UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

COPLAM - CONTINUOUS PLANNING ADOPTION METHOD

Rafaela da Fonseca Sampaio

Advisor
Gleison dos Santos Souza

Co-advisor
Cristina Cerdeiral

RIO DE JANEIRO, RJ - BRASIL
SEPTEMBER 2017

COPLAM - CONTINUOUS PLANNING ADOPTION METHOD

Rafaela da Fonseca Sampaio

DISSERTATION PRESENTED AS PARTIAL REQUIREMENT FOR OBTAINING COMPUTER SCIENCE MASTER DEGREE TITLE FROM UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO (UNIRIO). APPROVED BY COMMISSION SIGNED BELLOW.

Approved by:

_____

Gleison dos Santos Souza, D.Sc (Advisor) – UNIRIO

_____

Cristina Cerdeiral, D.Sc – UNIRIO

_____

Leonardo Azevedo, D.Sc. – UNIRIO

_____

Marcos Kalinowski, D.Sc. – PUC-RIO

RIO DE JANEIRO, RJ - BRASIL
SEPTEMBER 2017

Sampaio, Rafaela da Fonseca
S192 COPLAM - Continuous Planning Adoption Method / Rafaela da Fonseca Sampaio. -- Rio de Janeiro, 2017.
118 f.

Orientador: Gleison dos Santos Souza.
Coorientadora: Cristina Cerdeiral.
Dissertação (Mestrado) - Universidade Federal do Estado do Rio de Janeiro, Programa de Pós-Graduação em Informática, 2017.

1. Planejamento Contínuo. 2. Desenvolvimento de software ágil. 3. Planejamento de projetos. 4. Planejamento no nível de time. I. Souza, Gleison dos Santos, orient. II. Cerdeiral, Cristina, coorient. III. Título.

ACKNOWLEDGEMENTS

First, I thank my parents. My mom Maria Helena and my dad Cosme have always been the greatest support and inspiration I have ever had. They sacrificed so much in their lives for me, and everything I accomplished is because of them. Their love and dedication gave me strength along the way and their incentive for studying and perusing knowledge is what motivated me into applying for a master's degree in the first place.

To my advisor, Gleison, words can barely describe what you mean to me and this work. Thank you so much for the guidance, the patience, the support and the friendship. We can get to know a few extraordinary professors in life, he is one of these few that every student is lucky to meet, and the most important to my career. I am deeply proud of having him as an advisor since college and hope will continue to do research together.

Pedro, for the love and friendship, it was essential to keep me going. His support and the faith he has on me helped through good and bad times. Pedro is also a source of inspiration and an example of researcher. I am grateful for having him in my life, loving someone who understands and shares your dreams with you is a precious thing.

Special thanks to my manager Gisele, the person who understood all the difficulties of perusing this degree in conciliation with our work and dedicated time of an impossible schedule to help me with it.

Cristina, for co-advising this work and dedicating long nights to review and help on COPLAM's construction.

To everyone at work that participated in the studies and contributed to the method.

I also thank PPGI teachers for all the knowledge given in this learning journey.

FONSECA SAMPAIO, Rafaela. **COPLAM – Continuous Planning Adoption Method.** UNIRIO, 2017. 118 páginas. Dissertação de Mestrado. Departamento de Informática Aplicada, UNIRIO.

RESUMO

Contexto: Muitas organizações enfrentam ambientes de negócio em constante mudança. O desenvolvimento de software em contextos dinâmicos está sujeito à necessidade de se adaptar de acordo com eventos internos e externos à organização, que podem afetar o seu planejamento. Eventos podem ser mudanças nas necessidades dos clientes, novas tecnologias disponíveis, legislações, lançamentos efetuados pela concorrência, etc. Os métodos ágeis tornaram-se populares por serem focados na flexibilidade e adaptação constante, porém, no planejamento de projetos ainda há uma falta de abordagens que permitam a evolução contínua dos planos. O planejamento contínuo refere-se ao processo de planejamento em ciclos rápidos e paralelos, de forma que os planos evoluam de acordo com eventos. Ainda não está amplamente estabelecido na literatura, faltam abordagens que orientem sua adoção, principalmente em níveis de planejamento diferentes de *release*. *Objetivo*: Desenvolver um método para a adoção de planejamento contínuo no nível de time do desenvolvimento de software ágil. *Método*: O COPLAM foi proposto com a finalidade de auxiliar as organizações a adotar planejamento contínuo no nível de planejamento de times, que abrange release, iteração e dia, de acordo com o seu contexto e projetos. O método foi avaliado em um estudo de caso com dois times de desenvolvimento em uma empresa de pequeno porte. *Resultados*: O Planejador foi capaz de definir ciclos de planejamento dos projetos e evoluir os planos de acordo com os eventos ocorridos. A percepção de utilidade, facilidade de uso e uso futuro foi avaliada por meio do Modelo de Aceitação de Tecnologia (TAM) e melhorias foram realizadas para futuras aplicações do método. *Conclusão*: A partir destes resultados obtivemos indícios de que o método é capaz de apoiar a adoção de planejamento contínuo no nível de times e foi aplicado com sucesso na indústria. O COPLAM foi capaz de apoiar a definição de planos em ciclos rápidos e paralelos e o gerenciamento de eventos durante a execução dos planos.

Palavras-chave: planejamento contínuo, desenvolvimento de software ágil, planejamento de projetos, planejamento no nível de time.

FONSECA SAMPAIO, Rafaela. **COPLAM – Continuous Planning Adoption Method.** UNIRIO, 2017. 118 pages. Master Degree Dissertation. Computer Science Department, UNIRIO.

ABSTRACT

Background: Many organizations face business environments in constant change nowadays. Software development in dynamic contexts like these are subjected to the necessity of being adapted according to internal and external events to the organization and that can affect planning. Events can be changes in clients' needs, new technology available, legislations, competitors' releases, etc. Agile methods became popular by focusing on flexibility and constant adaptation, however, project planning still lacks approaches to support the continuous evolution of plans. Continuous planning refers to the planning process in rapid and parallel cycles in a way that plans evolve according to the events. It is relatively new and not a well-stablished practice in the literature, there is a lack of approaches to guide continuous planning adoption, especially in planning levels besides release planning. *Goal*: Our goal is to develop a method for continuous planning adoption at the team level of agile software development. *Method*: COPLAM was proposed to help organizations on continuous planning adoption in the team level, which covers release, iteration and day, according to organization context and projects. The method was evaluated in a case study with two development teams on a small-sized organization. *Results*: Using COPLAM allowed the Planner to define planning cycles for projects and evolve plans according to events. The perceptions of usefulness, ease of use and self-predicted future usage were evaluated by the Technology Acceptance Model (TAM) and improvements identified in the results were executed for future applications of the method. *Conclusion*: From these results, we have evidences that the method can support the adoption of continuous planning in the team level and it was successfully applied in industry. COPLAM was able to support the definition of plans in rapid and parallel cycles and the management of events along plans' execution.

Keywords: continuous planning, agile software development, project planning, team level planning.

SUMMARY

FIGURE INDEX

TABLE INDEX

1. INTRODUCTION

1.1. Motivation

Agile Methods are highly popular nowadays (MISHRA and MISHRA, 2011). The current business environment of Information Technology (IT) organizations is constantly changing and very unstable, which leads to an increase in agile and lean development practices adoption (SUOMALAINEN et al., 2015b). Agile software development practices focus on flexibility, efficiency and velocity and, by doing that, led to a paradigm change about how software development is executed. Although these practices succeeded, they are not the final step of software development (OLSSON et al., 2013).

Organizations have changed the way they execute product, service and business development due to the adoption of agile and lean practices. Also, many business environments present more streamlined process structures and continuous competency development. Therefore, their strategy changes according to user and market needs, executions and identification of new opportunities. Innovative approaches to support continuous practices throughout the organization are needed. Continuous planning is one of them, although it is relatively new and not yet a well stablished field of research (SUOMALAINEN et al., 2015b).

Continuous planning refers to the planning process in a universe subject to continuous change, where the planning problem is frequently connected to adapting to the universe in which new information is sensed (BRENNER and NEBEL, 2009). On continuous planning, plans are artifacts that evolve in response to changes in the business environment and therefore, implicate on an approximation of planning and execution (LEHTOLA et al., 2009). In such context, providing re-planning quickly allows the interlacing of planning and execution (KNIGHT et al., 2001).

From the software development point of view, continuous planning refers to the organizational capacity of conducting planning in short and parallel cycles that can refer to hours, days, weeks or months, depending on the planning level (FITZGERALD and STOL, 2014). Despite of the fact that planning can be done at regular intervals (SUOMALAINEN et al., 2015b), the future horizon is not fixed (BOGSNES, 2008)

(SUOMALAINEN et al., 2015b), and the plans should also be adjusted according to internal and external events (SHALLOWAY et al., 2009). Software development organizations using agile methods mostly refers to continuous planning in a release planning context (FITZGERALD and STOL, 2014) (SUOMALAINEN et al., 2015b), but it is important to notice that continuous planning can be applied at several levels, such as organizational, product, portfolio, strategy, release, iteration and day (COHN, 2006).

As agile methods aim at allowing software development to follow frequent changes in business environments, the very nature of business requires planning activities to be executed more frequently to assure alignment between business needs and software development (LEHTOLA et al., 2009). Even with several existent agile methods, organizations have increasingly searched for customized in-house methods to match its own specific needs (AYED et al., 2012).

According to LEHTOLA et al. (2009), in continuous planning, plans are artifacts that evolve in response to changes related to business environment, therefore implicating in more integration between planning and execution. Even if it is expected from organizations to have continuous planning practices, only a few of them actually uses them (RICKARDS and RITSERT, 2012) and continuous planning is a new and not fully studied field of research, especially from the perspective of organizations using agile and lean development (SUOMALAINEN et al., 2015b).

New and innovative approaches to support continuous practices inside organizations are needed, continuous planning being one of them (SUOMALAINEN et al., 2015b). There is no standard method or approach for continuous planning (DE FRANÇA et al., 2017).

SUOMALAINEN et al. (2015b) define as elements of continuous planning organizational planning, strategic planning and business planning, and describe each one as follows:

- Organizational planning defines the organizational level and timeframe of a plan;
- Strategic planning is related to the overall plan of the organization;
- Business planning forms the budgeting frame of a plan.

This work is limited to organizational planning at the team level, which englobes release, iteration and day planning levels.

1.2. Goal and Research Question

The goal of this work is to develop a method for continuous planning adoption at the team level of agile software development. The method is expected to support organizations in adopting continuous planning at the team level on their software development process, assisting the projects to adapt according to internal and external changes considering business needs. To achieve this goal, we defined five requirements for the method, which are consistent with the Design Science Research (DSR) (HEVNER et al., 2004) methodology explained in the next section. The requirements are:

R1 – The method shall allow plans to evolve at any time given an internal or external event

R2 – The method shall guide the definition of planning in short and parallel cycles according to the organization needs

R5 – The method shall support continuous planning in the levels of release, iteration and day.

R4 – The method shall foster the alignment between planning and execution of projects

R5 – The method shall support the identification of strengths and weaknesses of the planning process and its improvement

This work aims at answering the research question “How to support continuous planning adoption at the team level of agile software development?”.

1.3. Methodology

Design Science Research (DSR) is the methodological approach used in this dissertation. DSR is a constructing research paradigm originated in Information Systems research that has the goal of developing new and innovative artifacts (GREGOR and JONES, 2007). According to HEVNER et al. (2004), *design science* is an approach to find solutions for problems in which the understanding of the problem and the solution are reached during the creation and application of a projected artifact.

DSR is motivated by the desire to improve the environment by the introduction of new and innovative artifacts (SIMON, 1996) and it is different from industrial design, i.e. creation of a new object, because it focus on building knowledge during the rigorous

process of creating and applying the new artifact. Thus, the creation of new artifacts is based on existing theories and generates new knowledge regarding the artifact (HEVNER et al., 2004).

DSR is, according to HEVNER (2007), an embodiment of three closely related cycles of activities: the Relevance Cycle, the Design Cycle and the Rigor Cycle. HEVNER (2007) defines each of these cycles as follows:

Relevance Cycle: is the first cycle in DSR and involves defining the problem to be addressed, the research requirements, and the criteria for evaluating the research results.

Design Cycle: is the one that involves developing and evaluating artifacts or theories to solve the identified problem.

Rigor Cycle: refers to using and generating knowledge. Rigor is achieved by appropriately using foundations and methodologies from a knowledge base grounding the research, and adding knowledge generated by the research to contribute to the growing knowledge base.

According to HEVNER (2007), the Relevance Cycle bridges the contextual environment of the research project with the design science activities. The Rigor Cycle connects the design science activities with the knowledge base of scientific foundations, experience, and expertise that informs the research project. The central Design Cycle iterates between the core activities of building and evaluating the design artifacts and processes of the research. The author states that these three cycles must be present and clearly identifiable in a design science research project.

The Continuous Planning Adoption Method (COPLAM) is the resulting artifact of this work and it was developed based on the existing literature and studies conducted together with industry.

To define and evaluate the proposed solution, the following steps were followed:

- Literature review to gather information about continuous planning concepts and its application at the project level as well guidelines on how to apply continuous planning in practice.
- Action research study on continuous planning adoption in an agile software development project aiming at identifying strengths and weaknesses regarding continuous planning adoption.

- Development of the method, idealizing the method phases and creating the material to explain and instruct its use, as phases descriptions and artifacts.
- Case study for evaluation of the method, applying COPLAM in industry at the same organization as the action research study but in a different context and with the method execution conducted by someone other than the researcher.
- Improvements on COPLAM according to the results of the case study.

The DSR cycles of this work are represented in Figure 1.

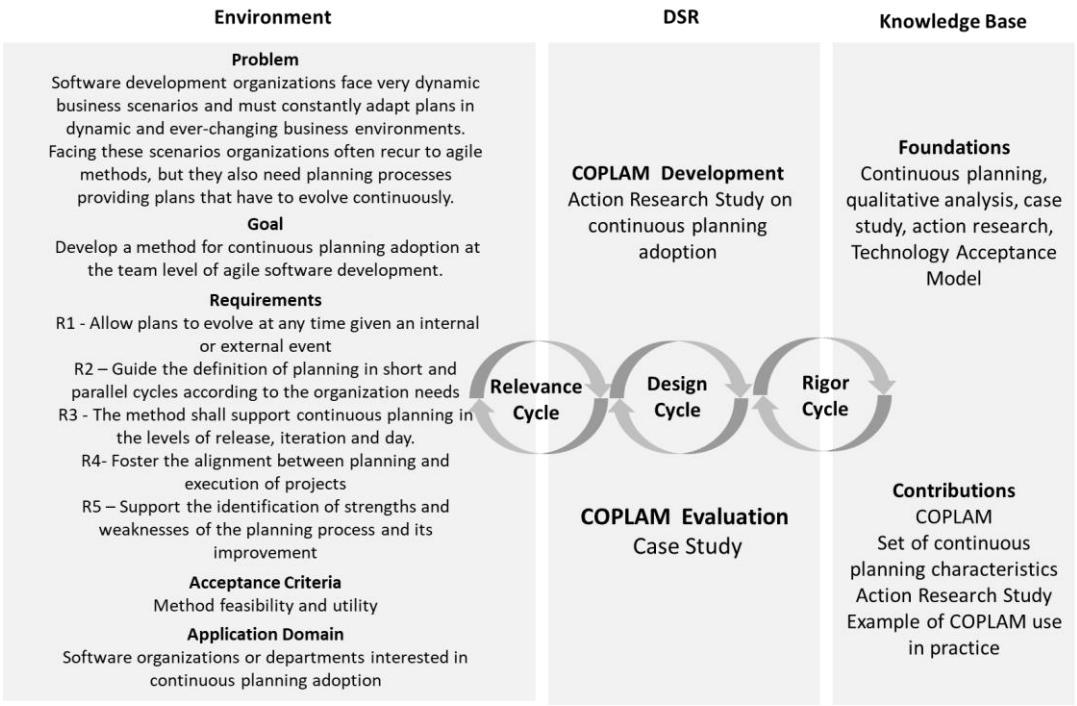

Figure 1. Design Science Research Cycles (Adapted from HEVNER, 2007)

In the Relevance Cycle, the requirements were stablished based on the literature. When defining continuous planning, RICKARDS and RITSERT (2012) and SUOMALAINEN et al. (2015b) state that plans should react to environmental changes, adjusting according to internal and external events (Requirement 1). Also, according to (SUOMALAINEN et al., 2015b), in terms of software development, continuous planning refers to the organizational capacity to conduct planning in rapid parallel cycles (in hours, days, weeks, or months) (Requirement 2) depending on the level of planning (Requirement 3). KNIGHT et al. (2001), SUOMALAINEN et al. (2015b) and FITZGERALD and STOL (2014) state that continuous planning involves a tighter integration between planning and execution (Requirement 4). Finally, based on the previous requirements and in the fact that continuous planning refers to the planning

process in a universe subject to continuous change, where the planning problem is frequently connected to adapting to the universe in which new information is sensed (BRENNER and NEBEL, 2009) we concluded that the method should also provide support on the evolution of the development process to help with continuous planning adoption (Requirement 5).

The acceptance criteria involve the requirements mentioned previously and the feasibility and utility of the proposed method.

The Design Cycle is related to the construction and evaluation of the method. For the construction of COPLAM, an action research study on continuous planning adoption was conducted and for its evaluation, a case study of COPLAM application in industry was executed.

The Rigor Cycle involves the foundations used during the development and evaluation of the method, as the literature on continuous planning, qualitative analysis, action research method, case study method and the Technology Acceptance Model (TAM). This cycle also contemplates the contributions of the research to the knowledge base, which are the COPLAM, the set of continuous planning characteristics identified in the literature, the action research study on continuous planning adoption and the case study of COPLAM used in practice.

1.4. Final Considerations

The current chapter presented the introduction, explaining the motivation for this work, its goals and research question, and the details of the followed research methodology. The reminder of this dissertation is organized as follows:

- CHAPTER 2 – CONTINUOUS PLANNING FOR AGILE SOFTWARE DEVELOPMENT: presents a literature review on continuous planning and its adoption.
- CHAPTER 3 – COPLAM – CONTINUOUS PLANNING ADOPTION METHOD: describes the COPLAM method proposed in this dissertation to guide continuous planning adoption at the team level of agile software development.
- CHAPTER 4 – ACTION RESEARCH STUDY: describes the action research study conducted on continuous planning adoption and its results, which guided COPLAM elaboration.

