AGILE REQUIREMENTS ENGINEERING IN BUSINESS INTELLIGENCE PROJECT: A CASE STUDY

Osvaldo Araújo Machado, William Chaves de Souza Carvalho
Universidade Federal de Uberlândia
E-mails: osvaldo@si.ufu.br; william.carvalho@ufu.br

Abstract: To develop quality software that meets and satisfies the customers needs is a major concern for professionals working in software development centers. In this context, agile development methods have emerged as alternative to the plan driven methods because they deal efficiently with major problems found in traditional methodologies, such as lack of customer communication, delivery time, and especially the changing requirements which are responsible for the failure of many projects. However, traditional methodologies also have good practices which can and should be used in software development. This article presents, through a case study, the usage of a hybrid methodology for implementing systems called Scrum-RUP. This framework uses the main factors of the Scrum agile methodology, without giving up some practices in the traditional model through the RUP. As a case study, it was considered an application of Business Intelligence which produced satisfactory results. It can be seen that if used in the correct manner, the agile practices can indeed be used in conjunction with traditional methods, resulting in quality projects, on time and budget.

Keywords: Requirements Engineering, Agile Methods, Software Engineering, Business Intelligence.

1. INTRODUCTION

Modern organizations operate in a business environment in which market and technological changes are increasingly rapid which poses additional challenge to traditional approaches of Requirements Engineering (D'AMORIM, 2008). The changes in software projects occurs because the software development is not a static process and this requires the adoption of new measures and methods for the projects became successful.

An increasingly adopted solution to this is the use of Agile Methods to improve software development as its main feature is the easy acceptance of changing requirements, even at the end of development. As agile processes suit changes the customer can get competitive advantages (Beck, 2001).

The software development projects has common elements that are noticeable in the developed applications, which can help increase productivity. Teams of success, both agile as traditional, enjoy the following “recipe for success”: it is necessary to provide more time for staff to learn. The teams have to learn about technology, project domain, client characteristics and project risks. A team that succeeds learn quickly because learning is the key to successful software engineering (ELSSAMADISY, 2008).

The software development according agile methods have gained the sympathy of a growing share of development professionals as they seek solutions to the problems encountered in the traditional model. Agile methodologies are configured in the solution to many of these problems in software development, especially those with regard to the frequent changes in requirements, lack of communication with the client and a lot of bureaucracy.

(GROUP, 2015) conducted a study that included 50,000 projects around the world, of all kinds, sizes and complexity in order to analyze the current scenario of the current software development industry. Figure 01 shows the results of this research.

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

Figure 01 – Agile versus Traditional Methods (GROUP, 2015)

The results show that small projects have more chance of success, and make it clear that regardless of the size, the use of agile methodologies in projects, give better results compared to traditional practices (GROUP, 2015).

A recurring question among project managers is when it is most advantageous to use certain methodology. This work mixes both practices, resulting in a hybrid model. Current research will also reveal how Business Intelligence is being used more every day in organizations to support decision-making, based on strategic business data (DE SILVA DOS S .; SOUZA, 2013), (COSTA, 2011), (MILANI, 2014).

2. Initial Concepts

This section aims to presents the fundamentals and concepts related to the areas of knowledge involved in the research and review of the literature on the research topic of this work.

2.1. Metodologia Ágil

The software improvement development process is not new. Since the mid-90s, began to emerge so many alternative processes forward option to traditional, considered slow, bureaucratic, rigid and inadequate to the nature of the activity. Initially, these processes began to emerge they were called "light" since the former were considered "heavy" (MILANI, 2014).

After the Agile Manifesto (release Beck, 2001) these new methodologies, initially named "light", became agile calls. This manifesto was created by a group of 17 specialists who aimed to discuss ways to develop software in a lighter, quickly and people-centered. They emerged thus the term "Agile Software Development" and "Agile Methods" (MILANI, 2014).

Agile methods can be characterized as incremental planning strategy in which various parts of the system are developed separately and integrated when complete. These methods are also based in rapid development cycles; cooperation and interaction between staff (ABRAHAMSSON et al., 2003), (CARVALHO, 2012).

The final goal of agile methods is to enable an organization to be responsive. Thus, while the agile techniques may vary in practices and emphasis, they share common features, including iterative development, focus on interaction and communication, increased customer involvement and especially identifying and responding to changes quickly. This last point is its great advantage rises traditional approaches because iterations allow the team to adapt quickly to changing requirements (COHEN; Lindvall; COSTA, 2004).

