were the gains I had for doing it? I could get them, but it took too long to get them, because it was too scattered. They were the main problems. While you were speaking to someone, what was the version of the document? Was it X or Y? I can't open it, because I don't have docx, if it was in excel… send me that in pdf, now I can't see the table… send me that on an e-mail… A lot of time in crap, in troubleshooting information access and little time to analyze what you really want. At any time you can share, and the fact that you share doesn't duplicate the text.

MP: The information yes…

DB: When you do it by e-mail, direct e-mail, you wrote it once and then you hit reply and the people is already on another e-mail, fine it's all connected but you have that written a lot of times on the e-mail. When you share your text is only there, you only point out people to access and comment that information. The information doesn't get lost.

MP: That was your main problem, loosing and information management…

DB: and the time lost in all that…

MP: You've already answered to two questions in one…

DB: Great

MP: You are on half the time now…

DB: Great (long sigh)

MP: Are the estimates made for the development currently compared to what was done?

DB: with what was spent by the end of the project.

MP: Yes.

DB: Ah… I still didn't have the opportunity to do it, since we are using the tool, on the other projects I had never done that type of analysis, or I end up doing it when I spent more time, because it stands out, the question was just about analysis, is that it?

MP: Yes, analysis. If you do it or not.

DB: There are people who do it for me necessarily, while doing the bills they see how much time I spent. And now it is much easier to be on top of that sort of matter, because things stand out much more clearly.

MP: So before you didn't…

DB: Before I didn't do it. By my own initiative, not directly, the results appeared to me.

MP: And now with the tools available, do you think you will be able?

DB: No.

MP: That being, you cannot have any idea on what the activities that are underestimated…

DB: I have an idea, because erroneously or mistakenly, when you do the first proposal and when a commercial thinks about something to give to the client, there are several things he doesn't take into consideration. More and more I have been searching for those points and including them. And we reach the conclusion that a project of 100K, it's difficult sometimes for clients to understand it, you spent about 40 or 50% effectively on development, all the rest is what involves handing in the project, all sorts of management not only project management but managing the team, the issues, not just development…

MP: Yes all the communications, all those things and preparing…

DB: And then the other sixty aren't just this, it's all the testing, training and documenting. All that is nearly half the project.

MP: The next step is do you generally have defined the critical factor for it to be considered successful?

DB: The main features being available…

MP: Do you define at the start, which are the main features for it to be ok? This because I don't have as much perception about sugar projects… At a site, is the site being operational, but maybe it's not just that… At a sugar project it may be different.

DB: I'll give you an example, Tecnidelta, for the project to be working there are several modules and normally those modules are identified at the proposal, if the proposal says storage management, then the storage management needs to be available. Now are all the features present? That is a posterior breaking down. But for the project to be a success is, really, the users having a working storage management feature.

MP: Ok

DB: Easy. It's like that. Why does the client come to us to develop something?

He has a need and he can't do it some other way or does it in an archaic manner, we have to guarantee that what he couldn't do or did in an archaic manner, can now do with our product, and that's the decisive factor. At sites, however, I can give you Verti's case, there were several sites, actually it is an application in Drupal's case, and

they wanted, in the case of the occurrences, a portal where all the company could submit contents and have the HR department to manage those contents, from the moment they give feedback and they can manage that feedback it's success.

MP: Is there a compartmentalization of the points where the modules are installed to determine, for instance…

DB: The client doesn't want to know what modules are installed…

MP: That's not the question, from the features, for instance when it is estimated a certain time and actually the development requires a longer time, at least at Agile it is stated, then see with the time available what can…

DB: And that is something, sincerely in every project I've participated up till now, I still haven't had the opportunity to do and that is always going to depend upon the client's mentality and if it is closed at the beginning or not, if it's time and materials. And what happens always is that there are three or four blocks of features which are contemplated at the proposal and was accepted by the client, then you are going to detail, it is best to be very detailed to know what it contemplates, when it's not, you have to detail afterwards. There are clients that don't want to know more, you develop what you competed for and it's the project manager that guarantees that there is all that was supposedly sold, some other clients are involved and they ask for more and here there are two ways: one - it was not contemplated, it transfers to another project that you want to pay after, normally on the projects that I'm involved with, it's this that happens even in my former company. If it's out of the project's scope, we'd do a proposal of continuous maintenance and upgrading or a new project to which we would assemble those functionalities. Now I always try, if it is possible for the client, because they have closed budgets and it isn't always as easy or as agilely as Agile wants. What Agile says is ok you need to have all the features or he asks for one or one takes longer, the last one drops and then you have two ways or it dies out and it is never done or he pays more and you feed it bigger.