- CHAPTER 5 – METHOD EVALUATION: presents a case study focused on the evaluation of the COPLAM method in practice.
- CHAPTER 6 – FINAL CONSIDERATIONS: presents final considerations of this work, scientific contributions, limitations and further work.
- APPENDIX I – Presents the consent terms used during the action research study and the case study.
- APPENDIX II- Presents the translated questions used for the Evaluate Planning phase of COPLAM during the case study.

2. CONTINUOUS PLANNING FOR AGILE SOFTWARE DEVELOPMENT

2.1. Introduction

The current business environment of information technology (IT) organizations is very unstable and constantly changing, this have lead organizations to adopt agile and lean development practices (SUOMALAINEN et al., 2015b).

A fundamental focus of Lean Thinking is to reduce waste by removing from the process activities that do not add value (OHNO, 1988). According to FITZGERALD and STOL (2015) there are seven types of waste identified by OHNO (1988): Overproduction, waiting, transportations/hand-overs, too much machining (over processing), inventories, moving (motion) and making defective parts and products. The authors have given examples of these wastes in software development as presented in Table 1.

Table 1. Examples of Waste in Software Development (FITZGERALD and STOL, 2015)

Waste	**Example in software development**
Overproduction	Unwanted features
Waiting	Waiting for build process (compilation) or tests suites to finish
Transportation/hand-overs	Handing over software from an agile development team to a traditional operations team
Too much machining (over processing)	Recompiling unchanged software; running test suites unnecessarily
Inventories	Unfinished features
Moving (motion)	Task switching (e.g. developers working on different projects, losing 'state of mind' whenever they refocus on a different project)
Making defective parts and products	Software defects

The waste of overproduction (i.e. producing something that is unwanted, such as unused product or feature) (FITZGERALD and STOL, 2015) is common in traditional plan-driven software development methods (PETERSEN and WOHLIN, 2010).

Software development has been characterized by harmful disconnects between important activities, e.g., planning, analysis, design, programming and implementation (FITZGERALD and STOL, 2014) (FITZGERALD and STOL, 2015).

According to OLSSON et al. (2013), by focusing on flexibility, efficiency and velocity, agile development practices led to a paradigm change about how software development is executed. Even if those practices were successful, they are not the final step of software development (OLSSON et al., 2013).

Research on improving agile methods has tended to focus on the software development function within organizations and little attention has been given to functions as planning, deployment, operations and maintenance (FITZGERALD and STOL, 2015). Whereas agile software development is mostly focused on the software development function, Lean Thinking focus on the end-to-end process: from customer to delivery (FITZGERALD and STOL, 2015).

According to SUOMALAINEN et al. (2015a), many software intensive companies consider applying lean principles in their businesses. Their research lists eight major challenges caused during lean transformation that are related to communication, strategy, customer value, organizational structure, culture, leadership, learning, and transparency.

Flow is a central concept within Lean Thinking (WOMACK and JONES, 2003) and refers to a connected set of value-creating actions, once a product feature is identified, it is immediately designed, implemented, integrated, tested and deployed, stablishing. Establishing a continuous flow should not be related only to a particular software development function in isolation, instead it should be an end-to-end concept that considers functions in an organization such as planning, deployment, maintenance and operation (FITZGERALD and STOL, 2015).

The concept of flow is connected to continuous software engineering due to the need for software engineering to stablish a continuous movement rather than a sequence of discrete activities, performed by distinct teams or departments (FITZGERALD and STOL, 2015). Continuous approaches are needed in software and business development and continuous planning is one of them (FITZGERALD and STOL, 2015; FITZGERALD and STOL, 2014; SUOMALAINEN et al., 2015b). The focus of the

present work is continuous planning adoption, in this chapter we present a literature review on continuous planning and related works on continuous planning adoption.

In the traditional planning model, a failure in the play may require another cycle of planning activity before it is resolved (FITZGERALD and STOL, 2014). The adoption of lean and agile practices has encouraged organizations to change the way they execute product, service and business development and a clear need for continuous planning exists since organizations face difficulties in developing long term plans due to constant changes in clients and market basis, as well as development of products and technologies. Besides that, recent financial crisis has led companies to rethink their planning approaches and realize the importance of continuous planning from both financial and operational perspectives (SUOMALAINEN et al., 2015b).

When starting the literature review a systematic mapping was considered, but continuous planning in software development is relatively new and due to this fact, few results were found. Using the research string: "continuous planning" and "software development", updated in September 2017, the following results were found: Compendex base 6 results, Scopus base 8 results, ACM Digital Library base 4 results, SpringerLink base 43 results. Excluding duplicates, the total results were 53. After reading some abstracts we found that some articles were focused on budget planning for instance, and not project planning. Given that in the literature there was few results and in a systematic mapping fewer results would be considered after exclusion criteria use, we decided not to pursue a systematic mapping but execute a simple literature review and extract from it a set of continuous planning characteristics. To cover the most relevant part of the literature without executing a systematic mapping or systematic literature review, applied a backward snowballing search approach using SUOMALAINEN et al. (2015b) work as a starting point. From this literature, we have identified continuous planning characteristics, presented in Table 2. We also used the information gathered as basis for executing the Action Research Study presented in Chapter 4 and the construction of the method.

2.2. Continuous Software Engineering

Harmful disconnects between activities such as planning, analysis, design and programming have been characterizing software development. In the past two decades there has been a widespread recognition that increasing the frequency of certain critical

activities helps to overcome many challenges (FITZGERALD and STOL, 2015). The common adoption of agile methods (KURAPATI et al., 2012; PAPATHEOCHAROUS and ANDREOU, 2014) indicates the need for flexibility and rapid adaptation in the current software development environments (FITZGERALD and STOL, 2015).

According to FITZGERALD and STOL (2015) the continuous practices involved in software engineering are defined by the term "Continuous*". Continuous* considers the entire software life-cycle divided into three main sub-phases: Business Strategy and Planning, Development and Operations. Figure 1 illustrates the authors classification for each continuous practice identified and the respective classification.

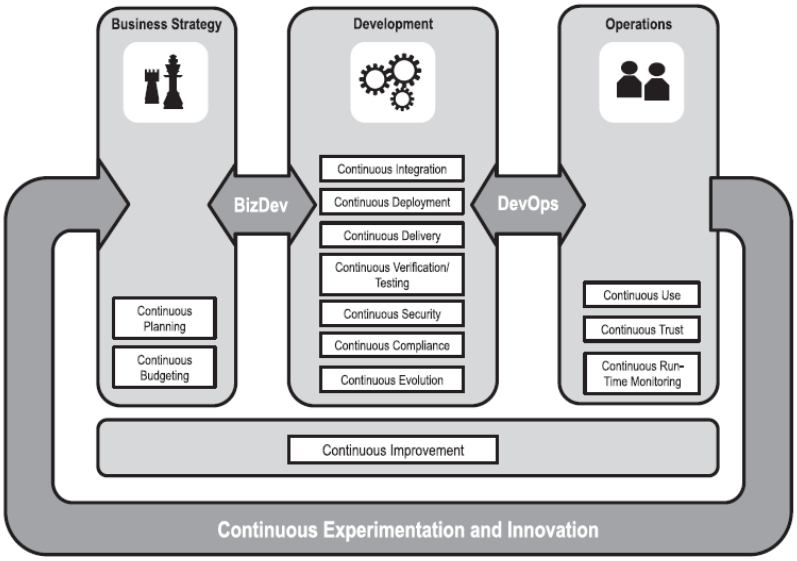

Figure 2. Continuous* (FITZGERALD and STOL, 2015)

The continuous activities identified by the authors are as follows:

- Business strategy and planning: continuous planning, continuous budgeting;
- Development: continuous integration, continuous delivery, continuous deployment, continuous verification, continuous testing, continuous compliance, continuous security and continuous evolution.
- Operations: continuous use, continuous trust, continuous run-time monitoring.
- Improvement and Innovation: continuous improvement, continuous innovation, continuous experimentation.

Figure 1 also presents the concept of BizDev as the link between Business Strategy and Development and DevOps as the link between Development and Operations. The authors argue that continuous planning would facilitate a closer linkage between business strategy and development as it requires a tighter connection between planning and execution.

FITZGERALD and STOL (2015) also identify three main challenges for continuous software engineering:

- Win the war, not the battles: true continuous software engineering is more than adopting continuous delivery and continuous deployment. These are merely techniques, but the ultimate goal is to take a holistic view of a software production entity. Rather than focusing on winning these battles (i.e. successfully implementing such techniques), the holistic view that we advocate is that of winning the war; in this case, to focus on pursuing the Continuous∗ agenda and establish a holistic view from customer to delivery.
- The importance of culture and context: it is not obvious how to establish such a continuous process in a real business environment through the delivery of real and significant new functionality to production systems. A lot of work needs to be done to understand the specifics of different development contexts. The importance of context becomes immediately clear if we take avionics software as an example, in that few people would be willing to fly in an airplane in which a new version of the software was being deployed every 11.6 seconds. There are numerous dimensions in which contexts vary, for instance the business domain in which organizations operate. Similar to what could be observed in some lean transformations, a disbelief that "this could work here" may result in considerable resistance to change within organizations. This cultural change may very well be the most significant barrier to change. Another dimension is that of software sourcing; the use of out sourcing of components or the use of commercial off-the-shelf (COTS) components are very common approaches in numerous domains. Such dependency on software components produced elsewhere may introduce additional challenges when aiming for delivering new software releases frequently.

- Misplaced focus on speed rather than continuity: achieving flow and continuity is much more important in first instance than speed.

2.3. Agile Planning

Planning should not extend beyond a planner's horizon, instead, it should allow time for a planner to pause, examine the changing horizon and make adjustments with a progressively evolving plan (COHN, 2006). Approaches offering adaptability to changes during the project lifecycle are considered more important than predictability (ŠPUNDAK, 2014).

According to SUOMALAINEN et al. (2015b), agile teams achieve this by planning in three different horizons: release, iteration and daily planning. Still according to the authors release planning considers user stories or themes in relation to the new release with the goal of determining the scope, schedule and required resources for a project. A release plan should be updated through a project so that it will always reflect the current expectations as to what will be included in the release. During iteration planning, a product owner identifies the work that a team should address for a new product iteration and daily meetings are meant for organizing work and synchronizing daily efforts.

COHN (2006) defines an agile approach to planning called the "planning onion" (see Figure 3). The planning onion is composed by the following levels of planning: strategy, portfolio, product, release, iteration and day. The levels have a hierarchical relationship with each higher level englobing the lower levels below it. Agile teams focus on release, iteration and day levels.

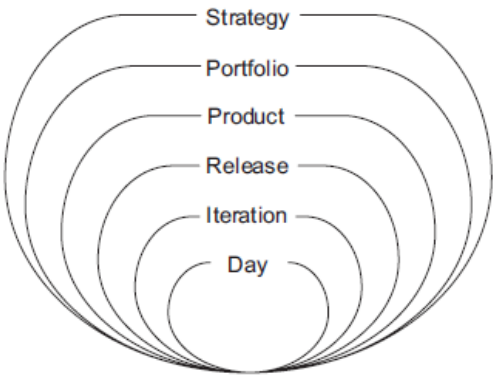

Figure 3. Planning Onion (COHN, 2006)

LEFFINGWELL (2011) presents a framework that considers the planning levels as team level, program level and portfolio level. The framework is called “agile enterprise big picture” and it is depicted in Figure 4 .

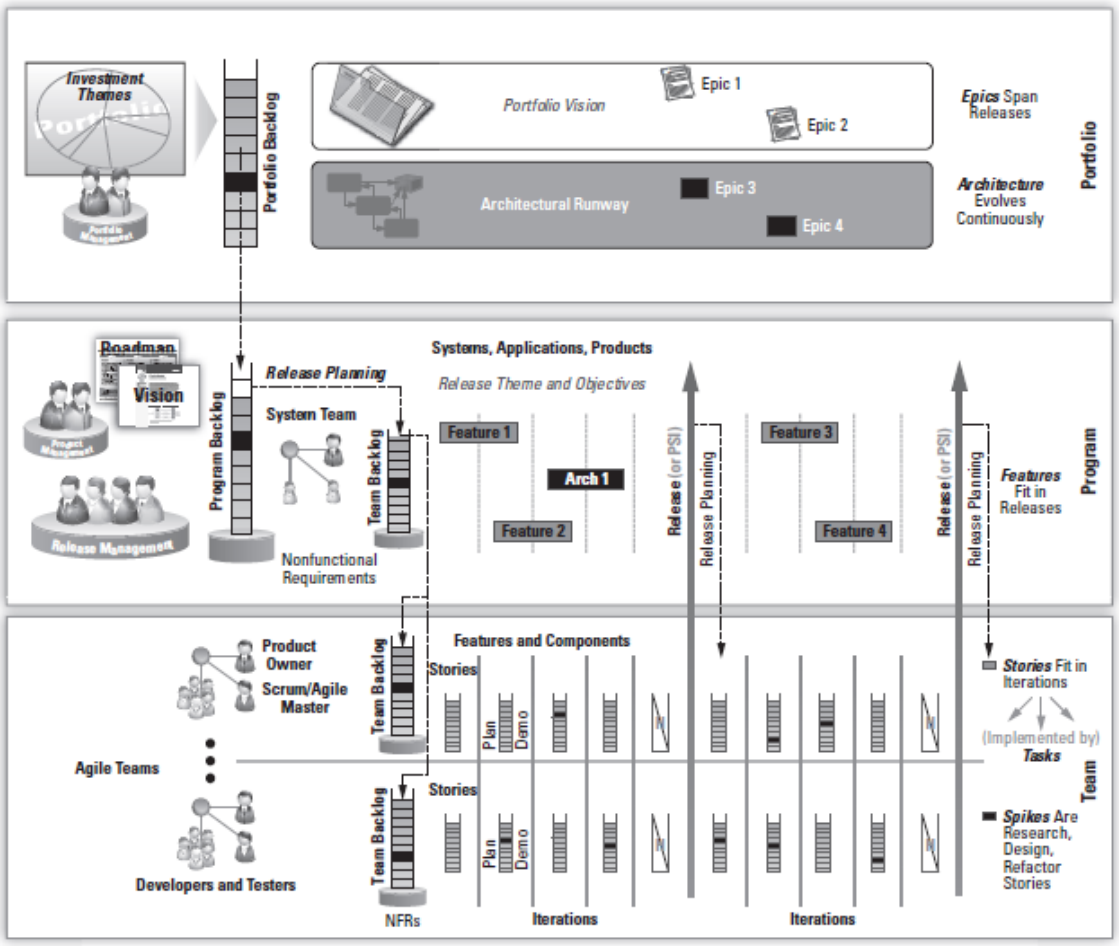

Figure 4. Agile Enterprise Big Picture (LEFFINGWELL, 2011)

According to LEFFINGWELL (2011) at the Team level, agile teams of 7±2 team members define, build, and test user stories in a series of iterations and releases. In the smallest enterprise, there may be only a few such teams. The Product Owner is responsible for managing the backlog of user stories and other things that the team needs to do. At the Program level, the development of larger-scale systems functionality is accomplished via multiple teams in a synchronized Agile Release Train (ART). The ART is a standard cadence of timeboxed iterations and milestones that are date and quality fixed, but scope is variable. At the Portfolio level, the author talks about a mix of investment themes that are used to drive the investment priorities for the enterprise. Investment themes drive the portfolio vision, which will be expressed in as a series of larger, epic-scale initiatives, which will be allocated to various release trains over time.

Scrum is one of the most adopted agile methods in the industry (DINGSØYR et al., 2012), it is designed to add energy, focus, clarity, and transparency to project planning and implementation (SUTHERLAND, 2010). At the beginning of each Sprint the Product Owner and Scrum Team (with facilitation from the Scrum Master) review the Product Backlog, discuss the goals and context for the items, and the Scrum Team selects the items from the Product Backlog to commit to complete by the end of the Sprint (SUTHERLAND, 2010).

One of the pillars of Scrum is that once the Team makes its commitment, any additions or changes must be deferred until the next Sprint. This means that if halfway through the Sprint the Product Owner decides there is a new item he or she would like the Team to work on, he cannot make the change until the start of the next Sprint. If an external circumstance appears that significantly changes priorities, and means the Team would be wasting its time if it continued working, the Product Owner or the team can terminate the Sprint. The Team stops, and a new Sprint Planning meeting initiates a new Sprint (SUTHERLAND, 2010).

Agile approaches for planning consider frequent cycles, but they do not explicit guide the continuous adaptation of plans and Scrum for instance, do not accept altering plans after the start of execution besides in the release level. Continuous planning involves the implementation of planning practices instead of only predefined and regular planning occasions (Suomalainen et al., 2015b), environmental changes trigger planning instead of the financial year and thus, that plans should be adjusted according to internal and external events (RICKARDS and RITSERT, 2012).

2.4. Continuous Planning

Continuous Planning is a holistic endeavor involving multiple stakeholders from business and software functions whereby plans are dynamic open-ended artifacts that evolve in response to changes in business environment, and thus involve a tighter integration between planning and execution (FITZGERALD and STOL, 2014). Continuous planning is about developing planning practices continuously, not just once or twice a year (HOPE and FRASER, 2003).

According to BRENNER and NEBEL (2009), refers to the process of planning in a universe subjected to continuous change, where the planning problem is frequently

connected to adapt to the universe where new information is sensed. Instead of thinking about planning and monitoring execution as separated steps, where one passes information to the other, both can be seen as a unique process. KNIGHT et al. (2001) affirm that providing replanning quickly allows to interlace planning and execution.

Planning can be understood as consisting of two things: actions and forecasts (i.e. expected outcomes). Whereas forecasting can relate to technology or market trends, planning can relate to products, product lines, resources or an entire company (VAN DE WEERD et al., 2010).

MYERS (1999) define plan as dynamic and open-ended artifacts that should evolve in response to an ever-changing environment. These plans should be updated timely in response to new information and requirements to guarantee they continue to be viable and relevant. Plans execution involves more than a blind adherence to previous assumptions, but decision making during execution to adapt, initiate or abandon plans and activities due to considerations made from the operating system. Still according to MYERS (1999), plans should evolve gradually, with minor changes in the environment or current goals that result in proportionally small changes in the plan. Minimizing changes is important, according to the author, to assure the continuousness of the plan.

According to COHN (2006), planning for product development can be organized into several levels as day, iteration, release, product, portfolio and strategy. Explicit planning levels and time horizons help practitioners to analyze the gaps in their planning process between business decisions and requirements engineering (LEHTOLA et al., 2009). DE FRANÇA et al. (2017) conclude from industrial experiences that continuous planning is a feasible approach for dynamic scenarios in which organizations need to adapt to constant changes in market conditions. Risks are discussed continuously, with the main idea being to continuously change the project plan to eliminate risks. Risks feed planning actions and changes to the plan (SUOMALAINEN et al., 2015b).