2.2. Scrum

Ken Schwaber and Jeff Sutherland created the Scrum method in 1996 (Carvalho, 2012), and according to its own creators, it can be defined as "a framework where people have the freedom to address and resolve complex and adaptive problems, while productively and creatively deliver products with the highest possible value "(Schwaber K .; SUTHERLAND, 2013)

This framework does not consider that be feasible to plan software development projects fully at the beginning, assuming that they are highly complex and unpredictable (Schwaber, 2004). Principles as small teams of up to seven people, with requirements that are unstable or unknown and with short cycles, in which the development is divided into time-boxes of up to 30 days, also called sprints, are the foundation of Scrum (CARVALHO, 2012).

According (Milani, 2014), Scrum adapts and responds to changes as well, because its incremental and interactive feature for development. Therefore, maximizing the delivery of software effectively. Three main roles take on team activities (CARVALHO, 2012), (Schwaber, 2004):

- **Product Owner**: The Product Owner represents the interests of everyone in the project, especially the client.
- **ScrumMaster**: The ScrumMaster ensures that everyone follows the rules and practices of Scrum and is responsible for removing the impediments project.
- **Team**: The team is responsible for developing the product features, define implementation strategies and manage their own work in a sprint.

According to (MILANI, 2014), Scrum also has some artifacts that enable design progress of vision. They are: the Product Backlog, Sprint Backlog and Increased Product. They are used throughout a project, but have different setting times.

- **Product Backlog**: It consists of a list of prioritized items that include functional and nonfunctional requirements of the system / product being developed in the project.
- **Sprint Backlog**: Corresponds to the list of tasks that define the project team to implement in the sprint, the selected requirements of the Product Backlog.

- **Increase Product**: At the end of each sprint, the Development Team delivers an increment of the product, the result of which was produced during the sprint. It allows the product owner to realize the value of the investment and also envision other possibilities.

2.3. Sistemas BI

According (Watson, Wixom, 2007) and (Batista, 2014), the Business Intelligence concept is comprehensive, but can be considered as "processes that support analytical processing applications." According more specific definitions, (BATISTA, 2014) defines BI as:

> "...integrated systems that combine data acquisition and storage, with knowledge management and analytical tools to provide inside information, highly complex and represent a competitive advantage to those responsible for planning and decision making process in organizations."

However, according to (GILAD, 1989), Business Intelligence are the process of collecting and interpreting relevant information about a foreign environment, generating data that can assist in strategic decision making process and generate or support long-term competitive advantage in organizations. BI provides knowledge to the company in relation to its environment.

Business Intelligence projects have to gain data from various existing sources in the organization for further processing to fit the previously modeled DW, and finally, load the data that has been transformed to the DW structures. This process is called ETL and represents a very significant part of the overall project effort. It is estimated that 80% of the total effort is related to extraction processes, processing and loading the data. (BATISTA, 2014). According to (BOUMAN; DONGEN, 2009), ETL processes can be subdivided into three main tasks:

1. The data extraction can occur from any source that contain useful information for making business decisions;
2. The transformation suits the extracted data to be entered into the data warehouse destination;
3. The loading is the storage of data that has been the target of transformation in the respective tables in the data warehouse of destination.

The classical architecture ETL is illustrated by Figure 2.

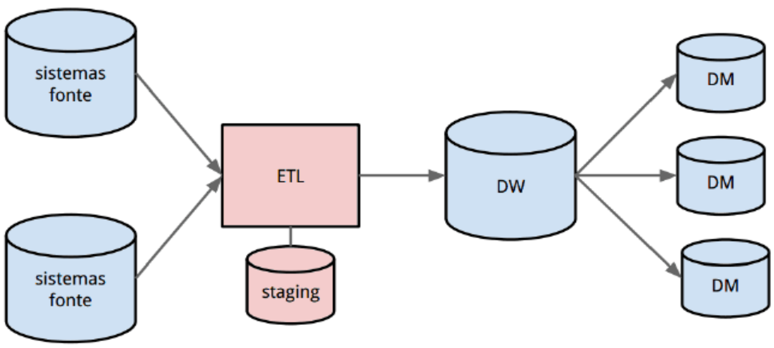

Figure 02 – ETL Classical Architecture (BATISTA, 2014)

As the ETL process has great importance in business intelligence projects, there are a wide range of specific tools in the market for this activity (BATISTA, 2014). In that case study it was used the BI SQL Server Integration Services tool, a powerful tool currently used in very large companies.

2.4. SQL Server Integration Services

SQL Server Integration Services, also known as SSIS, is a BI tool from Microsoft. It's a data integration platform for enterprise-class and data conversion solutions. SSIS can be used to solve complex business problems by updating data warehouses, cleaning, data mining and managing SQL Server objects. Integration Services can extract and transform data for a wide variety of sources such as XML data files, flat files, and relational data sources, and transfer data to one or more destinations (MICROSOFT, 2014).