MP: Activities, success… So basically we can say that the projects we have worked on are all considered successes…

DB: In which parameters?

MP: By your parameters…

DB: Nothing is lost, all is transformed. Even a bad experience a project you considered went wrong, why because we spent longer or effort to develop and we lost money with it, because the client send it to the trash can, the several reasons why the project may have went wrong, you always learnt and the next project you'll do you will

be able to avoid those types of situations, avoid some confrontations. A project went wrong, because you aggravated yourself with the client and you had an argument and he went away, you now, for the next project, the limits you have to guarantee for that not to happen again, so to me in global terms all is success, better off, not success, with every project whether they are good or bad you will learn and that is always a success.

MP: And for those problems, they are always time problems or is there any other problems that conditions…

DB: No, it can be technology or lack of vision from the client's part or from our part, we don't understand something, that's not typified I guess.

MP: Do you have any idea on those problems? The time, the client's vision…

DB: Here, as you now, I have eleven years' experience from another company and I can give you several examples of things that went wrong and why. Here, from the projects I'm in and I can say something has gone wrong… When we say went wrong what do you mean? Is not working with them again or the client being unsatisfied? From Verti he was satisfied and hasn't continued, because they have no budget at the moment; Tecnidelta is glad and continues to works with us, there are some problems on the client's side and on communication with other partners that need to be done; SIASP there is no problem, Fifty-six the problem that may exist is the minor priority that has delayed the project. Now, other projects, things gone wrong in my other experiences: You have much effort on the project and suddenly the client ends their budget you need to drop it, it died, on another client the project is ok, fine, you deliver it, the client changed the director and he does it all again with another company, for you it's a success because you did it and delivered it but it's done…

MP: Did you get paid?

DB: Yes, it's the same thing if you buy a TV, ok I don't like it, it shouldn't be black, but it is white… bummer, your specifications said you wanted it black you want a white one, buy another… It went wrong; we didn't do anything the client wanted, is that correct, and is it written you validate it and you find the client was right you do it again or the client doesn't pay and we are still in good terms, it depends… But the question, I am talking to you about the negative aspects, but your question was about the positive aspects…

MP: No, it was: What are the main problems that lead a project not to be deemed as a success? It's exactly what you were saying.

DB: Success for whom? And what are the parameters? If you are from the financial area, the project is not successful if you have 0 or loss. The time consumed and the estimated was the same instead of being lucrative, "a cent earned, a cent spent", you got experience and notoriety, you can say you did that project, you can earn more projects for that client; the client is very demanding and it takes too long to pay. It depends on what's positive or not in the financial area, in the commercial area, you did it, it was ok, but it doesn't suit you to say you did it…

MP: Yes, this is just from a managerial point of view…

DB: In project management I think it is never a fail, it's a fail… I have already had clients, not here but elsewhere, we did it, it's in production, and the people are using it, but I never want to see that client again because he is very annoying. You can't say the project was not a success; the personal relationship may sometimes spoil the project management. It's too stressful. I forgot Controlinvest: we developed the things are in production, the client is a bit annoying but ok. So…

MP: So there's not a frequent problem…

DB: The recurrent problem to the projects is to put a break on what the client asks while the project is running. Because he always wants more, he wants everything and even if it's not contemplated if you give him a candy, he will take it, but he rarely wants to pay for the candy… If you don't put a stop to these demands, you end up giving without him recognizing he's getting it for free.

MP: So if you have a system to support you, it helps you on that…

DB: To support this, it's what happens with the product backlog, now I put it inside and ok "you say that" it's not inside the project (scope) but I understand your need, at this moment we are focusing on developing what we committed to do, when we finish this project we speak about it. Sometimes they forget and never speak of it again…

MP: (laughter)

DB: but it's a motive to collect all and at the end they pay for that one and then you remind him and propose, if he says yes you give him a budget and he accept or not…

MP: Ok, at the time this was made, JIRA wasn't very developed still. Have you already tested JIRA?