Continuous planning elements are (i) organizational planning, to define the organizational level of plan and its deadline, (ii) strategic planning, to define a broad plan of an organization, and (iii) business planning, which defines the budget of a plan (SUOMALAINEN et al., 2015b). One of the most important aspects of organizational planning are required planning levels. However, there is no simple answer as to how many

levels of planning a company should have, as both company size and organizational structure play a role in this decision (LEHTOLA et al., 2007).

In terms software development, continuous planning refers to the organizational capacity to conduct planning in rapid parallel cycles (in hours, days, weeks, or months) depending on the level of planning (SUOMALAINEN et al., 2015b). At the project level is done in relation to what is known (e.g., looking two to four weeks forward), the plan is for the next iteration and the work that will be for today.

SUOMALAINEN et al. (2015b) also affirm that various industrial experiences (e.g., LEHTOLA et al., 2007 and LEHTOLA et al., 2009) of companies have shown that they perform open-ended planning with a pre-defined rhythm (SHALLOWAY et al., 2009).

However, while planning can be undertaken at regular intervals, the horizon of the future is not fixed. Company planning is often performed looking only one to two releases ahead, with planning for the near future given greater detail than for the remote future, which is only roughly outlined (SUOMALAINEN et al., 2015b). A more dynamic planning process that is more event-based than calendar-driven with no fixed update frequency and with no fixed time horizons should be developed (BOGSNES, 2008). Continuous feature planning does not mean that you need to change everything all the time. If a change is needed, a set of features may be fixed. Simply by monitoring progress and recognizing that work remaining in the queue matches with the capacity available constitutes continuous planning (SUOMALAINEN et al., 2015b).

Continuous planning may not be possible through the entire organization and can comprehend only specific levels of planning (SUOMALAINEN et al., 2015b). This work comprehends continuous planning on the project level, since according to (FITZGERALD and STOL (2014) and SUOMALAINEN et al. (2015b), in software development organizations that use agile methods, continuous planning is limited to, mostly, to release planning.

We identified continuous planning characteristics based on research found in the literature. These characteristics were an inspiration to build the method and define the phases and activities. Each characteristic is presented in Table 2.

Table 2. Continuous Planning Characteristics

Description	Sources
Support the application of planning practices in a continuous way and not just once or twice a year.	HOPE and FRASER (2003)
Support planning according to environmental or context changes and not only on pre-determined periods.	Rickards and Ritsert (2012)
Support adjustments to plans according to internal and external events.	RICKARDS and RITSERT (2012)
Support the software development planning in rapid parallel cycles (in hours, days, weeks, or months) depending on the level of planning.	SUOMALAINEN et al. (2015b)
Support the understanding that plans are dynamic and open-ended artifacts that evolve in response to ever-changing environments.	MYERS (1999)
Integrate users to the planning process in terms of insights that will influence the type of plan that is generated, the number of options to be considered, the evaluation of failure and strategies for replanning and repairing.	MYERS (1999)
Support the planning of project iterations creating open-ended plans with a pre-defined rhythm.	LEHTOLA et al. (2007) LEHTOLA et al. (2009) SHALLOWAY et al. (2009)
Support planning to be undertaken at regular intervals, but also with a not fixed horizon.	SUOMALAINEN et al. (2015b)
Support practices of governance, leadership, transparency and competency development.	SUOMALAINEN et al. (2015b)
Support the definition of planning levels according to the organization size and structure.	LEHTOLA et al. (2007)
Support development of a dynamic planning process that is more event-based than calendar-driven with no fixed update frequency and with no fixed time horizons should be developed	BOGSNES (2008)
Support the continuous discussion of risks, focusing on alter the plan continuously to eliminate risks.	SUOMALAINEN et al. (2015b)
Support the progress monitoring and recognition that the work left in line matches the available capacity.	SUOMALAINEN et al. (2015b)

2.5. Related Work

AMELLER et al. (2017) present an approach to continuous software release planning focused on continuous integration and the automation of release process but not considering the business environment part of the approach. Also, their approach is partially executed manually by project leader and part automated and does not include other project stakeholders. They define seven type of events that can be detected automatically by a tool during continuous release planning: changes in the dependencies (to existing or new requirements, changes in the effort, changes in the availability, changes in the skills, changes in the cost, risk of overrun and unfeasible solution issue). This approach is still being developed and the tool to identify the events is not developed yet, the proposition is to integrate it with development tools as Git, Trello, Slack, etc.

The Continuous Planning and Execution Framework (CPEF) is a system proposed by MYERS (1999) that employs plan generation, execution, monitoring and repair

capabilities. The system was applied for plans in gaining and maintaining air superiority within a simulated operating environment.

The author defines monitors as event-responses rules for which detection of the specified event leads to the execution of the designated response and these monitors are a critical part of the framework. When considering software development projects, it might be extremely difficult to define monitors since most of event details might not be available before the event occurrence. Also, the response rules may vary even for similar events given the time they occur. We believe this framework can be suited for well-known contexts where events can be mapped before happening but that is not the case for agile software development projects.

SUOMALAINEN *et al.* (2015b), conducted a multiple case study research aiming at identifying current continuous planning methods and practices, focusing on define the meaning of continuous planning and determine how is it perceived in industry. Research questions addressed in the study were "How is continuous planning being conducted in agile and lean software development context?" and "What are the main benefits and challenges of continuous planning?". Three case studies were conducted in three different organizations. From the studies researchers identified as motivators to adopt continuous planning (1) the fact that now-a-days organizational plans cannot remain with fixed focus during a long-term period, as a year or more; (2) organizations business environments are in constant flow and these organizations must adapt to changes and benefit from opportunities these changes can bring; (3) financial crisis in the last decades also caused organizations to rethink their planning approaches; (4) organizations internal problems as conflicts between business and research and development areas, also developers long-term goals to achieve shorter planning cycles; (5) difficulties in maintaining closed scope sprints; (6) and need for more transparency and knowledge sharing. Factors involved in continuous planning highlighted by the authors are: governance, leadership, transparency, knowledge sharing and competency and human aspects development.

The authors present important findings on continuous planning on different planning levels but do not present a structured way of continuous planning adoption.

HEIKKILÄ et al. (2013) describe a case study in a large-scale organization on how continuous release planning was performed by the organization. The process for continuous release planning was characterized by regular scoping and prioritization

decisions and incremental elaboration of features. The authors highlight as challenges the over commitment caused by external pressure, managing non-feature specific work and balancing between development efficiency and building generalist teams. As benefits they describe waste eliminated in the planning process, increased flexibility and decreased development lead time. These study presents an interesting approach to continuous release planning on scaling agile software development, but it does not provide a structured approach for guiding for other organizations to adopt continuous planning.

DE FRANÇA et al. (2017) report an experience of continuous planning adoption at a Brazilian software company. The company originally based on agile practices that started the transition towards continuous planning in a development environment. The work is not a report of a systematic observation like a Case Study or Action-Research, it represents the perspective from the involved people from organization along with the critical point of views of researchers. Concerns about transparency and social aspects such as collaboration and people involvement are identified as key factors when introducing continuous planning. Furthermore, the authors understand from their experience compared to the ones presented by SUOMALAINEN et al. (2015b) that continuous planning has no standardized way of performing the planning activities, requiring the tailoring for specific organizational contexts. The authors highlight that although observing evidence on benefits and challenges, the continuous planning adoption at all levels using a “big-bang” approach may be disastrous. Therefore, they advocate the need for more systematic studies on continuous planning, despite the complexity of observing it into real case environments.

2.6. Final Considerations

This chapter presented a literature review regarding agile planning and continuous planning. Agile teams can plan on three different levels: release, iteration and day. Two approaches on planning levels were presented, the planning onion and the agile enterprise big picture. Also, when it comes to Scrum, one of the most popular agile methods used, the planning focus is on iteration level. Scrum calls iterations sprints, but one of the characteristics of Scrum planning that does not apply for continuous planning is that sprint scope, once defined, is closed. That means that if an event occurs that changes priorities,

the scope cannot be affected, in this case the only solution is a disruptive action on plans that ends immediately the current sprint and a new sprint is planned from the beginning.

Continuous planning is not a common term in software development and there are few studies that addresses its definition, however, while reviewing the literature we have identified some continuous planning characteristics and they were presented in Table 2.

Since there are few studies on the definition of continuous planning, also there is a lack of approaches to guide its adoption. Related works presented are mainly focused only on release planning. This work aims at providing a method for continuous planning adoption for the Team level, defined as release, iteration and day. The next chapter presents our Continuous Planning Adoption Method (COPLAM).

3. COPLAM – CONTINUOUS PLANNING ADOPTION METHOD

3.1. Introduction

Continuous planning is still in exploration by researchers and industry, being a relatively new and not yet well stablished field of research, especially from the agile software development perspective (SUOMALAINEN et al., 2015). There are few empirical researches on continuous planning describing how it is conducted at different levels of planning (SUOMALAINEN et al., 2015). Therefore, we believe there is a need for a more structured way on continuous planning adoption.

Although continuous planning can be applied in several planning levels in agile projects, most of the studies found are limited to release planning (SUOMALAINEN et al., 2015). Planning in the team level involves more than release planning, since not all features delivery are releases, and iteration and daily planning are usually needed. Therefore, a necessity for a more structured view on continuous planning for agile software development is needed. The goal of this work is to create a method to help organizations in adopting continuous planning in agile software development at the team level.

Due to lack of guidance in the literature, the research methodology chosen was Design Science Research (DSR) (Gregor and Jones, 2007). In the first learning cycle, we conducted a literature review. In the second learning cycle, concepts identified in the literature were applied in practice in an action research study that helped building the Continuous Planning Adoption Method (COPLAM). Finally, a case study was conducted to evaluate COPLAM in practice.

This chapter presents COPLAM, its phases, artifacts and actors involved in its execution.

3.2. Method Overview

COPLAM is a method that aims to help organizations to adopt continuous planning in the team level. It is designed to help software organizations that develop software using agile methods to improve their planning process towards a continuous planning dynamic. To apply COPLAM in practice, organizations should be open to adapt their processes and experiment or abandon agile practices to adapt the process to the context and to continuous planning. This is important because some agile practices may not be aligned with continuous planning characteristics, such as closed scope sprints used in the SCRUM. Continuous planning is about adapting to change at any moment, so, closed scope iterations are not adequate to continuous planning.

COPLAM is depicted in Figure 5 to show its phases and Figure 6 and expands each phase with activities and artifacts used and produced. Each of the phases are presented later in individual figures with more details. The method has four phases and each phase is represented in a different color. The figure presents the activities, incomes and outcomes associated to each phase. COPLAM phases are Elicit Context, Define Planning Cycles, Execute Planning Cycles and Evaluate Planning.

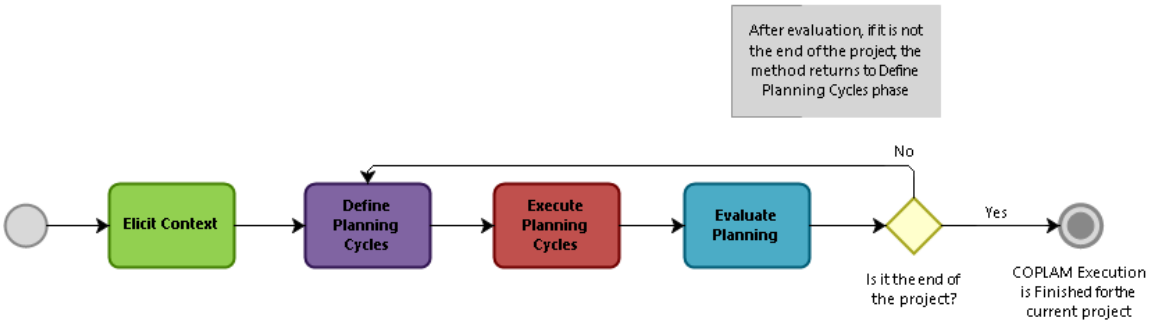

Figure 5. COPLAM Phases

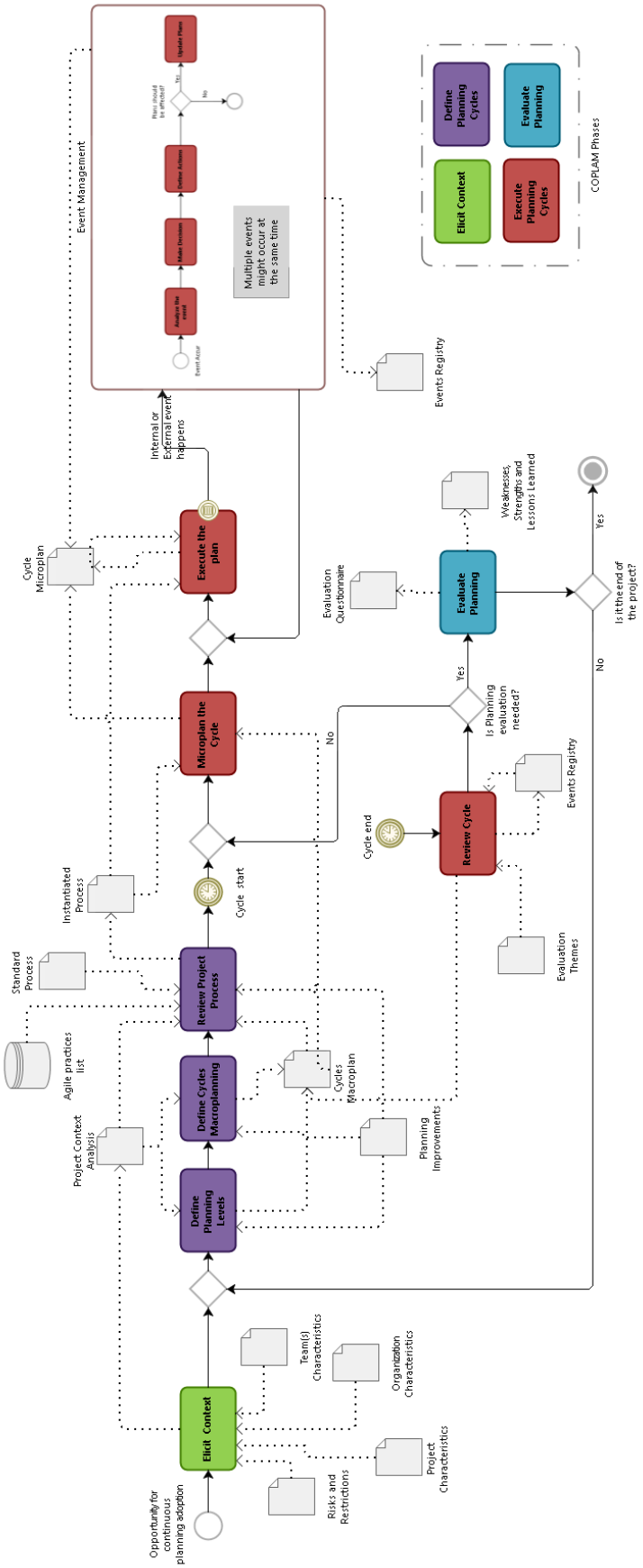

Figure 6. COPLAM Phases Detailed

To better understand COPLAM and the terms used in its description we present important term definitions in Table 3.

Table 3. Important term definitions

Term	Definition	Examples
Project	The definition of project is flexible as it varies according to organization characteristics. According to the Project Management Institute (PMI) a project is temporary in that it has a defined beginning and end in time, and therefore defined scope and resources (PMBOK, 2013). Besides PMI definition, projects can also be continuous, as a set of product functionalities with the goal of delivering or aggregating value to clients or the organization. In this case the project might not be previously attached to a time limit to any of its deliveries.	Examples of project scope include: 1 - Deliver new product module 2 - Perform maintenance 3 - Migrate technology 4 - Implement one or more specific requirements 5 - Create AB Test 6 - Solve technical debt
Project Management Plan	A project management plan is a formal document that defines how the project is executed, monitored and controlled (PMI, 2013). COPLAM does not support the definition of a project management plan as defined by the PMBOK. Therefore, the organization or project manager may use anyone that fits better on its needs.	Examples of project plan information include: 1 - Baselines for Scope, Schedule, Cost 2 - Management Plans for Scope, Schedule, Cost, Quality, Human Resources, Communications, Risk and Procurement 3 - Requirement management plan, Change management plan, Configuration management plan, Process improvement Plan
Project Process	Software development organizations usually have a standard process for creating and delivering new software and managing software projects. The project process is the instantiated version of the standard process of the organization, applied to a particular project or set of projects.	Examples of activities in a project process include: 1 – Elicit Requirements 2 – Elaborate tests 3 – Prepare environment 4 – Execute Tests
Planning Levels	Planning levels define the granularity of items to be planned and the deliverables that are expected. Also, depending on the level of planning different people will be interested in its plans. Someone who is interested in release planning for instance might not be interested in day planning. The level of details needed and available for release planning is different than for iteration or day, therefore the granularity of plans is different too.	Examples of planning levels include: 1 - Day 2 - Iteration 3 - Release
Cycle	A cycle is the planning of activities to be executed during a determined period. Each cycle is revised with a	A cycle associated to: 1 - a specific release

Term	Definition	Examples
	pre-defined periodicity but the planned horizon is not fixed. Also, a cycle can be revised at any moment if a need for that is identified and not only in a pre-defined time. Each cycle is related to a planning level (e.g. strategy, portfolio, product, release, iteration, day). COPLAM focus on team level planning that comprehends release, iteration and day. Cycles can be time-oriented but during the cycle's execution this period can be adjusted. Also, cycles can be parallel in one project, having more than one cycle in execution at the same time. Planning cycles are a fixed period to review plans, but this does not mean that changes cannot be done during an execution of a cycle. As a matter of fact, this is exactly what continuous planning aims to achieve, i.e., the capacity of changing plans at any time should an internal or external fact leads to that. Even with the ability of changing plans at any moment, a fixed agenda provides a more complete and deep review of planning and execution. Planning cycles should be related to project milestones as teams' goals, feature deliveries or software releases.	2 - a specific iteration 3 - a period of one day 4 - a specific milestone: focused on a critical set of functionalities to be developed, that can be a scope of days, iterations or releases.
Event	An event is the occurrence of a fact, internal or external to the organization, that generates a new need or opportunity. An event can occur during the execution of a cycle and might impact on the scope of the current execution of the cycle or in future ones. Also, an event can impact cycles in different planning levels at the same time (e.g. release and iteration). In COPLAM events are classified according to their impact and there are two types: 1) Rapid Resolution Event (RRE): atomic and easy to treat actions that are quickly solved. 2) Long Resolution Event (LRE): new needs or business opportunities that change what is currently being developed or introduce new items to develop in the current cycle or future ones.	Examples of events include: RRE events: 1 - Small maintenance issues LRE events: 1 - New legislation approved 2 - Feature released by competitor 3 - New technology available

Different authors define multiple planning levels. Cohn (2006) defines Strategy, Portfolio, Product, Release, Iteration and Day as planning levels. Leffingwell (2011) considers Portfolio, Program and Team levels and distributes releases inside the Program level and Iteration and Day inside the Team level. In COPLAM we focus on Release, Iteration and Day as planning levels.