Using SSIS, data can be extracted from various sources such as text files, Excel spreadsheets, or databases, such as SQL Server, for example. They are processed by changing types, shapes, sizes, among others, and extracted in a data source, which may also be a text file, a spreadsheet or a database. The source of origin and destination need not be equal, one can, for example, extract data from a spreadsheet, treat them and save the results in a table in a database. The set of objects used by ETL tools SSIS to extract, transform and load information is known as DTS.

Figure 03 illustrates an ETL process being carried out in SSIS, where data were extracted from a SQL Server database table, they underwent a transformation, were sorted and loaded to another Teradada table bank.

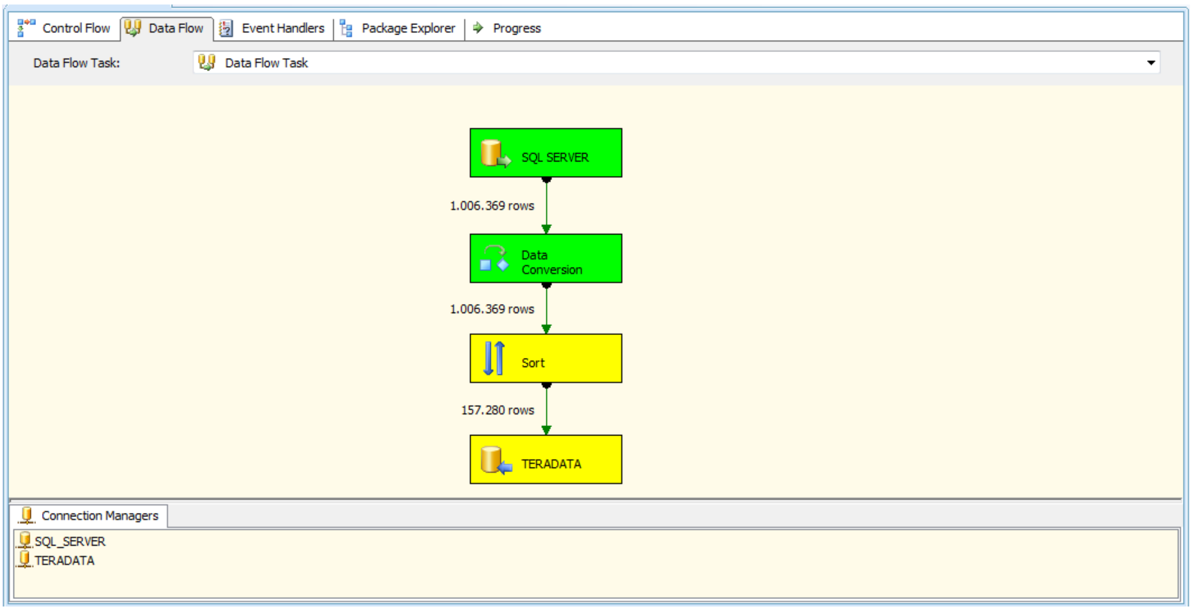

Figure 03 – SSIS ETL Process

3. Hybrid Method for Business Intelligence Requirements, Design and Implementation

This section presents the hybrid framework Scrum-RUP used in the case study. Scrum-RUP is a framework for integrating the agile method Scrum practices with traditional methodology RUP in a Business Intelligence oriented development process.

3.1. Model lifecycle

The model lifecycle this framework incorporates practice of evolutionary life cycle and time-boxed model, as shown in Figure 04, for the following reasons:

- **Evolutionary delivery**: unlike the purely agile development life cycle, the architecture design stage is anticipated for the early stages of the project, as in the RUP (KRUCHTEN, 2003). Thus, it can be understood that the proposed lifecycle partially incorporates the idea of evolutionary delivery model;
- **Time-boxed**: the time-boxed model also partially fall within the life cycle of this framework, for its development, as Scrum, is divided into fixed time iterations (usually two weeks) incremental delivery possibility the product to the end each. However, the application of time-boxing here is not fully adherent to agile proposal, there is a difference that characterizes the hybrid nature of this framework.