DB: Yes, and I'm loving Confluence.

MP: Ok.

DB: And JIRA too.

MP: I've used it now for the training and it came in handy. Do you think there is something that needs to be improved? Can you point out anything that needs to be improved?

DB: In what?

MP: In JIRA or Confluence, in the system.

DB: We are still in a learning period and knowledge; we have already detected some mistakes that are appearing in the reports.

(interruption - phone call)

MP: We were talking about problems that may be happening at JIRA…

DB: They were just those ones.

MP: What were they?

DB: The reports aren't reporting some things correctly, some things that you feel intuitively it should be capable to do but it doesn't, then you can do it in another way. For instance, a massive copy of issues from one project to the other you can't do, you have to copy issue by issue; a massive update, and things like that, but it's just details.

MP: Do you think you can still benefit even more on the way you can work with Confluence?

DB: Benefit more, yes, yes… and over time even more.

MP: Even more…

DB: Yes, I'm still a bit green, 20% of what I can take out of results.

MP: Do you think in the tool, in the EPM, there is something missing?

DB: I can't answer that because I don't know all of it, do you understand?

I have noticed we have some widgets / macros we can include and there aren't all there, so we need to search on the net which ones come in handy and get them. I don't know all of them; I thought oh this one would be good, this is better than that and that one could do that… but I cannot answer that with certainty from what I know. What I can tell you is my first comment about the Atlassian Confluence / JIRA was 'Uh! How ugly!' Because I did a lot of research and I was used to seeing things with a cuter graphic interface, but that's not to be cute, it's to be functional so really as cuter it is, the slower it gets too, normally. So that is exactly with the contents you need, when you need them, with the permissions you need, with the shortcuts you need and with the links you need, it can adjust and I really like to adjust windows to be able to see the most information I possibly can and it has certain link positioning that does it for you.

So you can have in your vision field all the information you need to solve a certain issue. We win by not being too complex, the simplest it can be. It could be more appealing, but it has what it needs and no more. It's great…

MP: So a tek session soon…

DB: Are you doing it?

MP: No, you are…

DB: (laughter) …. Done?

MP: Done.

DB: bye! See you tomorrow

MP: Thank you

7.1.3. Interview 3

- Interviewer: Miguel Pereira
- Interviewed: Paulo Mártires
- Date: Friday, 7th of December of 2012

MP: First question, well I'll ask… From the project management point of view, how does the information get to you? Where does it come from? Does the commercial area pass the information on to you? Or while the commercial area is dealing with the client do you go talk to him?

PM: It depends, you have both cases. There are projects that arrive pretty much already defined by the commercial area and the client, but these are rare, normally what arrives in an initial stage is very little information and most of the times the project is already sold. Then on the other hand, more and more even before a proposal is made, who manages the project or who contributes the most strategically in an initial stage is present during these first meetings and gets much more information at that time. Anyway we are always speaking about very global information, only from the moment when you start to do the project will you effectively pull out what you really need. And in the majority of the projects the information isn't given to you, you need to find it out. Because either the client doesn't know or nobody thought of it. In a general manner you never get the information you need in the beginning of the project.

MP: You go and get it directly with the client?

PM: Yes, it's usually so.

MP: So basically the flow of the project is commercial area, then project management and finally development?

PM: If you think of it in a normal project management perspective, it's like that. But in my case, I'm present, most of the times, when the project itself doesn't exist, yet. I access information at a stage when if I was merely a project manager in the normal flow I wouldn't. Increasingly less I'm taken by surprise.

MP: Do you think that is our way to do it, that you are there? Or are you seeing it from the UX perspective?

PM: It isn't about the UX perspective, it's really being called. More and more it is tried to do it like that: the teams, the project managers or strategists to be present at an initial stage of the project. Frequently the project doesn't exist, yet. We are only investigating and make proposal or gather information to do a proposal, but I think there has been an effort to try to involve the team at the early stages of the project. It isn't enough to have the manager or PO or whomever with all the information, you need to have the whole team or else it affects everything. I think there has been an effort accordingly. In my particular case I get involved in the beginning, because I care, but there is an effort from the company in that direction.