As stated in Table 3, the definition of project can vary. When the organization works on projects in a continuous way and not a well-defined and limited way (as described by the PMI), COPLAM does not need to be executed for each project, instead it should be executed to define planning as a standard for all or most projects.

When executing the method, if the project context changes during the execution, e.g. if the change is drastic and deep in the context, one might consider starting a new execution of COPLAM. This might be necessary because if the context changes drastically the project needs to change drastically. If changes are small and gradual, they will be assimilated and treated during the phases Define Planning Cycles, Execute Planning Cycles and Evaluate Planning.

3.3. Roles involved in COPLAM

During COPLAM execution two roles are involved. Table 4 presents roles and the respective responsibilities.

Table 4. Roles and responsibilities involved in COPLAM

Role	Profile	Responsibilities	Skills
Planner	An executor can be someone from inside the project as a technical leader, project owner, project manager or scrum master.	Planner is responsible for method application and the overall continuous planning adoption. The Planner is involved in all stages. Also, this role must have autonomy to propose changes in the project and process.	The Planner should have knowledge of planning and development practices. He/She needs to have critic and analytic rationale to propose changes on project process, planning cycles definition and cycle plans when needed.
Team	The people that design and develop the software. A project can involve one or more teams and team members can be of different technical expertise (e.g. design, user experience, front-end development, back-end development, testing, etc.).	The team is essential in Definition of Process Planning, Planning Cycles Execution and Evaluation of Planning stages.	COPLAM does not require any specific team skills besides the ones associated to the activities performed by team members.

3.4. COPLAM Phases

This section explains the details of each method phase. COPLAM is divided into four phases: Elicit Context, Define Planning Cycles, Execute Planning Cycles and Evaluate Planning.

Phase 1: Elicit Context

The goal of this phase is to identify if the project context is suitable for continuous planning adoption. Also, if the Planner is someone outside the project or the organization,

this phase helps she/he to be familiar with the context and the needs of the project. Figure 7 presents the details of this phase.

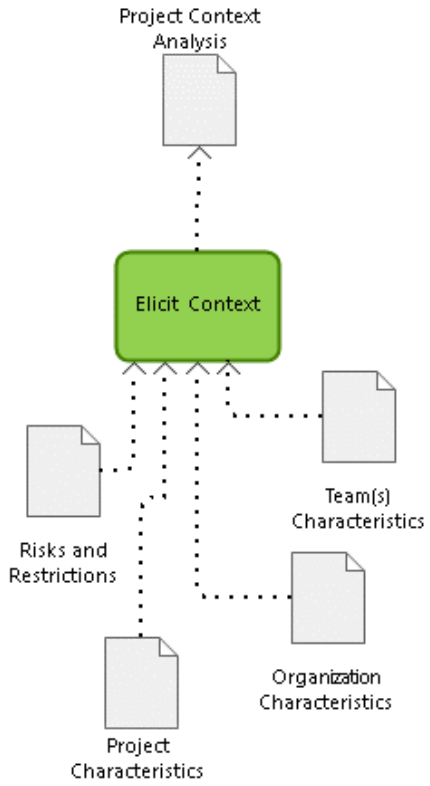

Figure 7. Phase 1: Elicit Context

The deliverable of this phase is the **Project Context Analysis**: this artifact is the result of the phase Elicit Context. It has information about the organization, the project and team's characteristics, also previous problems regarding planning, projects risks and an analysis of the context. All information should be gathered or produced by the Planner. The content of this artifact is explained in Table 5.

The Planner elicits and analyzes the characteristics of the organization, the project and involved team's characteristics to understand the context of the project and identify where are the main motivation for adopting continuous planning. The executor must gather information about the current context and understand the needs of the project and its teams regarding planning as presented in Table 5. This phase is executed at the beginning of the method, when the project is adopting continuous planning for the first time. Although not depicted in Figure 6, it can be revisited whenever a major change in the project's context occur. Table 5 explains the information to be gathered and the rationale to why the information is important.

Table 5. Project Context Analysis

Information	Description	Rationale
Organization's Characteristics	Set of organization characteristics relevant for understanding the business context in which the project is executed and any characteristic that might affect macro planning cycles definition or project process review (e.g. organization size, number of employees, business focus, organizational planning levels, if teams are geographically distributed, if organization is subjected to any specific legislation that can affect continuous planning adoption, if there are any norms that projects must comply with).	Listing organizations characteristics helps understanding the context and identifying the proper planning levels.
Project's Characteristics	Set of project[1] characteristics relevant for understanding the environment and focus of the project or any characteristic that might affect macro planning cycles definition or project process review (e.g. project goals, technical aspects, business aspects, scope, milestones, important dates)	Listing project's characteristics helps understanding the context and identifying the proper planning levels, milestones and periodicity of the cycles.
Teams' Characteristics	Set of characteristics of people that work in the project that can affect project process review (e.g. number of teams, number of members in each team member's profile, experience, function in the company, role in the project)	Listing teams' characteristics helps identifying changes in the process that can better suit the teams' reality.
Previous problems regarding planning	Problems regarding planning that occurred previously in the current project or similar projects and that are likely to happen again. Describe the problems and for each one list possible solutions, teams and roles impacted by it.	Listing planning problems can help identifying changes in the process that could mitigate them.
Risks and Restrictions	Set of restrictions that can impact project's planning and execution (e.g. not having all the information about the scope, suppliers schedule for delivering information needed for development, important dates as Black Friday). For each restriction provide a description, classify it in business or technical restriction, inform if it has impact on planning, execution or both and explicit teams and roles impacted by it. Also explain the risks related to the project. Risks can impact in the macro planning cycles definition and project process review.	Restrictions might influence the periodicity of a cycle and the process review. For example, if deploys of the software happens every week, the release cycle cannot be shorter than that. Risks can influence the cycles macroplan or the process. Also, in the future the occurrence of a risk can be an event. Identifying possible risks can help dealing with them in the future.
Main motivation for adopting	Motivation to adopt continuous planning in the current project based on the previous planning problems, risks and restriction listed above.	Describing the need for continuous planning can help identifying possible changes in

1 The definition of project is flexible, it can be as stated by PMI, temporary in that it has a defined beginning and end in time, and therefore defined scope and resources (PMBOK, 2013) or continuous, as a set of product functionalities with the goal of delivering or aggregating value to clients or the organization. In this case the project might not be previously attached to a time limit to any of its deliveries.

Information	Description	Rationale
continuous planning		the process and the definition of planning cycles.
Analysis of the information gathered	Planner analyzes all the information previously gathered and describes the critical points of the context that should be considered when planning the project(s). The analysis must highlight the main challenges regarding the project(s) and team(s) described before, summarize the business context, its needs and characteristics to be considered when planning the project(s).	The analysis summarizes the main needs and characteristics of the current context to be considered when planning project(s). This will help to define planning cycles and review the process.

After gathering information about the project context, the Planner describes the main motivation for adopting continuous planning to adopt continuous planning, and finally, analyzes the collected data and produces the analysis of the information gathered, which summarizes the main needs and characteristics of the context to be considered during planning. This will help the Define Planning Cycles phase.

Phase 2: Define Planning Cycles

After the phase Elicit Context, the phase Define Planning Cycles takes place. This phase is when the planning levels and cycles are defined and project process is reviewed. These activities consider the Project Context Analysis produced in the Elicit Context phase, the Standard Process of the organization and/or the Planning Improvements identified in the Evaluate Planning phase. Figure 8 presents the details of this phase.

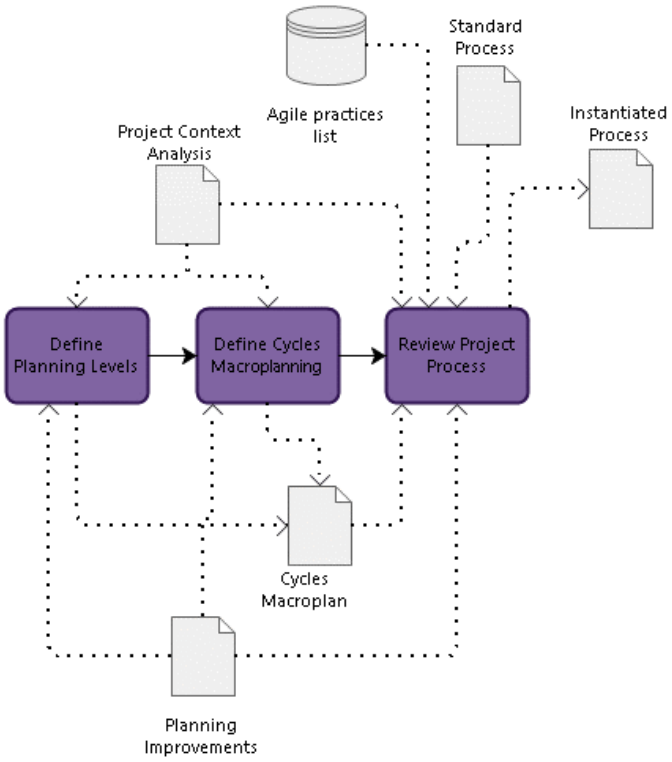

Figure 8. Phase 2: Define Planning Cycles

This phase requires the involvement of someone with autonomy about the project's routine, that is why the Planner needs to have a role as a project manager, project leader, scrum master or product owner.

Phase 2 - Activity 1: Define the planning levels

The goal of this activity is to define which levels of planning are going to have planning cycles. The planning levels can vary according to organizational structure. In the literature authors define planning levels in different terms. In COPLAM we consider possible the levels of release, iteration and day.

The Planner uses the Project Context Analysis to define the planning levels necessary, mostly they are daily, iteration and release planning, but they can vary according to the structure of the organization and the teams. To decide the planning levels, the Planner should consider the granularity of items to be planned, for example release planning requires less details than iteration planning. So, it is necessary to consider in which detail plans should be done.

The result of this activity is the list of planning levels chosen by the Planner. Table 6 defines the levels considered in COPLAM and granularity of items to be planned in each level to help in this activity. COPLAM does not define a template for each of the granularities listed, we believe the format of describing each one should be chosen by the organization and agile practices to help that can be chosen when the project process is reviewed. At this point the information produced is only the choice of planning levels. No documentation is needed yet. Further the Planner will define the planning cycles for each planning level chosen.

Table 6. Planning Levels Details

Level	**Granularity**
Release	Is planned in the granularity of features to be delivered to customers (internal or external to the organization). It is the lowest level of detail among the planning levels described.
Iteration	Is planned in the granularity of activities to be done by the Team during the iteration period. The iteration must have deliveries, but they are not necessarily features of the product.
Day	Is planned in the granularity of tasks to be done by Team members during the day or days planned. It is the highest level of detail among the planning levels described.

This is the first step towards defining the planning cycles. For each of the levels chosen in this activity at least one planning cycle will be defined.

Phase 2 - Activity 2: Define Cycles Macroplan

The goal of this activity is to define the structure and list of items to be planned of planning cycles for each level of planning. The deliverable of this activity is the **Cycles Macroplan**: Cycles macro planning is a set of items to be executed in next few cycles. The items are described in a high level of granularity because there is not much detail about what must be done yet. This artifact details are present further in Table 7.

According to the periodicity of the cycle, the items are grouped in sets that may last that periodicity to be executed. A cycle macro planning is created when planning cycles are defined and can be later updated during each cycle planning and execution. It can be a backlog of all items to be executed or future cycle executions roughly outlined.

Examples of items in a macro planning include:

1 - Create new email marketing

2 - Integrate with determined platform or supplier

3 - Develop functionality X

4 - Solve problem Y

There are six possible planning levels: Strategy, Portfolio, Product, Release, Iteration and Day. COPLAM only supports levels Release, Iteration and Day. Each planning level must have at least one planning cycle associated. Each planning cycle is related to only one planning level. One team can be involved in many planning cycles and each planning cycles should have a least one team. The team(s) is(are) responsible to produce a set of deliverable items, that can include features, stories or tasks. Each higher-level cycle includes the immediate lower level one. Figure 9 illustrates these relationships.

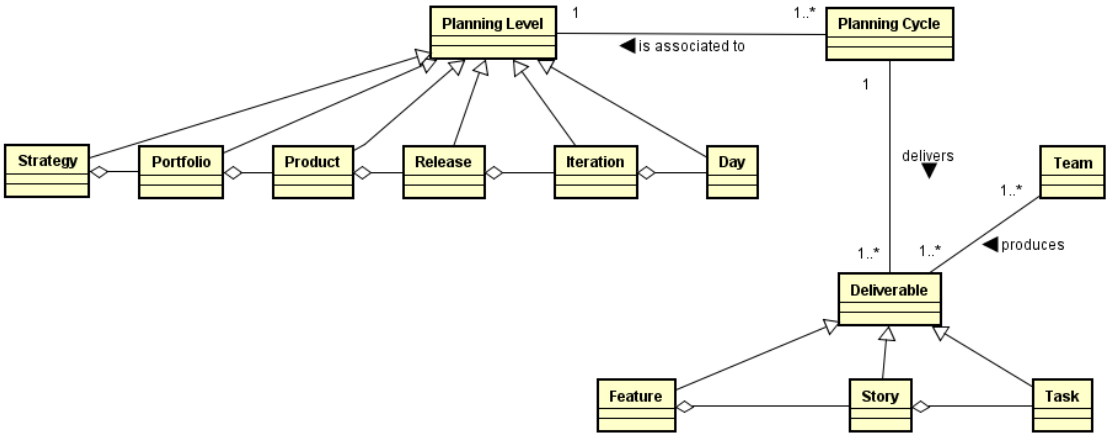

Figure 9. Planning levels, cycle and team relations

A group of cycles of any planning level that is time oriented is also possible to be planned. In COPLAM we call this type of cycle a Milestone. A Milestone is a given period of days or weeks that are planned in terms of deliveries that can embrace releases, iterations and/or days. Figure 10 illustrates releases, iterations, days and milestones cycles.

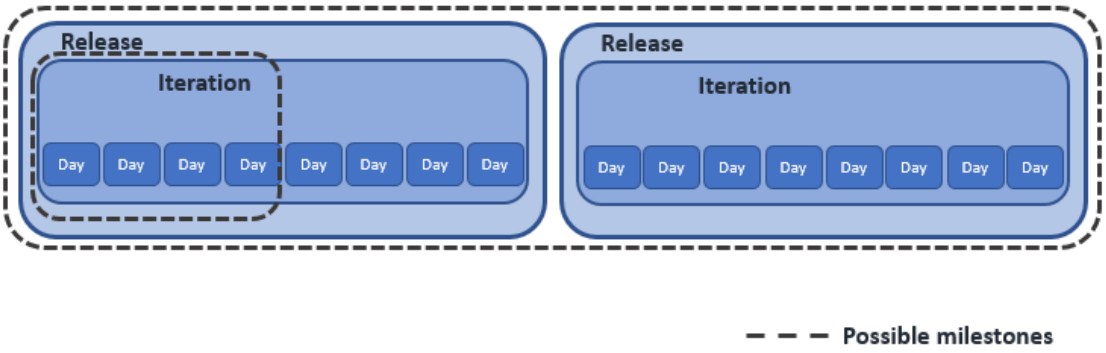

Figure 10. Planning Cycles Hierarchy

A cycle should last long enough to produce a delivery of value adequate to its planning level. For example, in a day level a commit can be a delivery but for a release it is not. To define the cycles, the Planner must consider the restrictions identified in the Elicit Context stage, the important milestones of the project and the planning levels chosen in the previous activity. According to this information, the Planner must define a cycle for each planning level and establish a periodicity for the cycle according to restrictions and project milestones. For each level, the Planner should think about what generates value in this level and how often can this value be delivered considering the restrictions and the needs of the project.

The Planner must define for each planning cycle the periodicity of execution in hours, days, weeks or months. Every planning cycle execution will produce a Cycle Micro Plan during the Execute Planning Cycles phase. Table 7 explains the information to be produced in this activity and the rationale that associates the information with the continuous planning characteristics.

Table 7. Cycles Macroplan

Information	**Description**	**Rationale**
Planning Level:	Defines the planning level of the cycle (e.g. release, iteration, daily). According to the planning level, the type of granularity of the items planned in the cycle is defined (See Table 6).	Defining the planning level helps defining the granularity of planning, the type of deliverables and the stakeholders of a cycle. For example, the managers might be stakeholders of release planning but not for iteration planning.
Granularity:	Is defined according to the planning level. Describes the amount of detail in the items planned in the cycle (e.g. features, activities, tasks). Here the Planner only needs to list the	Defining granularity helps alignment about the level of detail needed for planning the cycle between teams, Planner and stakeholders.

Information	Description	Rationale
	granularity of items for the planning level, for example “Activities and tasks”.	
Periodicity of the cycle:	Defines the timeframe to be considered when planning the cycle. Should be defined in how many hours, days, weeks, months each execution of the cycle will usually last.	Planning refers to the organizational capacity to conduct planning in rapid parallel cycles (in hours, days, weeks, or months) depending on the level of planning” (SUOMALAINEN et al.,2015b)
People involved:	Any stakeholders related to the deliverables or the Team(s).	It is important to list the stakeholders for communication regarding plans and events.
Deliverable	It is produced by actions that delivers value to customers or the organization during the cycle (e.g. release, feature, story, task, commit). Here the Planner defines what is considered as a delivery in the level of planning of the cycle. Deliverable differs from granularity as granularity refers to items planned and deliverable refers to the delivery that is a consequence of executing an item planned.	Defining deliverables helps alignment about the results of the cycle between teams, Planner and stakeholders.
Cycles Macro Plan	List of items, can be a backlog, to be executed in the next few cycles.	This list is useful for future cycles executions and to have an overview of the work ahead.

Cycles Macroplan must be considered when elaborating a Cycle Microplan. Also, when reviewing a cycle, the Cycles Macroplan can be updated. A Cycle Macroplan should be done for release and iteration level. If the organization chooses to plan for day level also, only microplan is needed because it represents a short period and more detail of the work to be done is needed from the start.