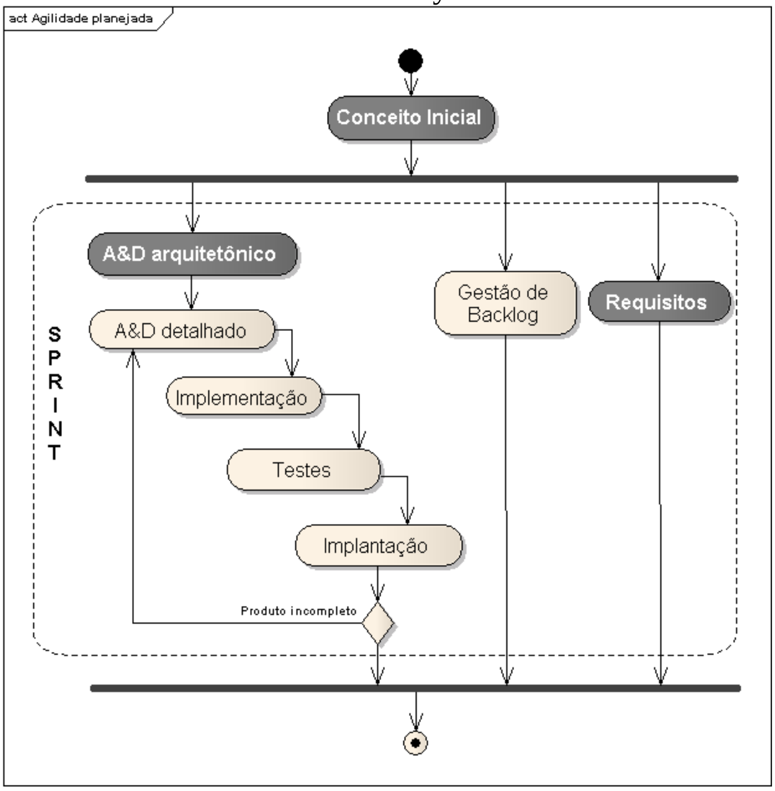

Figure 04 – Model lifecycle

Follow a brief description of the purpose of each sub-process indicated in the life cycle model shown in Figure 04.

- **Initial Concept**: The objective of this sub-process is to identify the initial requirements of the system and the most relevant architectural aspects. Based on this information, it uses some effort estimation technique to present a business proposal to the client.

- **Architectural Analysis & Design**: Its goals is the development of the system architecture and the mitigation of major project risks.
- **Requirements**: Comprises use case specification activities (COCKBURN, 2000) and its validation by the client. The requirements are not hard to change as in plan driven methods.
- **Detailed Analysis & Design**: Production of all types of non-architectural design artifacts such as use cases and user interface prototypes achievements.
- Implementation: Intended to carry out activities coding, unit testing, integration and packaging.
- **Tests**: Corresponds to planning activities and execution of tests.
- **Implantation**: Implementation activities of approval and production environments as well as production documentation.
- **Backlog Management**: Corresponds to the job of keeping the Product Backlog to date, generating and prioritizing stories (user stories) (COHN, 2004)

3.2. Characteristics

The hybrid framework illustrated in Figure 4 has the following characteristics relating to the integration of agile and traditional paradigms:

- **Time-boxing hybrid**: the time-boxing has hybrid character, for although fixed time cycles are established for potential delivery increases in product, scoping is done previously in Home Concept stage, which means it does not have fixed assets and time with estimated scope;
- **Sprints**: All "macro step" dotted lifecycle (see Figure 04) should be performed in the form of sprints, the short iterations defined by Scrum;
- **Architecture**: The life cycle shown in Figure 04 indicates that the architectural Analysis & Design of sub-process should be completed in some time of the project and, from that, subsequent activity analysis and design occur at a detailed level, since the system architecture is already stabilized. The decision to give strict focus on the development of architecture has, in part, in order to mitigate the risk of incurring a poor design because of the lack of focus on architecture (MCBREEN, 2002), (STEPHENS; ROSENBERG, 2003) (PETERSEN; WOHLIN, 2009);
- **Requirements**: Unlike what determines a fully traditional methodology, specification and validation activities may occur throughout the "macro step" dotted lifecycle as can be seen in Figure 04, but such efforts should focus on early iterations of the project, especially to prioritize the requirements that impact the development of architecture. Functional requirements are specified as use cases, following the principle of the outermost regions;
- **Tests**: The specification of test cases is needed to ensure that the use cases are being adequately met, and to ensure acceptable levels of quality and formality in the product validation. It shall not exclude the possibility of carrying out agile principles, such as continuous and rapid feedback tests, once the model development process life cycle is organized in short fixed period of iterations (time-boxes), and furthermore several test activities can be carried out more dynamically in conjunction with implementation activities.