MP: So the project preparation, how would you say it is? How is the process of beginning a project? The meetings… Is there any material already made? For example at the "Aldeias" you have the information architecture, but that is in a more advanced stage… how do you do the initial setup to start?

PM: It varies in each project, in the case of "Aldeias" it's a different project, you know you have to do three components, you that are some obvious things that need to be a part of those components but the most of the initial stage of the project is to understand exactly what the problem was, apart from what the client thought it was and then have a strategic approach to come up with a solution. So the project's scope changed after investigating and doing a workshop, and understanding the problem wasn't such and such, but really that. So here there was a discrepancy in terms of the proposal, what they wanted was a store and a community and the portal wasn't all that important, but their main problem was really the portal. They need a step to get to people, so the things were inverted. But then you have projects where it is much more linear. You know exactly what you have to build, you may not know every little detail and discover them on that first stage or along the project. It varies a lot, there isn't exactly a model. And moreover by a time constrain and lately by methodology you don't need everything at first to

start developing. You try to have the essential; you need to know the way and the set of features that are the basis, but rapidly we can start working, while before we started working without knowing exactly what we were going to do.

MP: You define a guideline and then at each iteration you…

PM: Exactly…

MP: Break down…

PM: If you really like to be Scrum or follow a more Agile style you need to follow that path. You may have a discovery phase, you should have it, it can be more or less structured and based on that initial discovery stage; you may call it sprint zero or whatever; you build a more assertive backlog. And supposedly then the team has the conditions to start working, it's not exactly what happens. It doesn't necessarily happen like that all the time but the effort is to walk that path?

MP: Isn't there a structure for that sprint zero?

PM: Not yet.

MP: Not yet. But is it planned?

PM: In my particular case I'm thinking of it, because the web area has certain particularities that maybe Sugar and other things don't, but I think so it's being thought of.

MP: Cláudio spoke of all the steps that are in a project: the beginning… I don't have it detailed here at the moment… the second to last is the development and then the delivery and the follow up, I can't remember the first ones now. But the way you manage the project, are all the stages different? Is there anything in common?

PM: Yes, in a general manner all the project have the same structure. You have the initial stage of discovery, requirements' assessment, study solutions etc. Then you have the actual development: the design, the UX, the programming, the tests. If you are working in an agile style you don't do it, you don't test just in the end, they are being tested…

MP: Yes, at the end of each iteration.

PM: Exactly.

MP: Almost at the end of each iteration.

PM: Then supposedly, you have a deployment stage and there you perform measurements and the upgrades or follow up. It's one of those things, it depends on the way things are structured and sold. But to me, it makes sense that when you have a project, right in the beginning, you define what the minimum product is, or

what is essential you deliver that and you save time and money from the project to keep on making it better. What is normally done is you develop the project within that budget and deliver it and then the improvements, not a support, but the improvements are a new project, with a surcharge. The ideal was to make a budget, deliver the project under the budget and then save time to improve. From the moment you have the project online or being used, you can pull metrics and things to improve on, up until then you're guessing.

MP: But that way of the lifecycle of the project is always the same… independently from the level, I don't know how to call it, more or less complex or with a larger budget…

PM: At first, yes. If it's a project, not just move something quickly, you always have that. You always have the initial stage of discovery and preparation, then the actual development and inside it you may have projects with a large design components and other not really, you have others with more frontend, others still with more frontend and backend, but if you consider it to be development, all you are doing is production, all have exactly that, which is I have the discovery, the execution and at the end the deploy, tests and follow up.

MP: Ideally it is really to deliver the documentation and the tests …

PM: Exactly.

MP: Ok. I had the questions in three groups, from this group…

Besides the development stage, while you are in contact with the programmers, well you must always be in contact, right?

PM: Always.

MP: How is the communication with the remaining departments done? The financial or the commercial area… Is it done verbally, by e-mail, is there a support like JIRA or…?

PM: No, there isn't. It's done verbally and by e-mail. There isn't any other… well; there are meetings, follow up meetings…

MP: Finances?