Phase 2 - Activity 3: Review Project Process

The goal of this activity is to analyze the Project Context and the planning cycles defined to choose planning and development practices that best fit the project. If the organization have a standard process it should also be analyzed to verify if any change is needed for its instantiation in this specific context.

The deliverable of this activity is the **Instantiated Process**: It represents the instance of the Standard Process that will be used to develop projects in the context described in the Project Context Analysis.

To review the process, the Planner can involve all or some of the team members. The team understands daily challenges and the development process and the Planner has

knowledge about project context and planning cycles defined. Most of this stage is executed by the Planner and, at the end, feedback from the Team(s) is collected to adapt the process proposed by the executor if any change is needed.

The process needs to fit team and project's characteristics and support the execution of planning cycles. In this way, if teams are geographically distributed for example, the instantiated process might need activities for facilitating communication between teams; the process needs to enable deliveries according to the periodicity of the cycles as well, for example, if testing features is a scheduled activity that occurs once a week and there is a cycle that aims at delivering features in less than a week, the process needs to be adapted. Besides that, the context and the planning cycles also matter when choosing agile practices. For example, daily meetings might not be possible if team members work in different time zones, physical Kanban boards are no good for geographically distributed teams, closed scope sprints might not work well with continuous planning because they will not enable plans for the current sprint to evolve during its execution.

The Planner must look at the **Standard Process**: the organization standard process for developing software projects. It can already be documented before COPLAM execution, but if it is not, we recommend that it is documented during the Review Process activity.

If it is already formalized, gather the documentation. If it is not, she/he must document it. The organization can describe its standard process in format. We recommend a graphical representation and a written description of activities, their incomes, outcomes and roles involved. If it is necessary to document the process, we recommend the template in Table 8 and Table 9.

Table 8. Activity Description - Adapted from BARRETO (2011)

Activity	**1. <Activity name>**
Description:	<Description of the activity goal, how is it executed, who executes it and what is the expected results.>

Table 9. Task Description - Adapted from BARRETO (2011)

Task:	**1.1 <Task name>**
Description:	<Description of the activity goal, how is it executed, who executes it and what is the expected results.>
Pre-task:	<If its the case, list the task executed immediately before the current task.>
Input criteria:	<Describe the input information required for the task to be executed.>

Task:	**1.1 <Task name>**
Output criteria:	<Describe the output information produced when the task is finished.>
Responsibles:	<Role(s) responsible for executing the task.>
Participants:	<Role(s) involved in the execution of the task.>
Required artifacts:	<Artifacts that are necessary for the execution of the task.>
Produced artifacts:	<Artifacts that are produced as a result of the task execution.>
Post-task:	<Next task to be executed in the process.>
Tools:	< Tools used to support the execution of this task.>
Agile Practices:	<Agile practices to support this task>

The Planner must describe each of the process activities and tasks, its inputs and outputs, events and restrictions. In order to review the process to support project execution, the following steps should be followed.

After gathering the information about the current process, the Planner must analyze if the process allows the execution of the planning cycles defined or if any change is necessary to allow the frequency of planning and deliveries defined in the cycles. Also, planning problems elicited in the Project Context Analysis should be considered by the Planner when considering any possible changes in the process that could help mitigating these problems.

The next step is to select agile practices to support planning and execution of the planning cycles defined. When selecting agile practices, the Planner should use Agile Practices List: this artifact presents a list of agile practices gathered from the literature. It is used for consultation on agile software development practices and is not produced during the execution of the Method. Table 10 presents the practices, but the practices are not limited to it. If the organization has its own agile practices list, it can also be used.

To select agile practices, the Planner must consider the instantiated process reviewed and the information gathered in the project context. Example: if the project has geographically distributed teams, a physical Kanban board may be a problem. Analyze if there are any other development practices that can support the execution of the project even though it is not directly related to planning as pair programming, TDD, BDD, etc.

Eventually, the Planner should associate agile practices with the activities and tasks of the process that each practice support. A practice can support one or many

activities/tasks. But not all activities/tasks will be supported by an agile practice. Also, by choosing a practice, it might be necessary to alter the process to fit the practice. For example, pair programming is related to coding activity, but testing activity is not related to any agile practices. If it is desired to use Test Driven Development to support testing activity, is also necessary to alter the process to have an activity for test definition before coding. An agile practice can also support more than one activity, for example continuous integration is related to coding, testing and environment preparation. The practices used and the process will vary according to each context. Table 10 presents a list of agile practices classified according to their focus. COPLAM does not support agile practices customization but recommends that each organization experiments the use of the practices according to its needs. Customizations might be tasks related to decisions made regarding events during the execution of planning cycles. Decisions can be altering items in the current cycle execution, altering the macroplan or altering the process.

Table 10. Agile Practices List - Adapted from SILVA (2013)

Agile Practice	**Focus of the practice**
Division in functionalities (features/stories)	Product Requirements
Product Backlog	Product Requirements
Metaphor	Product Design
Coding standards	Product Construction
Collective Code Ownership	Product Construction
Continuous integration	Product Construction
Pair programming	Product Construction
Refactoring	Product Construction
Small releases	Product Construction
Test Driven Development (TDD)	Product Construction
Automated testing	Product Testing
On-site customer	Organization of working environment
Sustainable Pace / 40 hour week	Organization of working environment
Whole team / multi-skilled teams	Organization of working environment
Planning Game	Project Management
Project visibility	Project Management
Retrospective	Project Management
Scrum Meetings and Stand-up meetings	Project Management

Agile Practice	Focus of the practice
Kanban Board	Project Management
Behavior Driven Development (BDD)	Product Construction
Continuous Deployment	Product Construction
Backlog Grooming meetings	Product Requirements

Once the project process is reviewed, the Planner reviews it with the team. It can be in a more formal ceremony as a meeting or sending the documented process from the project process template to team members and asking for feedback. Team(s) give feedback about the process, indicating if any part of it does not fit well or needs improvement. If adjustments are needed, the Planner changes the project process according to the feedback received.

Phase 3: Execute Planning Cycles

Execute Planning Cycles is the phase in which microplan is executed. If the organization has chosen to use new agile practices in the Review Project Process activity, they will start to be used in the first planning cycle after the decision is made. Figure 11 presents the details of this phase.

A planning cycle initiates accordingly to the periodicity defined in the stage of planning cycles definition. This phase is composed by three main activities: planning, when the plan is elaborated given the duration of the cycle, followed by internal and external events identification, where a need of change is identified according to these events, and finally the plan is evolved according to the needs of change.

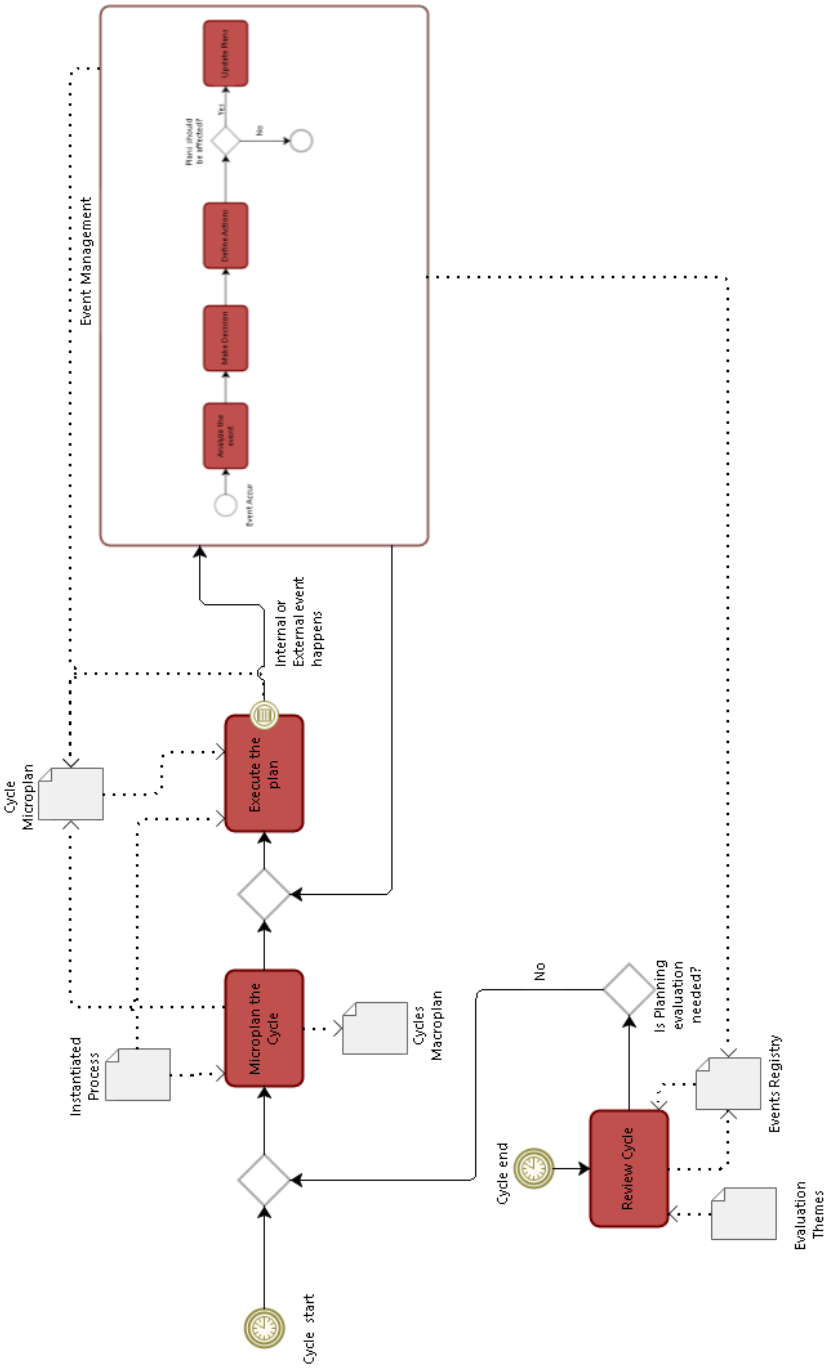

Figure 11. Phase 3: Execute Planning Cycles

Once the plan is evolved, new needs of change can be identified and new evolution is done or the plan continues to be the same until the end of the cycle and the start of a new one. The activities to be executed in this stage are discussed as follows.

Phase 3 - Activity 1: Microplan the cycle

The goal of this activity is to plan the work the team(s) will do during the cycle. The longer the periodicity of the cycle is, the less detail the work the plan will have. The Planner must analyze the risks listed in the Project Context Analysis to see if there is any risk listed that needs to be mitigated at the moment, check if there are any new risks for the project, analyze the backlog of work to be done and the priorities. Next, the Planner should consult the Team or a team member that is a technical leader to understand the amount of work that can be done in the next cycle according to the period it is initially planned to last.

The deliverable of this activity is the **Cycles Microplan**: A cycle micro planning is a detailed view of the macro planning for a specific cycle period. It represents a set of activities planned to be executed during the next period of the cycle. The planning of a cycle is executed according to the frequency defined to the cycle and is the action to determine which activities will be performed in the next period of the cycle. The items that compose a micro planning are features, stories and tasks. Usually stories are related to features and tasks to stories. Tasks can also be independent from stories because they cannot be related to requirements.

Examples of items in a micro plan:

Integrate with platform or supplier (Item from a macroplant)

1 - Read the platform or supplier's documentation

2 - create new server to connect with the platform or supplier

3 - Develop integration

4 - Test integration in test environment

5 - Test integration in production environment

The Micro Plan can be revisited, detailed and updated during the cycle's execution when more information is available for the Team and the Planner. When the Micro Plan is first created, the important thing is to have enough detail that the Team can start working on it.

The work planned needs to be formalized in a list that identifies each item and describes it. For that, the organization might use some tool for issues tracking as Jira. Table 11 presents the information needed in the Microplan.

Table 11. Cycles Microplan

Information	Description	Rationale
Work Item Identification	Short and unique identification of the item to be developed in the cycle execution.	A unique ID facilitates mentioning it and helps communication inside the Team and between Team and Planner. It mitigates the risk of ambiguous understanding of which item is mentioned. Issue tracking tools usually provide it automatically when the item is created.
Work Item Description	Description of what needs to be done.	This is for the Team to understand and when a team member gets responsible for it she/he can develop what is necessary. The level of detail can vary according to the level of planning. Less detail should mean that more detailed work items will emerge in the future in other cycles.
Responsible	Person that is the focal point to talk about the execution of the work item.	It is important to define a responsible for an item to balance how much work each team member has ahead and make plans more accurately. Also, if an event occurs and has actions that will impact in the current cycle execution, people should be notified. If it is something that impacts a specific item it is important to communicate with the responsible for the item.

Phase 3 - Activity 2: Execute the plan

The goal of this activity if for the Team to execute what is planned in the Cycle Plan. Depending on the level of planning, the list will have more or less detail and during the execution might be needed to better elicit the requirements of what must be done. Also, the Cycle Plan informs what must be done but not necessarily who will do each of the items. If that is not yet defined, during execution, each item will have a person responsible for its execution.

Phase 3 - Subprocess: Event Management

In execution of the Cycle Plan, events can occur and impact the plan. For that, every time an event occurs, a subprocess called event management is executed. The goal of this subprocess is to treat the event making decisions about it and, if necessary, updating plans. Multiple events can occur at the same time, so event management can also be multi instance. An event is not only related to changes in scope, events can be any occurrence that impacts on the plan. Some examples of events not related to scope changes include: the hiring of a new professional, an expired tool license or the need for new ones, an unexpected absence of a developer in a workday, etc. Figure 12 presents the Event Management Subprocess.

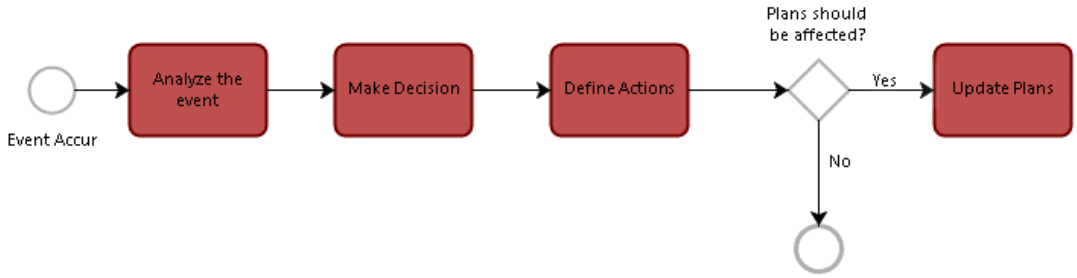

Figure 12. Event Management Subprocess

The deliverable of this sub process is the **Events Registry**: Documentation of the events that occurred in a determined cycle execution and the decision made about it. Every event is documented to compose the set of events to be analyzed in the Evaluation stage. Table 12 presents the information necessary for each event registration.

Table 12. Events Registry

Information	**Description**	**Rationale**	**Activity**
Date	Approximate date of the identification of the event	Time notion to when the event happened helps to recall what happened and how it was treated, it is important for the Review Cycle activity.	Analyze Event
Level of the cycle	Release, Iteration or Day	This information is to Identify in which planning cycle the event was identified. It is important for cycle review and evaluation of planning activities.	Analyze Event
Event	Description of the event and if it is internal or external.	Describing the event helps understanding it and its implications. It is also necessary to be used as reference in the Review Cycle activity and Evaluate Planning phase.	Analyze Event
Type	Definition of the event as RRE or LRE.	The type of event helps understanding its impact as described previously. Also, if a RRE event frequency increases, it might indicate that a different action should be taken and plans should contemplate a new functionality or a bigger effort in maintenance of the system.	Analyze Event
Decision	Description of the decision made regarding the event and the motivators for it. If necessary, discuss other possible solutions and why they were not chosen.	Documenting the decision and the possible solutions considered helps communication about the decision to be spread. Also, it facilitates stakeholders to understand the changes in the plans.	Make Decision
Actions	Actions or tasks to be executed (or	Listing the actions to be taken is important to later updated plans or conduct tasks needed for	Define Actions

Information	Description	Rationale	Activity
	already executed in case of RRE events) to implement the decision regarding the event.	the decision made. Also, it facilitates stakeholders to understand the changes in the plans.	

As explained before in Table 3 there are two types of events: RRE and LRE. Events that are of RRE type can be simple management and monitoring actions. Events that are of LRE type can be risks occurrence, identification of business opportunity, change in clients' needs or new business scenarios as competitors feature releases, new legislations, etc. RRE events should be documented to help identify when their frequency is increasing and there is a need for maintenance or development of a support functionality. RRE events, due to their dynamic nature, can be registered retroactively. LRE events generate bigger impact in the cycle(s) plan(s) and because of that should be registered when they happen. The subprocess Event Management activities are described in the next sections.

Event Management - Activity 1: Analyze event

The goal of this activity is to analyze every event, internal or external, and decide if it will impact existing plans. The event can be identified by the Planner, any Team member or stakeholder.

Once it is identified, it should be listed on the Event Registry and classified as RRE or LRE.

RRE events: these events can be registered at the time they occur or retroactively in the end of the cycle or in the Review Cycle activity. When a RRE event occurs, it needs a rapidly decision and usually the solution for it is already known, for example it could be the execution of a database script to clear some data or the analysis of a log activity from a server, to determine some unusual behavior of the system. RRE events have a sense of urgency since they are usually small problems that need rapid attention to mitigate their impact. Therefore, they might have to be treated before being documented. RRE events are mostly treated by the Team.

LRE events: these events should be registered as soon as possible because their analysis is more complex. A LRE event is a bigger change than a RRE event. LRE events

can be resolved in the current cycle or future ones, it is usually a change in requirements, client's needs, legislation, economy, new business opportunity or need. This type of event requires more time to be analyzed and can involve more people in the decision.

To analyze an event, the Planner must gather information about the event and decide if more stakeholders should be involved in the decision. She/he also needs to verify if the impact of the event will affect the current cycle's execution and/or future ones.

Event Management - Activity 2: Make decision

The goal of this activity is to decide on how to deal with the event. The Planner involves any stakeholders and/or team members to help the decision making. They analyze the possible solutions for treating the event and if the plan of the current cycle or other planning cycles will be impacted and decide what needs to be done. Event impacts can be treated immediately or plans can be adjusted to treat it later in the same cycle or in other cycles. The Planner documents the decision made and the motivators for it. This is helpful to understand, in the future, how the event was treated and why that decision was made. If there is no consensus on the decision from the people involved, someone in charge of the planning, such as a Product Owner or a Project Manager, should act as a mediator and decide. Therefore, it is important for the Planner to have a role like these. Table 12 presented before describes the information produced for each event identified and treated in this activity and in the previous one, Analyze Event. Decisions can be about planning but also about changing the process as using different Agile Practices, for instance. Figure 13 illustrates the relations between events, decisions, actions, plans and tasks.