- **Backlog Management**: The sub-process called Backlog management should occur throughout the "macro step" dotted indicated in Figure 04. To keep the current backlog, this sub-process must use information generated by sub Requirements, Analysis & Design and Architectural tests (the latter identifying defects that in turn generate change orders entered in the Product Backlog). In this framework, the subprocess backolog management can be seen as the communication interface between the portion agile and traditional part of the process;
- **Agile and traditional partitioning**: The sub highlighted in Figure 04, Analysis & Initial Concept and Architectural design should follow a consistent rigor to the traditional paradigm, as principles of process capability, verification and validation. The sub Requirements and Test are situated in a gray area of agile or traditional predominance as discussion presented above. The remaining sub area have agile predominance, as shown in Figure 05. This choice was taken from the idea that the primary objectives of the traditional paradigm (predictability, stability and high assurance) are best served if the architecture is developed in the early stages of the process and if a project from initial concept is formed and formally validated .

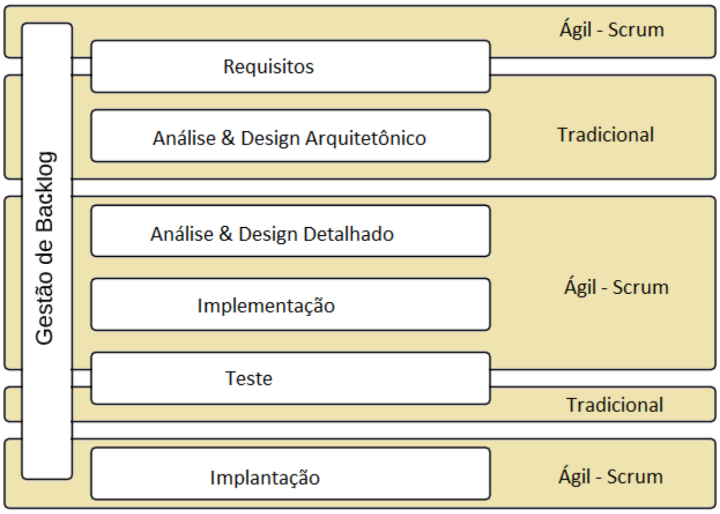

Figure 05 – Subprocess aggregated by Agil and Traditional paradigms

4. CASE STUDY

This section presents the BI project used as a case study developed using a hybrid approach presented in this work, using the SSIS Tool.

4.1. The Project

This project involves the development of a Management Information System for a telecommunications company area by creating, manipulating data and loading informations in a the relational database Teradata table. This system aims to unify various existing databases into a single table with information that will be of paramount importance for making management decisions of the requesting company.

The project was divided into only two stages or sprints. The first phase was related to the mobile part of the project, while the second was dedicated to the fixed part. Well-established business rules have been defined for both steps.

4.2. Solution Details

The first step was the creation of the table in the Teradata database, containing all the necessary fields to store the information needed to feed the solution of quality indicators. Figure 06 shows an overview of DTS, with all the system components involved. Soon after, each of the DTS will be explored and explained step by step.

From the overview of the DTS it is possible to clearly observe the separation of the two project phases (Mobile and Landline), each part is contained in a container and can be performed independently of the rest of the project.

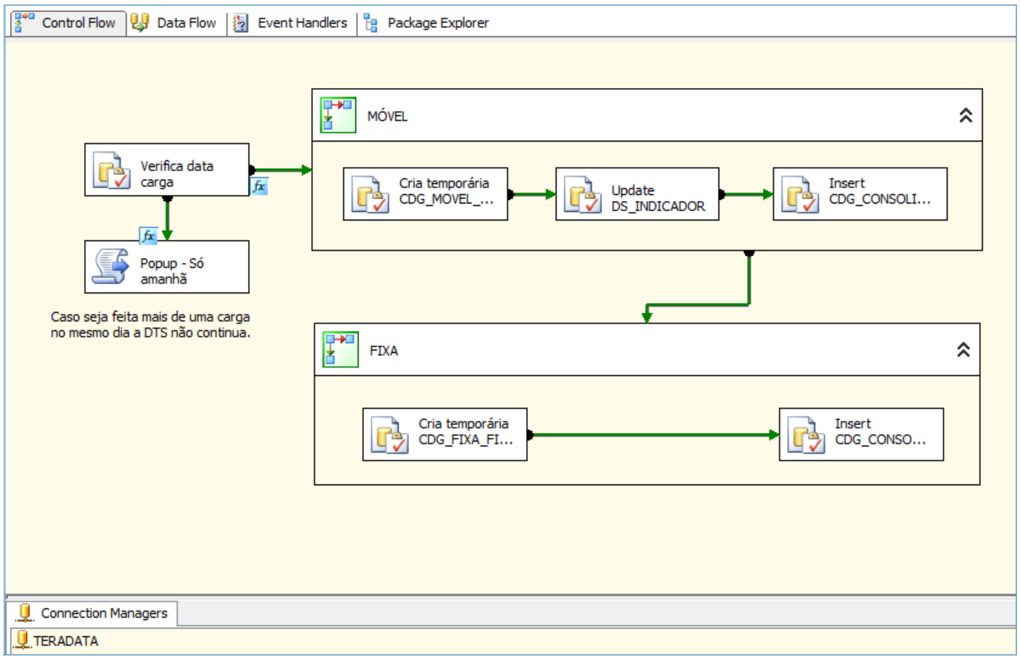

Figure 06 – DTS overview

According to the business rules specified in the project document, the process must be performed no more than once a day. To this end, the boxes were created "Checks load data" and "Popup - Only tomorrow", as shown in Figure 07.