PM: Yes, when you have the PO meeting there is someone from financial area there to see how the projects are going to perceive when they can bill.

MP: Ok.

PM: That kind of think, but otherwise it is done in an artisanal way. Sometimes someone asks about it and you say it's this or that…

MP: Do you think…

PM: We are waiting to close by this time. Usually there isn't much communication, with the commercial area then it's very rare, unless the client annoys the commercial, or the commercial needs to know a concrete item. With the financial area there is usually a contact when, from their part, they ask us if they can bill or if they can send the intermediate payment or something like that… But there isn't a process, it's not a regular thing, you're not always talking with the financial area to account for…

MP: Aren't the payments done at the end of each sprint or something?

PM: No, you have a payment plan which is agreed with the client and it varies accordingly to each project. There are clients that have motives, like for instance in the case of Aldeias, they need to make the payments in certain times when they get the money from the community funds. There are others, that it's in the beginning, then 30% more whenever and the rest in the end; or forty, forty, twenty. Or every month they pay X, I think it happened not that often. There are several models that can be adjusted to the client.

MP: Ok.

PM: But you don't deal much with that, as a PO you deal mainly with establishing a budget, obtain an execution schedule and your concern is that, to keep the schedule and inside the budget.

MP: After all that, what's the typical difficulty you find while managing projects?

PM: Here particularly it is managing times; you have a great difficulty knowing exactly how long you still have for the project. It's being mitigated, but it was a difficulty, because I remember that for a very long time we couldn't take out Sugar reports for the project. I don't know how long was already spent or I do the math by hand or I don't know. Even now to do it, it is almost by hand to understand exactly. On the other hand you have estimates done in a certain way, there are some things that aren't taken into account, there are hours that get in late, people have to fill out timesheets and sometimes it's a bit confusing and we don't understand exactly where have so many days gone into development, when in effect you know it wasn't exactly like that. The difficulty is to know exactly how long you have spent, how long you have left, in what stage exactly is the project. One thing is knowing you have twenty days but the project is 90% done and the other is knowing you have twenty days but the project is 2%, so … you start having mechanisms to understand but that's not very easy yet.

MP: Usually when people answer the first, they tend to answer the second: How can you mitigate these problems? It's being made. Is it with the support from the tool?

PM: Exactly, you have JIRA and everything you work on should have an issue and that issue must have an estimate time assigned and thus, the time should be automatically assumed and it should give you those metrics right away. It's not exactly so, but there is an effort being done in that direction.

MP: Yes, but these estimates are always more complex, they require a structure…

PM: The problem is that there is normally a difference, sometimes large, between the estimates to provide a budget in a stage you still don't know much about the project, but then it ends up being a reference for the time you have to execute it. And there resides the difficulty in marrying the two. Sometimes the project is way more complex and you need to do more things and it isn't possible or you have to fit in with what was sold. Someone sold 30 days, but the problem is that those aren't 30 calendar days but 30 FTEs, that in effect can be one week if you have six people working.

MP: So, accordingly to the way estimates are done, at the moment, you have no way of comparing the estimates with the real execution time by task.

PM: By task you are starting to have it, before there was no way. You had a very global idea by type of activity or stage. The design used up X days, the development used up whatever, but you had no way of knowing; ok to set up a CAS took about X time, to make the front end it took whatever, but now you start to have a way. You can understand that maybe your estimate which was sold and you made the project from doesn't match at all with the real execution. It's a question of understanding what failed: was your estimate incorrect? Or who was coordinating the implementation wanted to do more than what it was supposed to? It's always complicated. On the other hand, it's a question of orienting by the value to the client. Always trying to keep on budget evidently, and the question of what's good for the company but oriented toward the client. You normally have the tendency to put more than…, although you don't have time you deliver more in the best possible way, it makes sense for the project and the client; it's a question of vision. I don't take part in that, there are those who think we have a budget and if it fall under ok, if it doesn't though luck. We are not working to no avail.

MP: So based on… you don't have the possibility to see, yet or you do… Do you have any idea on what are the activities that are usually underestimated?

PM: Yes, normally the bulk is the design part, that it's not underestimated.