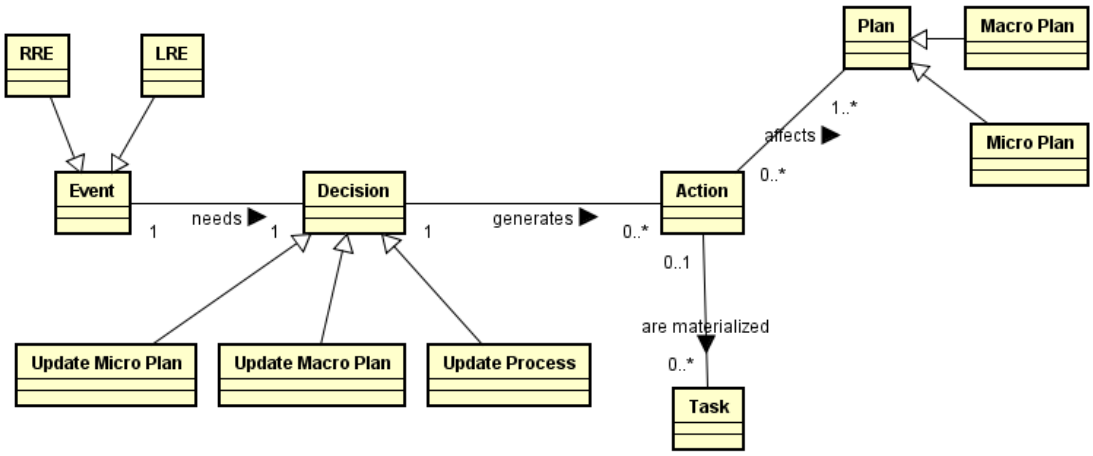

Figure 13. Relations between events, decisions, actions, plans and tasks

Event Management - Activity 3: Define Actions

The goal of this activity is to define which actions will be taken to execute the decision made regarding the event. Actions can be alterations in the microplan of the current cycle, in the macroplan for future ones or in the process. Also, actions can be monitoring or management tasks that will not be reflected in the macro or micro plan as hiring a new developer, buying or expanding a tool license, altering a contract with a supplier or partner, etc.

Event Management - Activity 4: Update plans

The goal of this activity is for the Planner to update plans to reflect the decision made and make them accessible to people involved. The update of plans can be including or removing one or many new activities on the current Micro Plan. If there is a need for change in the Macro Plans, this can be done by including or excluding any activities from the backlog. It is important to notice that not all actions in the previous activity will be updates in plans, for example, an action can be hiring a new developer. Therefore, this activity may not be executed for every event that occurs.

If there is no need for planning update, the cycle's execution can continue without this step. The cycle is executed until all planned work is done, the periodicity of planning is reached or the decision about an event is to stop the cycle and plan for a new one.

Event management can be necessary in any level of planning. Figure 14 exemplifies event management in release level. First in the Macroplan, two releases are planned,

Release 1 and Release 2. Release 1 is planned for starting in July 1 and finishing in July 30 and contains features F1 F2 and F3. Release 2 is planned for starting in July 31 and finishing in August 30 and contains features F4 and F5.

To exemplify how a Microplan is detailed we decomposed feature F1 in histories H1, H2 and H3. The stories could further be decomposed in tasks.

If during the execution of Release 1 an event occurs that introduces the need for delivering a new feature, F6, until July 25 and that happens when F1 and F2 are finished but F3 is not there needs to be a decision to how plans will be updated.

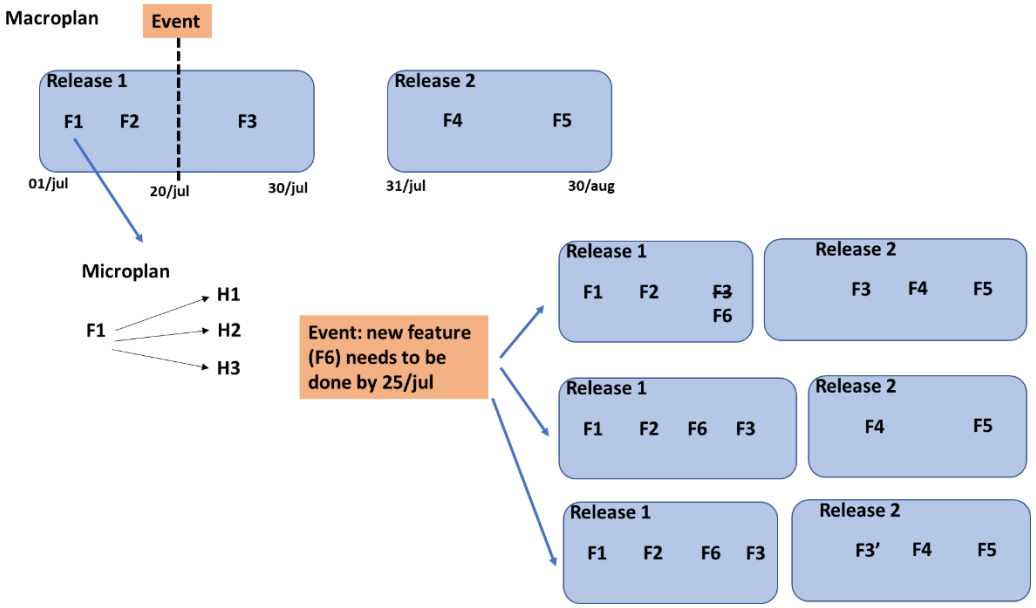

Figure 14. Event Management Example

We present three possibilities:

1. Release 1 finishes when F6 is delivered and F3 is delayed to Release 2.
2. Release 1 is expanded to last more and includes both F6 and F3. Release 2 remains with the same features but starts and finishes later.
3. Release 1 finishes in the time planned with F6 delivered and F3 partially developed. Release 2 starts at the planned time but includes the rest of F3 to be developed and might not end at the planned time.

The example does not consider increase of resources. Its goal is to illustrate that given an event occurrence, different decisions can be made, the more appropriate one will be choice of the Planner and the Team.

Phase 3 - Activity 3: Review Cycle

In this activity, the Planner must analyze the latter cycle execution and decide whether the Evaluate Planning phase should be executed at the moment or the method execution can continue to plan the next cycle.

The Planner must consider the stablished Cycle Microplan and assess the work done and the Event Registry to analyze the events that occurred and how they were treated. After that, she/he should reflect about the proposed evaluation themes and analyze if the current execution of cycles needs improvement in any regarding them. In case of any improvement need is identified, Evaluate Planning will be executed next to better understand the needs with the Team. In case not, the Microplan the Cycle for the next cycle will be the next activity. Table 13 presents the themes that should be considered and an explanation for each one.

Table 13. Evaluation Themes

Theme	Description
Communication	Continuous planning decentralizes planning and that makes communication a critical success factor[2]. The evaluation of this theme is necessary if communication flaws occurred during the cycles execution or any problems could be better solved if communication was better.
Events identification and treatment	As presented in the literature (Rickards and Ritsert, 2012), continuous planning is about adapting plans according to internal and external events. The evaluation of this theme is necessary in case there is any evidence that events were not properly identified and treated.
Planning and execution	As presented in the literature (KNIGHT et al., 2001), continuous planning brings proximity to planning and execution. The evaluation of this theme is necessary if plans could have been adapted more quickly or that events took more time to be identified than they should have taken.
Process Improvement	Project Process should provide ways to identify events and update plans accordingly. Also, it should help communication. The evaluation of this theme is necessary if process improvements could help to address problems with communication or events identification and treatment.
Agile Practices	Agile practices can help planning and execution but some practices are more suitable than others to the teams. The evaluation of this theme is necessary if some agile practices are not being executed properly or team(s) are not using them anymore.

Each evaluation theme is considered in the Cycle Analysis questionnaire (see Table 14) and according to the answers given, the theme will need evaluation or not.

[2] We have identified it as a critical factor in a study described in Chapter 4.

Before answering the Cycle Analysis, the Planner must check if any RRE events happened and were not registered yet. If so, he/she must collect that information with the Team and update the Events Registry. After that, the Planner must consider the initial Cycle Microplan, the deliveries of the cycle execution and the events that occurred and were documented in the Events Registry. With these information in hand, the Planner answers the Cycle Analysis presented in Table 14.

Table 14. Cycle Analysis

Cycle Execution: **Beginning date: dd/mm/yyyy** **Ending date: dd/mm/yyyy**	**Level of planning: ( ) Release ( ) Iteration ( ) Day**	
Question	**Answer**	**Evaluation Theme**
Did the cycle microplan evolve according to events identified during the cycle's execution?	( ) Yes ( ) No ( ) No events occurred	If the answer is "No" the Planning and execution theme must be evaluated.
Were all of the identified events treated properly?	( ) Yes ( ) No ( ) No events occurred	If the answer is "No" the Events identification and treatment theme must be evaluated.
If there was difference between the scope planned and the deliveries, was the difference related to the events registered?	( ) Yes ( ) No ( ) There was no difference	If the answer is "No" the Events identification and treatment theme must be evaluated.
Was the periodicity of the cycle adequate?	( ) Yes ( ) No	If the answer is "No" the Planning and execution theme must be evaluated.
Did any communication problem happened during the cycle's execution?	( ) Yes ( ) No	If the answer is "Yes" the Communication theme must be evaluated.
Does the current process provide ways to identifying and treating events properly?	( ) Yes ( ) No	If the answer is "No" the Process Improvement theme must be evaluated.
Does the current process provide support for communicating events and changes in the plan?	( ) Yes ( ) No	If the answer is "No" the Process Improvement theme must be evaluated.
Is there any problem in the execution of agile practices currently in use?	( ) Yes ( ) No	If the answer is "Yes" the Agile Practices theme must be evaluated.
Is there any agile practice not in use that should be used?	( ) Yes ( ) No	If the answer is "Yes" the Agile Practices theme must be evaluated.

The questions can be answered by the Planner alone or with participation of leaders or the Team. Each question is related to an evaluation theme and according to the answers

given, the Evaluate Planning phase will be executed next with the determined evaluation themes.

After answering the Cycle Analysis, the Planner analyzes if there is any Evaluation Theme to be evaluated according to the answers given, the column Evaluation Theme of the Table 14 explains which theme needs to be evaluated according to the answers. If there is a need for improvement in at least one of the Evaluation Themes, the Evaluate Planning Phase will be executed next to better understand the needs with the Team. If none of the themes were chosen, there is no need for evaluation and the Microplan the Cycle for the next cycle will be the next activity.

According to themes chosen, the Planner will collect information from the Team using a questionnaire. This is detailed in the next section.

Phase 4: Evaluate Planning

The goal of this activity is to analyze the last cycle execution, no matter the level of planning, by gathering information from the Team and the Planner to identify possible improvements. Figure 15 presents the details of this phase.

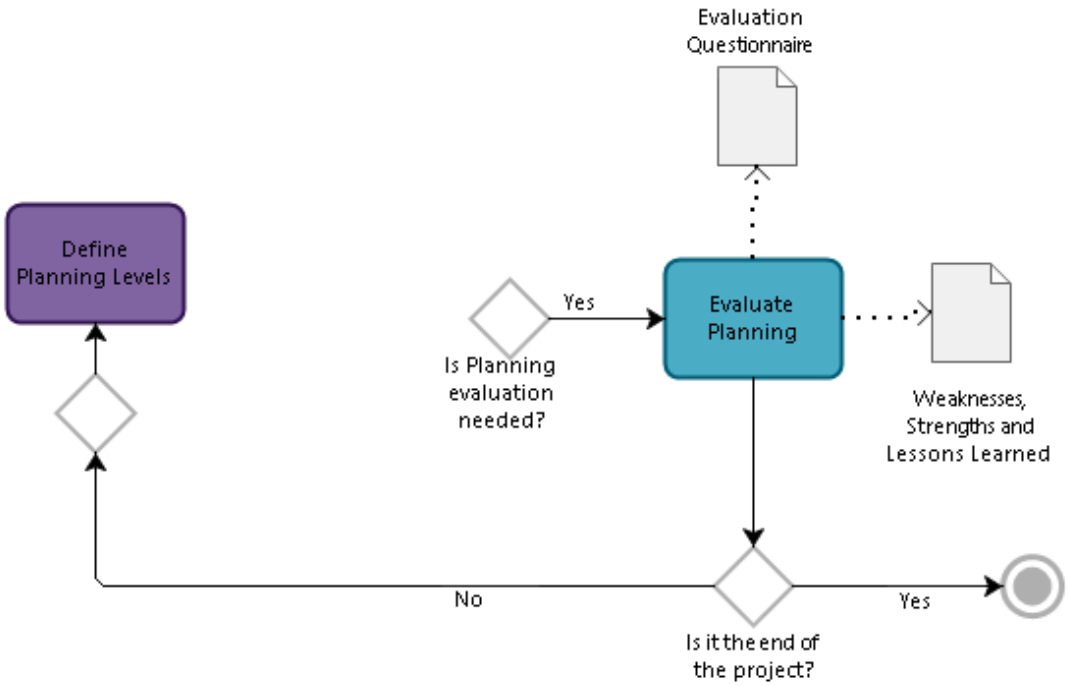

Figure 15. Phase 4: Evaluate Planning

This phase has two deliverables, the evaluation questionnaire and the planning improvements. Each one is explained as follows.

Evaluation Questionnaire: questionnaire to be answered in the Evaluate Planning phase. Its goals are to identify improvements in the process or in the cycles definition.

Weaknesses, Strengths and Lessons Learned: list of weaknesses, strengths and lessons learned and the actions needed to treat it. Type of actions include: (1) change in the planning cycles definition, (2) change in the planning levels, (3) improvement in the project process, or (4) no action needed.

The evaluation questionnaire must be prepared by the Planner containing questions related to each evaluation theme chosen in the previous activity. Table 15 to Table 19 present the questions for each evaluation theme, their goals and classification ("default" or "variable"). Default questions should be asked to the team as they are presented, variable questions must be complemented before being distributed to the Team. Goals and classification columns do not need to be included in the questionnaire as they intended solely to help the Planner executing this activity.

The Planner must execute the following tasks:

1. Set a unique questionnaire gathering all the questions related to the evaluation themes chosen in the previous activity.

2. Adapt the questions that need input from the cycle execution, these are the questions classified as "variables" in Table 15 to Table 19.

3. Review the questionnaire to check if all questions for each evaluation theme needed were properly included.

4. Answer the questionnaire him/herself.

5. Distribute the questionnaire to the Team.

6. Collect all the answers.

7. Analyze the answers to identify improvement needs.

8. Register the improvement needs according to the template in Table 20.

9. Make the results available to the Team.

Depending on the results, the planning cycles or the process project should be reviewed. In this case, the method execution continues to activity Define Planning Levels.

Table 15 presents questions regarding the communication theme.

Table 15. Questions for Communication Theme

Goal	Classification	Question	Type of answer
Identify process weaknesses and strengths regarding communication and improvements suggestions.	Default	How do you classify the quality of the communication between your team and other teams?	( ) Insufficient ( ) Regular ( ) Good ( ) Excellent
	Default	Do you suggest any improvement on communication between teams?	Open-ended.
	Default	How do you classify the quality of the communication between your team and stakeholders from other departments?	( ) Insufficient ( ) Regular ( ) Good ( ) Excellent
	Default	Do you suggest any improvement on communication between your team and stakeholders from other departments?	Open-ended.
	Default	How do you classify the quality of the communication inside your team?	( ) Insufficient ( ) Regular ( ) Good ( ) Excellent
	Default	Do you suggest any improvement on communication inside your team?	Open-ended.

Table 16 presents questions regarding the theme Events identification and treatment.

Table 16. Questions for Events Identification and Treatment Theme

Goal	Classification	Question	Type of answer
Understand if events are being identified at the appropriate moment and plans are adapted accordingly.	Variable	In your opinion, did planning adapted accordingly to internal and external events? (Planner should list here examples of events that occurred during the execution of the cycle(s))	( )Never ( )Sometimes ( )Most of the times ( )Always
Identify improvements suggestions regarding event management.	Default	Do you suggest any improvements when treating events during the execution of what was planned?	Open-ended.

Table 17 presents questions regarding the theme Planning and execution.

Table 17. Questions for Planning and Execution Theme

Goal	Classification	Question	Type of answer
Understand the opinions regarding planning experts' participation in the projects.	Variable	In your opinion, the participation of <give examples of planning experts in your case, like project manager or POs> in the project helped on what?	Open-ended.
	Variable	In your opinion, how can <give examples of planning experts in your case, like project manager or POs> help to improve planning?	Open-ended.
Understand the proximity of	Default	In your opinion, how is alignment between planning and execution of projects?	( )Insufficient ( )Regular ( )Good

Goal	Classification	Question	Type of answer
planning and execution.			( )Excellent
Understand if planning and execution are getting closer.	Default	In your opinion, did project planning and execution became more aligned lately?	( ) Yes ( )No
Understand if the there is need for change in the planning cycles, planning activities of the current process or agile practices used for planning.	Default	How do you classify the frequency of the planning cycles (i.e. the current frequency in which deliveries are planned)?	( ) Insufficient ( ) Regular ( ) Good ( ) Excellent
	Default	In your opinion, should any change be done regarding project planning? If yes, which one(s)?	Open-ended.

Table 18 presents questions regarding the theme Process Improvement.

Table 18. Questions for Process Improvement Theme

Goal	Classification	Question	Type of answer
Identify challenges, strengths and weakness of the current process.	Default	Which part of the current development process do you think was the best?	Open-ended.
	Default	Which part was the worst? If you could change something in the process what would be the first thing you would change?	Open-ended.
	Default	Compared to development process used previously, which are the advantages and disadvantages of the current one?	Open-ended.
Understand possible improvements in any part of the current process.	Default	In your opinion, should any improvement be done in the process? If yes, which one(s)?	Open-ended.

Table 19 presents questions regarding the theme Agile Practices.

Table 19. Questions for Agile Practices Theme

Goal	Classification	Question	Type of answer
Understand if the Agile Practices currently used are adequate.	Variable	For each agile practice used by the Team ask the following question: How do you classify the use of <name of the practice>?	( )Insufficient ( )Regular ( )Good ( )Excellent

The Planner answers the questionnaire him/herself and makes the questionnaire available for every team member with a deadline to answer. After collecting answers from the Team, the Planner should analyze the answers and list the weaknesses, strengths and lessons learned identified.