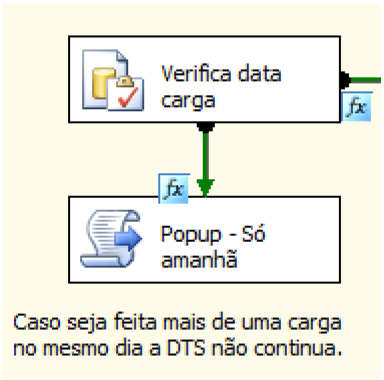

Figure 07 – Date of Last Load check

The box "Checks date load" checks the date of the charges in the destination table. If it is equal to the present day, the flow goes to the low box "Popup until tomorrow." The "Popup until tomorrow" has a script that returns an error message to the user that the DTS has been executed that day and therefore can only be performed again the next day.

4.3. Mobile Phase

The design of the Mobile phase correspond to the first customer delivery. If the testing box “date of cargo” verify that there has not been an execution of DTS that day, the process moves to its first container, the "Mobile", which includes part of mobile technology project. This container comprises three boxes that do all the ETL process of the project's mobile data. Figure 08 shows in greater detail.

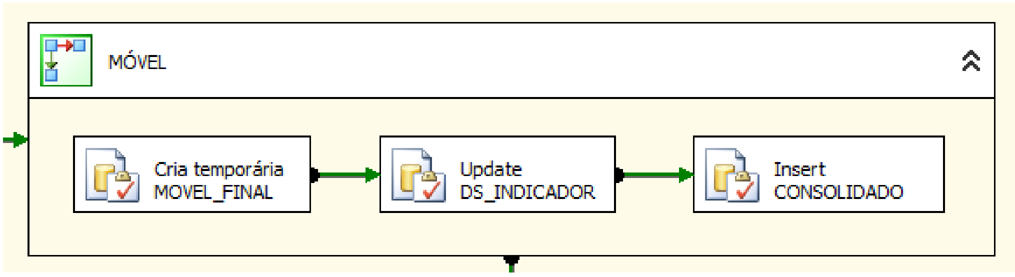

Figure 08 –Mobile Phase View

Creates MOVEL_FINAL: creates the temporary table MOVEL_FINAL into the Teradata database and populates it with data from FISICOS_MOVEL table, also from Teradata, according to the business rules specified in the functional specification document.

Update DS_INDICADOR: It has the function to enter data in the field of MOVEL_FINAL DS_INDICADOR temporary table, following the business rules specified in the Functional Specification document.

Insert CONSOLIDADO: Makes data loading to the final table CONSOLIDADO in the Teradata database. The data entered are from the temporary table "MOVEL_FINAL", created the first container housing.

4.4. Landline Phase

After delivering and obtaining the approval of the first sprint of the corresponding project to the movable part, it was carried out the development of its landline phase.

According to the business rule which states that only one load can be made per day, the second container checks if there is an execution of DTS that day. If not, the process follows the flow enters the container and "Fixed", which includes a fixed part of the project technology. This container includes two boxes that do all the ETL process of fixed technology project data. Figure 09 details the solution.

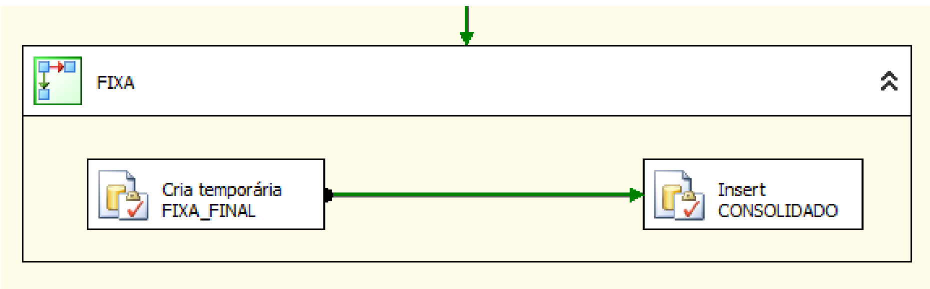

Figure 09 – Landline Phase View

Creates FIXA_FINAL: creates a temporary table FIXA_FINAL within the Teradata database and populates it with data from the table "FISICOS_FIXA" according to the business rules specified in the functional specification document. ·

Insert CONSOLIDADO: This box serves to enter data in the final table "Consolidated" in the Teradata database. But here, the entered data are from the temporary table "FIXA_FINAL" created in the first box that container.