MP: It's underestimated…

PM: I mean that it is underestimated not because people think it takes less time, but because normally in commercial terms is sold by less or they didn't accept more, it's getting better. Normally the design part takes longer, because it's a collaborative labor and it involves more people. It's a bit underestimated. The development part, too.

MP: Is it underestimated?

PM: I think so. I think it's normally estimated according to the features on their basic components, I need to get it working and it will take X time and normally it lacks time to refine it. One thing is having the form sending an e-mail; another is having a pretty form with error messages, the pretty e-mails and all those thinks.

Basically it's all the same feature, to have a form, but they only think to have the form I take X time, but then it comes the PO or someone and tells you it's lacking the error messages, this or that. In a general manner I think everyone underestimates the tasks. Then you have others that throw it upwards, as they have no idea and they don't want to get burnt, they say that's a whole week and when the time comes it's done in a day or less, but ok. That's a question of time, experience, having metrics that more or less allow you to understand that. The ideal is for example, working on a project with Drupal and you get that after 4 or 5 projects that on average that set of features takes x long. In the beginning, when the PO is doing the initial estimates or the planning sprint with the team already has the notion that they usually take X long to do this, somewhere in between the eight and the eighty, in the middle there's an average. At this time, we can't make any calculation of that sort.

MP: Is there a critical success factor defined by you or others for each project…? If this isn't met the project isn't…

PM: No, there isn't any structured thing. It's obvious that if you're doing a portal or an application they have to work, if they don't work you have a problem, but you don't have criteria…

MP: It's not commercially defined...

PM: You don't have anyone who says, this needs to work so and so or it's not viable. Who defines the criteria it's us and the client.

MP: Basically it's: this exists, all needs to work.

PM: Exactly, it's a question of commonsense and team adjustment. It's like the form story; you can have a form working correctly with the error messages and all. It works, it's all done, and there is nothing which isn't done. In mere terms this is an acceptance criterion, it works, we can send an e-mail, it registered, it saves on the database, the user got the e-mail, he clicked, confirmed it's all working. But then you can have, if there is time and will, you can have extras this will do it in Ajax or whatever, but it's not defined. It's one of the things that needs to start being defined by the PO, the acceptance criterion of a certain task. If the criterion is that the form needs to be in Ajax and change color, it was defined by him, or the team explains why they can't do it in that manner or if they don't, he has a right to say it isn't done, and get back. Usually, on the client's side, that doesn't exist. The client doesn't define the acceptance criterion, it's not expressed.

MP: So it's more a PO's decision then the client's…

PM: Yes.

MP: The client is just the main scope.

PM: Exactly. The PO needs to have the ability to understand when something is ready to be delivered to the client, and in advance he already knows if the client accepts it or not. Up until now, there weren't exactly things being returned.

MP: OK.

PM: And the ones which did weren't our fault. It's because he changed his mind or he expressed himself incorrectly. There was something in the middle, an external factor which was not related to the project. But the one who makes that filtering should be the PO. As if you have a technical responsible for a team or department it should be someone who looks at the project and states that from the technical point of view it's correct and ready to be shown or put into production or whatever.

MP: So there never was any project… you referred some that were returned. But the main problems are from the client's part or the communication with the client?

PM: There was never something which was developed, at our specific case everything that was returned was by issues with the design not functionality. It's not a question of the portal being poorly made, or the specifications weren't correct but simply because the client changed his mind or didn't even remembered approving it. Up until now, the problem has always been on the client's side.

MP: OK.

PM: There is a lot being returned inside the team.

MP: In tests…

PM: The return here is debatable. You consider something is done; someone looks and says it lacks something or it doesn't comply with the defined criteria or adopted standard but that's normal. Mainly because there isn't a definition of the standards anywhere. It needs to be defined.

MP: There are some, but…

PM: As I refer to standards, it's the acceptance criterion of what is considered done, made. It's easier to write I need a form then I need a form that does that, and if that doesn't happen, this needs to happen so my acceptance criteria are these. It needs to send mails; the other needs to receive and if it doesn't it needs to do whatever… You already know that someone who is developing a task knows exactly what it needs to do, what means being done and what's accepted or not.