The Planner must classify and describe each weakness/strength/lesson learned to inform the type of action needed to treat it. Type of actions include: (1) change in the planning cycles definition, (2) change in the planning levels, (3) improvement in the

project process, or (4) no action needed and describe the actions to be taken. Actions can include suggestions from the questionnaire answers or insights from the Planner after analyzing them. After this analysis, the Planner makes the results available to the Team. Table 20 presents what information needs to be elicited for each weakness/strength/lesson learned.

Table 20. Weaknesses, Strengths and Lessons Learned Template

Classification	Description	Type of Actions	Actions
( ) Weakness ( ) Strength ( ) Lesson Learned	Description of the weakness, strength or lesson learned identified.	( ) change in the planning cycles definition ( ) change in the planning levels ( ) improvement in the project process ( ) no action needed	Description of the improvement or change that is going to be done to treat the weakness or value the strength.

3.5. Final Considerations

This chapter presented COPLAM, a method for continuous planning adoption on the team level. COPLAM has four phases: Elicit Context, Define Planning Cycles, Execute Planning Cycles and Evaluate Planning. The roles involved in the method execution are the Planner and the Team. Planner is responsible for all phases and team is involved in phase two and essential in phases three and four. Events identification and treatment are the core of the method, this is what mainly allows plans to evolve according to changes internal and external to the organization. We presented the description of the phases, the roles and the artifacts present in COPLAM, this is information is expected to guide the method use in practice.

In Chapter 2 we have identified continuous planning characteristics from the literature, each COPLAM activity supports one or more of these characteristics. The characteristics and the activity(ies) that support(s) them are presented in Table 21.

Table 21. Continuous Planning Characteristics and COPLAM Activities

Description	Sources	COPLAM Activities
Support the application of planning practices in a continuous way and not just once or twice a year.	HOPE and FRASER (2003)	Define the planning cycles Microplan the cycle Execute the plan Event Management
Support planning according to environmental or context changes and not only on pre-determined periods.	Rickards and Ritsert (2012)	Event Management Review Cycle
Support adjustments to plans according to internal and external events.	RICKARDS and RITSERT (2012)	Event Management Review Cycle
Support the software development planning in rapid parallel cycles (in hours, days, weeks, or months) depending on the level of planning.	SUOMALAINEN et al.(2015b)	Define the planning levels Define the planning cycles Microplan the cycle

Description	Sources	COPLAM Activities
		Execute the plan
Support the understanding that plans are dynamic and open-ended artifacts that evolve in response to ever-changing environments.	MYERS (1999)	Review Project Process Event Management Review Cycle Evaluation of Planning
Integrate users to the planning process in terms of insights that will influence the type of plan that is generated, the number of options to be considered, the evaluation of failure and strategies for replanning and repairing.	MYERS (1999)	Not directly supported, but can be addressed during the Review Project Process activity.
Support the planning of project iterations creating open-ended plans with a pre-defined rhythm.	LEHTOLA et al. (2007) LEHTOLA et al. (2009) SHALLOWAY et al. (2009)	Define the planning cycles Microplan de cycle Execute the plan Event Management
Support planning to be undertaken at regular intervals, but also with a not fixed horizon.	SUOMALAINEN et al. (2015b)	Define the planning cycles Microplan de cycle
Support practices of governance, leadership, transparency and competency development.	SUOMALAINEN et al. (2015b)	Review Project Process Evaluation of planning
Support the definition of planning levels according to the organization size and structure.	LEHTOLA et al. (2007)	Define the planning levels
Support development of a dynamic planning process that is more event-based than calendar-driven with no fixed update frequency and with no fixed time horizons should be developed	BOGSNES (2008)	Event Management
Support the continuous discussion of risks, focusing on alter the plan continuously to eliminate risks.	SUOMALAINEN et al. (2015b)	Elicit Context Microplan the cycle
Support the progress monitoring and recognition that the work left in line matches the available capacity.	SUOMALAINEN et al. (2015b)	Event Management

The next chapter presents an action research study conducted to understand continuous planning adoption and construct the method. The research was executed concomitantly to the creation of COPLAM.

4. ACTION RESEARCH STUDY

4.1. Introduction

After literature review, the second step of COPLAM's construction was executed. It was an action research study on continuous planning adoption in an agile software development project. This chapter presents the study context, details and results.

4.2. Study Planning

We wanted to solve a real problem of the organization, thus characterizing the study as an action research (SHULL et al., 2008). Action research is especially relevant in situations where participation and organizational change processes are necessary (BASKERVILLE and WOOD-HARPER, 1996). From the paradigm of GQM (BASILI et al., 1994), the goal of this study is defined as:

Analyze the adoption of continuous planning

With the purpose of identifying strengths, weaknesses and challenges

Related to continuous planning adoption in the agile software development

From the point of view of the planning and development

In the context of agile software development projects

We defined three research questions as follows.

RQ1 – What are the challenges of continuous planning adoption on the project level? Due to the lack of guidance in continuous planning adoption in projects in the literature, the main question of this study aimed at identifying the challenges faced during its adoption in the project level.

RQ2 – Does continuous planning help to improve agility? Since the literature on continuous planning in agile is mostly limited to release planning we wanted to understand if continuous planning could contribute more in agile projects facilitating the improvement of agility.

RQ3 – How to support continuous planning adoption in software projects? Besides answering RQ1 and RQ2, we also focused on understanding the process of continuous planning adoption as we aim as part of an ongoing research at creating a method to support continuous planning adoption in the project level. We consider this study as an action research considering that is focused on understanding the problem whist while solving it, that one of the researchers proposed changes and improvements during the study and that it mixed research and industrial experience.

4.3. Data Collection

Data collection was planned to occur in 3 different ways:

(i) documentation of decisions made about the development process and the agile projects used and discontinued;

(ii) questionnaire to identify improvement on the development process regarding planning, estimating, requirements specification, and prioritization;

(iii) individual retrospective interview to understand positive and negative aspects of the continuous planning adoption.

Study data was collected by observation of one researcher, a questionnaire after three months of project and then an interview close to the end of the project.

The researcher that was collecting data by observation was also responsible to support the organization during the execution of a software project. This project was the first time the organization was adopting continuous planning. The researcher documented decisions to made changes in the process taken by the teams and proposed improvements when she felt it was needed.

The questionnaire done after the first three months had the goal of understanding how the project was going regarding the decision of changes in the process, planning activities and agile practices that could be useful in the project. The questionnaire was divided into three sections and answering it was not mandatory.

The first section analyzed whether the decisions made about the process and the way teams worked were common knowledge for the members (see Table 22).

The second section had the goal of analyzing the perception towards the activities that were related to planning and the need of more participation of the planning team of the company in this project (see Table 23).

Table 22. First Section of the Questionnaire

Questions
In your opinion which was the benefit expected from this improvement?
In your opinion, were the expected benefits achieved?
In your opinion, is the improvement being executed?
In your opinion, was the decision of implementing the improvement taken in the right moment?
In your opinion, is there another solution that can fulfil the expectation of benefits better than this one? If yes, which one?

Table 23. Second Section of the Questionnaire

Questions
What kind of actions of the planning area could help improving the day-by-day of the team or the execution of the project?
In your opinion, should any improvement in relation to estimates be done? If yes, which one?
In your opinion, should any improvement in relation to planning deliveries be done? If yes, which one?
In your opinion, should any improvement in relation to prioritization of requests be done? If yes, which one?
In your opinion, should any improvement in relation to requirements specification be done? If yes, which one?

The third section showed a list of agile practices with short definitions of each one and asked if the participant thought that each of the practices could be useful for the team, regardless of being already used. The agile practices considered were: Control Task Boars, Short and Frequent Deliveries, Retrospective Meetings, User Stories, Kanban Boards, BDD, Continuous Integration, TDD, Planning Poker, Pair Programming, Scrum of Scrums, Daily Meetings and Burndown Chart. Practices as Sprints and Planning Meetings and Frequent Refactoring.

Close to the end of the project the interviews started. They were semi-structured and comprised with open-ended questions aiming at answering the research questions. For example, to answer RQ1, two of the questions addressed were “Regarding the market place project, which part of the development process do you believe to be the best one?” and “And which part of the development process do you believe to be the worst one?”.

Interviews were recorded and transcript by the researchers. Table 24 presents all questions and the interview script. Since the interviews were semi-structured, the script could include complementary questions created by the researcher at the moment to better understand the participant’s opinion.

Table 24. Interview Script

Questions
Regarding the market place project, which part of the development process do you believe to be the best one? (RQ1)
And which part of the development process do you believe to be the worst one? (RQ1)
Comparing to the previous development process, when there was a unique IT team, sprints of closed scope, planning poker, which are the advantages and disadvantages of the process used this year? (RQ1)
In your opinion, this new work model brought more agility to the software development? What is the definition of agility to you? (RQ2)
In your opinion, the participation of members of the planning team in meetings and daily activities of teams contributed to the project? In which aspects? (RQ3)
In your opinion, this new process there was more proximity between planning and execution? Why? (RQ1)
In your opinion, planning was suitable to changes that emerged during the project? Why? (RQ1)
In your opinion is there a need for process improvement in relation to planning? If yes, which one? (RQ1)
In your opinion is there a need for process improvement in relation to communication inside your team, between teams or regarding other areas of the company? If yes, which one? (RQ1)
In your opinion is there a need for process improvement in relation to development of projects? If yes, which one? (RQ1, RQ3)
In your opinion, team's follow-up meetings helped to give visibility and alignment about project planning and execution? (RQ1)
In your opinion, project's follow-up meetings helped on project planning and execution? (RQ1)
In the previous questionnaire, the most useful agile practices in respondents' opinion were retrospective meetings, short deliveries, task control boards, user stories and Kanban boards. For each one of these, do you believe we use it? Should we start using them? (RQ1, RQ3)

For the data collections, the participants were asked to sign a consent term, the term is presented in APPENDIX I.

4.4. Study Context

We selected a small-sized organization in Rio de Janeiro city and focused on e-commerce which was about to start a 1 year project to evolve its main product to establish a Market Place functionality for the website and the mobile apps. The study was carried out from January 2016 to December 2016.

The organization's main goal towards the clients is to mediate the buying process offering information and helping clients to choose the best fit according to their financial and feature needs. This scenario changes in a very fast pace, new stores and clients access the company's systems every day, product offers must be instantly updated, prices and payments conditions change frequently, new products are launched and old ones are discontinued frequently. The organization had previous experience on agile software development, mainly using Scum practices, also, one of the researchers works in the organization, allowing to follow the project and the teams closely.

The study contemplated two development teams: team A had three backend developers, one of them joined the team during the project, two frontend developers, three user experience members; team B had four backend developers, one of them also joined the team during the project. During most of the project, each team had one coach: team A had a product manager as coach and team B had a commercial director as coach. All team members had previous experience with agile software development. The researcher conducting the study is also a planning team member and could observe closely and propose improvements in the process, although not involved in every meeting or decision about the project and not a member of any of the two development teams.

The project had four milestones of three to four months each. Every three weeks there was a follow-up meeting for every team with planning team, executive director and IT director. The main goal of this meeting was to present what had been done by the team in the last three weeks, what would be done in the next three weeks and address issues of alignment to the organization's strategy. The planning department oversaw coordinating both teams work and tracking the project's plan. During the first three weeks the teams worked freely, without pre-established practices. To guide the work a high-level software development process was created by the planning team. Teams could propose changes in the process. The only restrictions that both teams had to follow was the deploy schedule for one of the software, that occur once a week, and follow-up meetings.

4.5. Results

At the end of the third week of the project and during the first milestone, teams' first follow-up meetings occurred with teams' members, planning team's members, executive director and IT director. The planning team promoted a retrospective review focused on the development process of the development teams to understand challenges and successes of the period and need for improvement.

It became clear that more organization and planning were needed as communication inside team A was difficult due to the number of members and organization and visibility of the work to be done was not clear to everyone. We consider this event as the starting point of continuous planning adoption.

Improvements were made in the process and in the work organization during the whole project execution. They could be proposed by one of the researchers or for someone

inside the teams or the organization. For example, the decision to have someone from the planning area to keep up with each team was made from the start, but other improvements were made later, like the follow-up meeting of the project, that did not include all teams' members but the stakeholders with more decision power regarding the project.

The main actions to process improvement and work organization during the project are presented in Table 25.

Table 25. Actions to improve the process made during the project

Action	Goal
Having someone from the planning area of the company to keep up with each team	Help organize the work and facilitate communication with other areas and give transparence of the ongoing work and project.
Creation of role "IT Head" to have a technical leader inside each team	Help technical decision making and facilitate communication between teams
Having all IT Heads to participate in the teams' follow-up meetings	
Classification of the issues inside Jira tool according to objectives and needs each team had to accomplish or solve.	Organize and give more visibility of the work
Weekly meetings for each team	Prepare and plan the next iteration
Follow-up meeting of the project with main stakeholders	Organize and coordinate the project and make decisions with more velocity

4.5.1. Questionnaire Results

This section presents the results of the questionnaire applied after three months of work on the project. From team A we collected 6 responses out of 9 members, from team B 2 we collected responses out of 4 members.

Process Decisions: most the decisions were considered positive, indicating that the actions proposed were in fact needed. It was pointed out though that the communication of the decisions should be improved, providing more transparency.

Planning Activities: the questions were open-ended addressing activities that involved the planning team. We wanted to understand which improvements could be done in relation to planning. Figure 16 shows the results.

In all cases, at least 37.5% of the respondents indicated that there was need of improvement. Prioritization was the activity that needed more attention, with 75% of the answers and estimation was the one with less need of attention, with 37.5%. Readers should note that no estimating practice was used in the current process version, and the teams did not seem to miss it.

Prioritization was in fact a challenge. In previous projects, it was done by managers and directors. In this project, it became a team's responsibility. The role of the coach was responsible to prioritize project user stories versus the requests of other areas of the company that also had to be developed by the team.

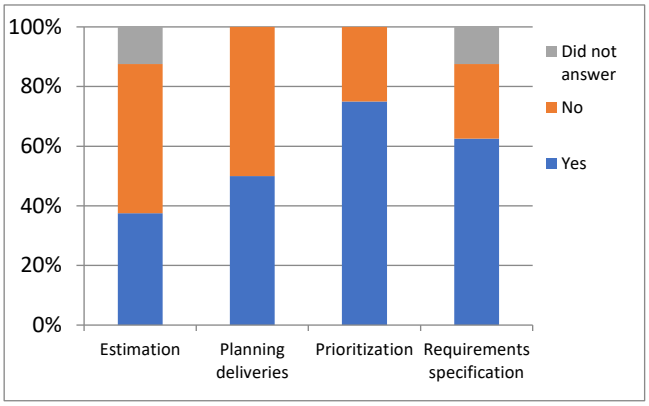

Figure 16. Improvement Needs

Regarding actions of the planning area, 5 out of 8 respondents expressed the need for improvement of communication and 4 expressed needs related to planning and execution. Table 26 summarizes these answers.

Table 26. Improvement on planning actions

Topic	**Answers summary**
Communication	Improve communication between teams
	Catalyze communication with the areas
	Catalyze and interface the resolution of problems and communication between areas
	Optimize meetings
	Status meetings should be done with only planning team and managers
Planning and Execution	Organize the big picture
	Reduce the amount of rework
	Schedules creation and control
	Managing tests schedules
	The best action was already being made, we have people from planning participating actively on teams

When asked about estimates, participants did not reach consensus: 4 of them said that there was no need of improvement, one did not answer and the other 3 expressed need for improvement. One person said that "all elements interested in the deadlines need to be informed about situations where rework can disrupt the prospect of future delivery", this also shows more need for communication. The other two cited the need of a technical leader to help team members to estimate.

About planning deliveries, according to 3 respondents there was no need for improvement, also 5 respondents gave some feedback or suggestions. Regarding requests

prioritization, two people said no improvement was needed and six said that it did. Table 27 summarizes these feedback and suggestions.

When asked about requirements specification, two people said there was no need for improvement, one claimed not to know and five gave suggestions. The suggestions about project requirements were better detailing, understanding of value, more talking to requestors, expectation of impact, justifying the need and in cases where more than one team needs to work in the same request it is necessary to have a previous commitment before beginning the development.

Table 27. Feedbacks and suggestions about planning deliveries

Topic	**Answers summary**
Planning deliveries	Split demands on smaller pieces
	Work on reprioritization
	The team gets frustrated when cannot fulfill the dates given by the company. We need better planning and more estimating.
	Schedules are helping to give visibility of the work
	Should be done with team leaders

Also, participants were asked if any other improvement should be done and which benefit the improvement would bring. Five participants answered "No", "I don't know" or did not answer. Three participants gave suggestions involving prioritizing requests outside the project according to their value and importance, reorganizing the teams' purposes and members and geographical positioning of members, claiming that all IT personnel should be physically together and not divided into teams.

Agile Practices: Since not all team members are developers, some of the listed practices are not applicable to their work. %). Figure 17 shows the results for the 16 asked practices.

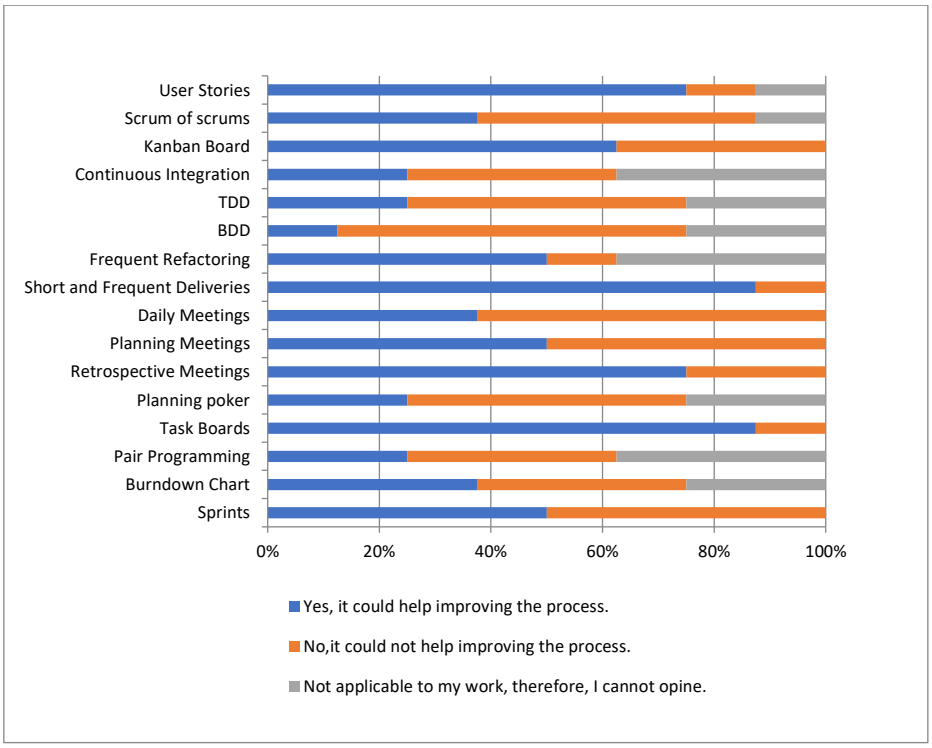

Figure 17. Usefulness of agile practices

The practices that more than half of the respondents considered useful were Control Task Boars (87.5%), Short and Frequent Deliveries (87.5%), Retrospective Meetings (75%), User Stories (75%) e Kanban Boards (62.5%). Practices that less than 50% of respondents considered useful were BDD, Continuous Integration, TDD, Planning Poker, Pair Programming, Scrum of Scrums, Daily Meetings and Burndown Chart. Practices as Sprints and Planning Meetings and Frequent Refactoring were considered useful by 50% of respondents.