4.5. Connections

The Microsoft SQL Server Integration Services packages use connections to perform different tasks and deploy Integration Services features. In this project, we use a connection to access the relational database Teradata and executing SQL statements (MICROSOFT, 2014th).

For this project, only one connection was used as the target data were stored on the same DW that source. An SSIS project can have as many connections as necessary.

As shown inFigure 10, setting up a connection is made via the Connection Manager. To set, you must enter the server IP to be used, username and password.

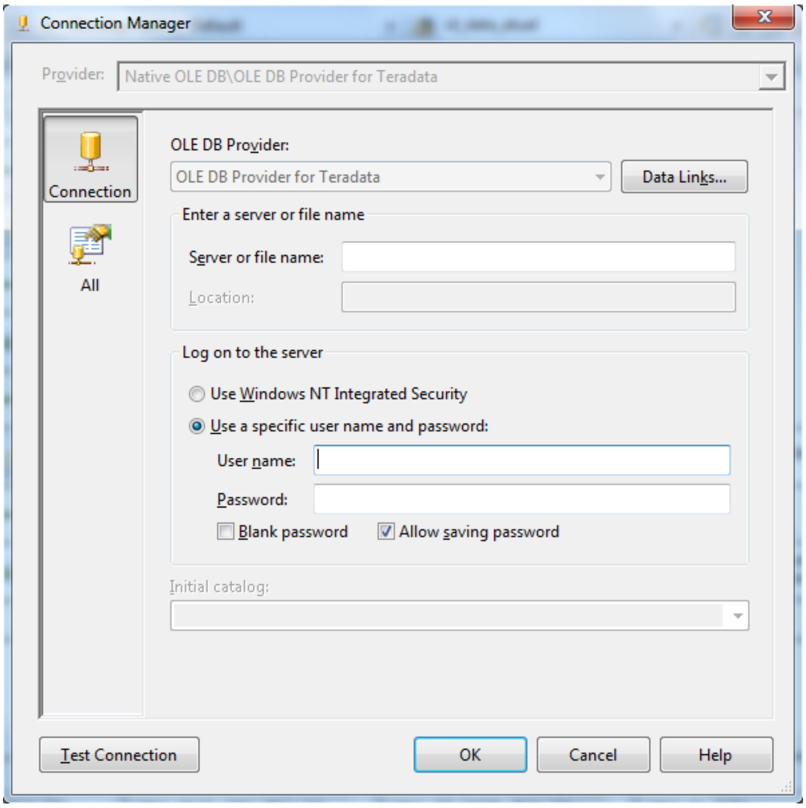

Figure 10 – Connection Manager

4.6. Results

The final deliverable of the project was a physical table created and populated in the relational database Teradata called "CONSOLIDADO". This table gathers data from both mobile and landline platforms of the contracting company and will be useful in making managerial decisions, enabling data search services since the information is now gathered in one place.

The overall balance of the project was very positive, since the proposal Scrum- RUP achieve the expected results. The project was developed on time and costs provided with all the features implemented and tested.

The most relevant results of the research indicate that:

- The perception that an integration of agile models and traditional methods is quite possible and productive, since this integration is carried out properly.
- The biggest cause of failures in software projects is the high vulnerability of requirements, suffering constant changes during the project. The flexibility with which requirements are covered by the agile methodologies emerge as a great solution to this problem.
- The biggest difference found between Agile and traditional is the way they react to changing requirements.

5. CONCLUSION

This paper presents an application of a Business Intelligence application development hybrid method. That method aims to address solutions for the recurring problems encountered in the use of absolutely traditional methodologies such high requirements volatility and related constant change requests incomes. The applied method uses both key concepts found in agile Scrum and traditional RUP.

Using a pure agile methodology to treat the traditional software development problems would solve some problems, especially those related to the great vulnerability of the project scope changes, but it would be wrong to disregard completely all traditional practices because agile do not stimulates great level of formalisation. It was in this context that the idea of using a hybrid method, which was put into practice in the present case study, I try very satisfactory results.

To integrate de best practices of two paradigms and to add concepts as diverse was not a trivial task. It took a lot of study and effort to know in depth the fundamentals and concepts of both the Scrum agile methodology and RUP that was chosen for this work. Once extracted the most adequate from each one, we used them together. To create a usefull scenario of application has been the biggest challenge of this work.