MP: And what about the question of the client saying he doesn't remember, is there anything being done to mitigate those problems?

PM: You have no process and no tools to do that officially, as the communication with the clients happens mainly in person or by e-mail, they are the two …

MP: the two channels…

PM: You don't exactly have a collaborative platform where the e-mails that get out or come in are always registered for the team knowledge. There is the commonsense practice that if there is something asked via telephone that alters or may go wrong, there is the sensitivity to register it in writing. According to what was spoken there is an e-mail with it and by principle if no one contests it, which was accepted. Document the most…

MP: the decisions…

PM: the decisions. Via e-mail or you send a design and you get something saying I like it very much, but that doesn't work for you. What works is expressing it is accepted, go ahead or someone saying yes to it. You don't have a place where that is established, saved or documented to everyone. We tried with (active)Colab but…

MP: Because the client didn't…

PM: There you go, if you had something capable, Colab isn't the best example of it. You have a number of applications where the teams collaborate with the

clients and the clients go there without any problem. Maybe the tool wasn't the best for that to happen, it depends of the clients.

MP: Do you think this one has those features?

PM: Up until now I haven't seen any of that. The place where the documentation is, isn't available to interact with the client. I have not seen anywhere I can communicate with the client and it is registered. So, no.

MP: OK. At the end… What about the system itself? You have tested it immensely, now. When it was being developed did you had a say in it?

PM: Yes, I had some inputs. The things were being implemented in a certain way, little by little, getting a lot of input from the people. It changed; you have a methodology which is being applied in a certain way, more forcefully. There is still the input gathering; people say what they need to say, but up until now the methodology is being applied forcefully.

MP: Do you think it's working well?

PM: It's still early to tell. There are some things that cannot be applied exactly as books tell you to, because the teams are different, the people are different and the companies are different. So I don't believe there is one single model applicable to all. Let's see. Up till now those changes weren't incorporated. What the teams are saying is trying to be mitigated inside the constraint of the methodology, but there were no changes.

MP: In terms of methodology, about the system do you think it needs to be altered?

PM: What system?

MP: JIRA, Confluence…

PM: I think no tool is perfect, none can predict every case. You try to adjust according to the tool you have, as you can't change, you work with it and that I think people are doing, while they get to know the platforms. I get that in terms of collaboration and documentation Confluence is good and brings advantages and we are not even getting half the productivity we could. About JIRA, it has some good items but there are others, that either by issues of configuring or lack of knowledge transforms it in a hurdle. On the other hand you have a place where you can see all the projects and a visual representation of what is happening or is going to happen, people can log in the times with the advantages and disadvantages that brings, it's a bit odd you can only log time on things available at JIRA, it doesn't give you reliable metrics. Or you log too much time in issues that really didn't take that long, because

you need to set 8 hours, well at least 6, because 2 you can put elsewhere, when for real things don't work in that way. Where do I put this half an hour here? Do I put it in the two hours I have to speak with my colleagues, but what if I have already spent those, because of other tasks? But these are things that need to be tuned. In a general way, JIRA is better than what we had. The suite is better, now we will see.

MP: Do you think it's still very much at the beginning? At what percentage do you say we are at?

PM: You still didn't have a whole complex project being managed there. If you really realize, we are only using JIRA at about one month, even less officially is the only tool in project management since November 14th, at three weeks. There were already some projects using it before, but the timesheets and times where elsewhere. I guess three weeks or a month of intensive usage isn't enough to draw conclusions.

MP: Do you consider that is helps you with the tasks you regularly have? Are you being more efficient?

PM: I think I'm being more efficient, because I have less "responsibility", one week sprints with four hour meetings, no. The greatest advantage of Scrum or Agile is the collaborative environment among the teams, and so you should need so much time with rituals, because things should happen organically, during the sprint. But with one week sprints, you can't be organic. You started the sprint and suddenly it has ended. We can't even understand much. When we have sprints of two, three weeks it will different, possibly. You have a meeting sprint planning retrospective that takes half a day, or even a whole day, that happens sometimes, then you start a sprint with just four days, you don't have time to enter….

MP: the development.

PM:… the regime. But these are things…

MP: to improve. Ok. It's all.

PM: Done?

MP: Done.