Since one of the most glaring differences between agile and and traditional methods is how the analysis and response to changing requirements is treated, a case study was presented, in which the hybrid methodology Scrum-RUP used in this work was put into practice by through a BI project. Even the Business Intelligence along with other technologies and tools relevant to the study, were also discussed and detailed.

Master the tools and technologies needed for the actual case study development developed during this work was overcome another challenge. The project included the use of an analytical database that has the ability to process large volumes of data and is more focused on organizations as well as an ETL tool, all governed by the rules of the hybrid framework.

We conclude that agile methods have almost always successful development projects, but it is not correct completely rule out the techniques used in traditional methodologies, for they, when taken properly, can bring many benefits for the project as predictability and stability. The best thing to do is to leverage the strengths of both, discard the weaknesses and thus create a hybrid methodology that may suffer small changes according to the needs of each project, while respecting the overall structure.

REFERENCES

ABRAHAMSSON, P. et al. **New directions on agile methods: A comparative analysis.** In: 25th International Conference on Software Engineering. [S.l.: s.n.], p. 244–254, 2003.

BATISTA, L. P. L. **Implementação de uma solução open source de business intelligence com metodologia Ágil.** Lisboa, Portugal, 2014.

BECK, K. et al. **Manifesto para desenvolvimento ágil de software**. 2001. Disponível em: <http://www.manifestoagil.com.br>.

BOUMAN, R.; DONGEN, J. van. **Pentaho Solutions Business Intelligence and Data Warehousing with Pentaho and MySQL.** [S.l.]: Wiley Publishing, Inc., 2009. ISBN 978-0-470-48432-6.

CARVALHO, W. C. de S. **Análise dos efeitos do turnover na produtividade de processos de software tradicionais e híbridos.** 2012.

COCKBURN, A. **Writing Effective Use Cases.** 1st. ed. Boston, MA, USA: Addison- Wesley Longman Publishing Co., Inc., 2000. ISBN 0201702258.

COHEN, D.; LINDVALL, M.; COSTA, P. **An introduction to agile methods.** Advances in Computers, Elsevier, Amsterdam, v. 62, p. 2–67, 2004.

COHN, M. **User Stories Applied: For Agile Software Development.** Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2004. ISBN 0321205685.

COSTA, G. H. d. C. **Engenharia de requisitos no desenvolvimento de software Ágil.** 2011.

D'AMORIM, F. R. d. S. **Engenharia de requisitos, metodologias Ágeis.** 2008.

ELSSAMADISY, A. **Agile Adoption Patterns: A Roadmap to Organizational Success.** 1. ed. [S.l.]: Addison-Wesley Professional, 2008. ISBN 9780321514523.

GILAD, B. **The role of organized competitive intelligence in corporate strategy.** Columbia Journal of World Business, v. 24, p. 29-35, n. 4, 1989.

GROUP, S. **CHAOS MANIFESTO 2015**, 2015.

KRUCHTEN, P. **The Rational Unified Process: An Introduction.** 3. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003. ISBN 0321197704.

MCBREEN, P. **Questioning Extreme Programming.** Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002. ISBN 0201844575.

MICROSOFT. **Conexões do SSIS (Integration Services).** 2014. Disponível em: <https://msdn.microsoft.com/pt-br/library/ms140203>.

MICROSOFT. **SQL Server Integration Services.** 2014. Disponível em: <https: //msdn.microsoft.com/pt-br/library/ms141026>.

MILANI, R. P. R. W. F. **Métodos Ágeis Para Desenvolvimento de Software.** [S.l]: Bookman, 2014. ISBN 9788582602072.

PETERSEN, K.; WOHLIN, C. **A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case.** J. Syst. Softw., Elsevier Science Inc., New York, NY, USA, v. 82, n. 9, p. 1479–1490, set. 2009. ISSN 0164-1212.

SCHWABER, K. **Agile Project Management with Scrum.** 1st. ed. [S.l.]: Microsoft Press, 2004. ISBN 9780735619937.

SCHWABER K.; SUTHERLAND, J. **Guia do Scrum.** 2rd. ed. [S.l.: s.n.], 2013.

SILVA D. E. DOS S.; SOUZA, I. T. C. T. **Metodologias Ágeis para o desenvolvimento de software: Aplicação e o uso da metodologia scrum em contraste ao modelo tradicional de gerenciamento de projetos.** Computação Aplicada, p. 39–46, 2013.

STEPHENS, M.; ROSENBERG, D. **Extreme Programming Refactored: The Case Against XP.** [S.l.]: APress L. P., 2003. ISBN 1590590961.

WATSON, H. J.; WIXOM, B. H. **The current state of business intelligence.** Computer, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 40, n. 9, p. 96–99, set. 2007. ISSN 0018-9162.