During the project, agile practices that were used in practice were: Kanban board, planning meetings, daily meetings, sprints and retrospective meetings.

A retrospective meeting was proposed before the questionnaire and another one after, both did not go well. First one was optional and a lot of team members did not show up. The second one was focused on communication, a problem identified in the questionnaire, but conflicts between team members happened during the meeting and was decided not to continue with it. After that the researcher proposed the final interview to be an individual retrospective to improve maturity for collective retrospectives in the future because at that moment teams were not mature enough for that. The interview was used to provide insights into continuous planning challenges, also it was considered a retrospective for team members.

4.5.2. Interview Results

At the end of the project we conducted individual interviews with team members to understand process weakness and strengths and the planning activities. The interviews lasted from 19 to 47 minutes and 13 people were interviewed. At this point people of different positions were interviewed, for example junior and senior developers, an IT coordinator and UX leader.

When comparing the current process with the past one, most of the interviewees, twelve of them, said that the process was improved, only one person did not perceive any improvements. About the follow-up meetings of the teams, most respondents agreed most of the time there was no need for the entire team to be present, also, the perception was that this meeting as a status meeting instead of an alignment meeting.

Communication between teams was pointed out as something that need to be improved and in some cases interviewees said it was the worst aspect of the process. Also, communication inside the team was an issue for team A, but not for team B. Some participants attributed that to the size of the teams, since team B had four members and team A had nine.

Observing this project, we could understand also that continuous planning differs deeply from traditional planning, for example when a task depends on another one. In this case, the teams could skip de pre-requisite when necessary and create a "shell" of what the team supposed was going to be delivered to them before it was. Later there was reworking to adapt an integrate different parts developed, but the dynamic created by continuous planning made this reworking be embraced as something necessary for the project evolution. Minimum Viable Products (MVPs) were made during the project to meet these restrictions. A lot of rework had to be done because of that, but it allowed the development to go on even with serious restrictions. Continuous planning helped process improvements “on the fly” since members of both development and planning teams made changes to the process during the project’s execution.

In the beginning of 2016, when the new organization of the teams and new development process had started, some people inside a team executed a small software development project and they reported having a great experience in creating and experimenting solutions for the problem they had to solve. Also, participants told us that small projects (with up to 3 team members) in the past had worked well in a similar way.

We can assume that as evidence that continuous planning is easier in smaller teams and projects.

From researcher's perspective, the use of continuous planning on the Market Place project helped the company to deliver a product simpler than first idealized and at the same time filling better the needs of the customers. Continuous planning also helped teams to have a common vision of their goals and commitment to a different dynamic. In the end, the team could deliver high quality products.

One may ask what makes continuous planning different from the traditional agile planning. First, when comparing to Scrum, in continuous planning there is no need for closed scope of sprints or releases, teams have more confidence that they are doing what needs to be done and what is best and viable at the moment. It is possible to build a product without specifying every feature of a MVP (Minimum Viable Product) and this gives more opportunity for innovation.

In the previous process, planning poker was used to estimate stories and during the interviews participants told that it was stressful and they like it better without it. When using continuous planning the focus of estimates can be left to only critical tasks, and there is no fear of estimation mistakes because the plan will evolve according to the circumstances. One of the participants said that this new process carried less stress and more quality to the work.

4.6. Findings

In this section, we discuss the findings of the study and their relation to each of the research questions.

RQ1 – What are the challenges of continuous planning adoption on the project level?

Communication is the key: this was the most difficult aspect of the process. When decentralizing planning, communication becomes more fundamental than ever, because plans are drawn and altered by more people. To face this challenge, the organization defined roles of leadership like the coaches and the IT heads. The coaches were an unsuccessful experience because they had little communication to each other about the teams' work. On the other hand, the definition of IT head role was an excellent improvement according to interviewees. At the end of the project we understood

communication between and inside teams is still a problem, it became better during the project but there is a lot to improve for the projects to come. We decided to make it better by starting a discussion whenever a potential problem is identified.

Balance the autonomy: to allow changes in the project to be executed with more agility it is necessary to give more autonomy to the teams. The challenge is to balance this autonomy with directions from the company that must be given and limitations to the project. Communicate the limitations that must be faced in the project to every person involved in the development process is very important.

Continuous planning is not lack of organization: there can be a restriction date from an external supplier, or too much work to do that you need to organize things, especially for big teams, in a schedule. In our opinion, having these kinds of restrictions do not mean the project cannot use continuous planning. Continuous planning refers more to the ability to adapt to the restrictions than to have or not have a schedule or a strong commitment to a certain date or delivery.

Planners will be planners: decentralizing the planning does not mean that a planning team is not needed. It is necessary to have people specialized in planning practices working closely or inside the development teams on a daily basis to help organizing, coordinating the work and identify events that influence the planning and help teams to adapt it, thus approximating planning and execution.

Evolve the process "on the fly": the use of agile practices needs to be flexible. Teams need to use the practices that best fit their needs, and when the plan changes, the practices also might need to change. In order to propose and execute the changes in the process teams need to have ownership of the process.

Timebox is not a requirement: planning and executing can have a fixed timebox or not. During the project frequency of planning meetings and sprint duration were changed, team B stopped dividing the work on regular intervals, therefore not having sprints anymore, and it worked well. So, once again, teams must use the practices that best fit their needs.

RQ2 – Does continuous planning help to improve agility?

When asked about agility, most people agreed that this new process brought more agility to the teams, but the concept of agility was perceived in more than one way, the different definitions were mostly regarding development velocity, agile methods and

decision making. For example, one participant said agility is “Not to use paperwork, the need to maybe go through the planning, generate an issue in Jira, to fix or develop something small that I can solve quickly.”, another participant said, “The agility that I see is the fast decision making.”, another one affirmed that it was “the use of agile practices of software development. Therefore, we can affirm that the perception of agility was improved, but there is no consensus about the definition of agility among participants.

RQ3 – How to support continuous planning adoption in software projects?

To answer this question COPLAM was created. The method goal is to support the continuous planning adoption on the team level of agile software development projects. COPLAM is explained in Chapter 3.

4.7. Limitations and threats to validity

Every study is subjected to threats to validity. We discuss here four types of threats, internal, external, construct and conclusion, and their influence in this study.

Internal Validity: individual wear was mostly identified in the first questionnaire since the part that presented questions about changes in the process was very long for one of the teams. Also, not everyone answered the questionnaire and some people reported to the researchers that it was too long. This was mitigated in the interview that had less questions and was semi-structured, allowing researchers and participants to focus on the questions listed but not limiting to that. Regarding environmental coincidences, the study took place right after a structural change in the organization, that divided a single IT team into three (two which participated in the study) and the change in the team’s organization and process might have affected participants’ perception regarding the project and continuous planning adoption as well.

Besides the threats already discussed, the fact that one researcher is a planning team member of the company, which brings the threat to participants tending to omit negative aspects of the process. This was mitigated by asking them to point out at least the worse thing they saw regarding the process and asking for improvements needs and opportunities. Also, the development team that worked more closely with the researcher was not considered for this study.

External Validity: This study was limited to only an organization and two of its teams, limiting its findings to this context and difficulties to generalize its results. As future work, we intend to execute more case studies in different teams and organizations.

Construct Validity: this regards the willingness and ability of the teams to adopt continuous planning. The organization wanted to change the way the planned projects. Given the constant change in the organization environment and very dynamic business context the organization and the project were suitable for the use continuous planning. One of the researchers works in the planning team of the company and could follow the continuous planning adoption guiding people to execute many activities from the COPLAM and proposing improvements in the process.

Conclusion Validity: the study was executed with a small number of participants and questions asked were interested in capture participants' perception, in this case no statistical analysis was executed, the purpose of the study was to explore continuous planning adoption providing insights to build a method for a structured way of adopting it, more studies are necessary in the future.

Another limitation is that the Continuous Planning Adoption Method (COPLAM) presented in the previous section was constructed during the study, with insights researchers had observing the organization and the study execution. This means the method was not yet evaluated, but many of the activities described were used during the study. Therefore, we believe there are indications that COPLAM can help continuous planning adoption in the project level of software projects, but its application in other projects or organizations is still needed.

4.8. Final Considerations

This chapter presented an action research study on continuous planning adoption comprehending one project and two development teams on a small sized organization in Rio de Janeiro. This study was conducted as part of a DSR cycle to create a method for continuous planning adoption in the team level. The resulting method was presented in Chapter 3.

Next chapter presents the evaluation of COPLAM in a case study and the improvements made on the method according to the results.

5. METHOD EVALUATION

This chapter presents the methodology and the case study executed to evaluate COPLAM.

5.1. Introduction

After the method construction, it was necessary to evaluate it. We decided to evaluate COPLAM in practice by executing a case study in the industry. A small sized organization was chosen for the case study execution and the method was adopted in the team level planning. This chapter presents the case study details and its findings.

5.2. Case Study Planning

The case study was executed to evaluate the use of COPLAM in practice using the Technology Acceptance Model (TAM) proposed by DAVIS et al. (1989). This model proposes to evaluate technologies regarding usefulness and ease of use. From the paradigm of GQM (BASILI et al., 1994), the goal of this study is defined as:

Analyze the method COPLAM

With the purpose of evaluating its applicability

Related to perceived usefulness and ease of use

From the point of view of the Planner

In the context of agile software development projects

According to DAVIS (1989) the definitions of perceived usefulness and perceived ease of use are as follows:

Perceived Usefulness: the degree to which a person believes that using a particular system would enhance his or her job performance. This follows from the definition of the word useful: “capable of being used advantageously.” Hence, a tool high in perceived usefulness is one for which a user believes in the existence of a positive use-performance relationship.

Perceived Ease of Use: refers to "the degree to which a person believes that using a particular system would be free of effort." This follows from the definition of "ease": "freedom from difficulty or great effort". A tool that is easy to use is more likely to be accepted by users.

Regarding Perceived Usefulness we defined one research question:

RQ1: Does COPLAM support Continuous Planning Adoption?

The goal is to evaluate if the method supports the continuous planning adoption in terms of allowing the definition of short and parallel cycles, the evolution of plans at any moment and more proximity between planning and execution.

Regarding Perceived Ease of Use we defined three research questions:

RQ2: Was the Planner able to execute the method independently, without external help, using only the method content (phases description, templates and documents)?

To evaluate the help needed during the execution of the method we analyzed how many times the Planner asked for help and the type of help that was needed. Also, we classified if the Planner asked about an information that is described in the method or not, if it was a simple or complex problem for the execution, the level of help required (simple, medium, complex) and if it was an obstacle for the execution continuity.

RQ3: Is the content of the method adequate?

This question goal is to evaluate if the information provided by the method is more than necessary, incomplete, ambiguous, difficult to understand, if there is wrong information, not clear or conflict.

RQ4: Is the content suitable?

The goal is to evaluate if the information is described in the appropriate moment and if there is any activity described outside its phase, activity associated with the wrong role, stakeholders not mentioned/described.

The research questions were answered based on questionnaires to the Planner and the information gathered by the researcher during the case study execution.

We used as a basis the questionnaires proposed by DAVIS (1989) and the Likert scale based in the questionnaires from LANUBILE et al. (2003), the scale is "fully agree", "strongly agree", "partially agree", "partially disagree", "strongly disagree" and "fully

disagree". A seven scale containing a neutral value was not used because according to LAITENBERGER and DREYER (1998) a neutral value does not give information about the direction the participant is inclined to (agreeing or disagreeing). Table 28 and Table 29 present the questionnaire for Perceived Usefulness, Perceived Ease of Use and Self-Predicted Future Usage respectively. We added a few open-ended fields in the questionnaires for comments on the answers with the purpose of allowing the Planner to explain the answers given.

Table 28. Questionnaire for evaluating Perceived Usefulness

Statement	**Answer**
1. Using COPLAM in my job would enable me to accomplish tasks more quickly.	( ) Fully agree ( ) Strongly agree ( ) Partially agree ( ) Partially disagree ( ) Strongly disagree ( ) Fully disagree
2. Using COPLAM would improve my job performance.	( ) Fully agree ( ) Strongly agree ( ) Partially agree ( ) Partially disagree ( ) Strongly disagree ( ) Fully disagree
3. Using COPLAM in my job would increase my productivity.	( ) Fully agree ( ) Strongly agree ( ) Partially agree ( ) Partially disagree ( ) Strongly disagree ( ) Fully disagree
4. Using COPLAM enhances my effectiveness on the job.	( ) Fully agree ( ) Strongly agree ( ) Partially agree ( ) Partially disagree ( ) Strongly disagree ( ) Fully disagree
5. Using COPLAM makes it easier to do my job.	( ) Fully agree ( ) Strongly agree ( ) Partially agree ( ) Partially disagree ( ) Strongly disagree ( ) Fully disagree
6. I would find COPLAM useful in my job.	( ) Fully agree ( ) Strongly agree ( ) Partially agree ( ) Partially disagree ( ) Strongly disagree ( ) Fully disagree
Comment on your previous answers:	

Table 29. Questionnaire for evaluating Perceived Ease of Use

Statement	**Answer**
1. Learning to operate COPLAM would be easy for me.	( ) Fully agree ( ) Strongly agree ( ) Partially agree ( ) Partially disagree ( ) Strongly disagree

Statement	Answer
	( ) Fully disagree
2. I would find it easy to get COPLAM to do what I want it to do.	( ) Fully agree ( ) Strongly agree ( ) Partially agree ( ) Partially disagree ( ) Strongly disagree ( ) Fully disagree
3. My interaction with COPLAM would be clear and understandable.	( ) Fully agree ( ) Strongly agree ( ) Partially agree ( ) Partially disagree ( ) Strongly disagree ( ) Fully disagree
4. I would find COPLAM to be flexible to interact with.	( ) Fully agree ( ) Strongly agree ( ) Partially agree ( ) Partially disagree ( ) Strongly disagree ( ) Fully disagree
5. It would be easy for me to become skillful at using COPLAM.	( ) Fully agree ( ) Strongly agree ( ) Partially agree ( ) Partially disagree ( ) Strongly disagree ( ) Fully disagree
6. Overall, I find the COPLAM method easy to use.	( ) Fully agree ( ) Strongly agree ( ) Partially agree ( ) Partially disagree ( ) Strongly disagree ( ) Fully disagree
Comment on your previous answers:	

Table 30. Questionnaire for Evaluating Self-Predicted Future Usage

Statement	Answer
1. Assuming COPLAM would be available on my job, I predict that I will use it on a regular basis in the future.	( ) Fully agree ( ) Strongly agree ( ) Partially agree ( ) Partially disagree ( ) Strongly disagree ( ) Fully disagree
Comment on your answer:	
2. I would prefer using COPLAM to the previous way of planning and executing projects that I used.	( ) Fully agree ( ) Strongly agree ( ) Partially agree ( ) Partially disagree ( ) Strongly disagree ( ) Fully disagree
Comment on your answer:	

5.3. Data Collection

This study was conducted by one researcher that worked in the organization. The Planner was a project manager that also has the role of Product Owner in the organization. During the execution of the method there were two data collections. First, during the execution of the phases Elicit Context, Define Planning Cycles, Execute Planning Cycles and Evaluate Planning the researcher collected doubts, problems, help needed and feedback from the Planner. Second, at the end of the method execution the researcher collected feedback about the method from the Planner using the Technology Acceptance Model.

For the data collections, the Planner was asked to sign a consent term, the term is presented in APPENDIX I.

5.4. Case Study Context

The case study was executed in the same organization as the action research study presented in Chapter 4, but in different project and team contexts. The organization is a small-sized company in Rio de Janeiro city and focused on e-commerce. After the end of the project explained in the action research study, the development teams were reorganized according to different business focus and planning were not a continuous process yet. We used COPLAM for continuous planning adoption in two teams. The study was carried out from July 2017 to August 2017.

The Planner was a project manager at the organization working as product owner in the two development teams. Teams were composed by front-end developers, back-end developers, designers and testers. One team was called Comparator and the other Market Place, each one focused on the respective business model of the company. Comparator was composed by three back-end developers, one front-end developer, one Android developer, one designer and one tester. Market Place team was composed by three back-end developers, one front-end developer, one designer and one tester. Not all team members were exclusive dedicated for the team, specially, tester and designer were the same person in both teams and one front-end and one back-end from Comparator also worked in other development teams not included in the case study.

5.5. Results

This section presents the results gathered during the data collection steps. As explained before three data collections were executed: during the execution of each phase the researcher collected doubts, problems, help needed and feedback from the Planner; at the phase Evaluate Planning feedback from team members was collected as a part of COPLAM execution; finally, at the end of the method execution the researcher collected feedback about the method from the Planner using TAM.

The study was executed with two development teams. At phase 1, Elicit Context, the Planner listed characteristics of the organization, characteristics of the projects being developed and future ones, characteristics of the teams, one previous planning problem related to one of the projects and several possible risks and restrictions. Because of confidential information contained in this data it is not possible to present it. After listing the data before, the Planner described the motivation for continuous planning and the analysis of the information gathered, that we present as follows:

Main motivation for adopting continuous planning: "Since we do not adopt a traditional methodology to projects' definition, and because of the characteristic of our business, we have some points in which continuous planning should help:

- Lack of clear and objective definition of scope. Usually we initiate the projects without that definition and opt to start working and prioritizing without many details.
- A lot of projects have the need for experiments with users, that can affect planning and create uncertainties because we do not have a lot of experience in this dynamic.
- The new business, the Market Place, also generates a lot of uncertainties because it is a new business inside the company. A lot of strategic and business decisions are difficult to be taken.
- There is lack of a better planning for evolution of the product that also introduces uncertainties and changes in the plans without much predictability.
- The interference of people outside the development teams also generates the need of revision or rework.

- The Objectives and Key Results (OKRs) [3] help us direct the planning, but we do not have clearly the success indicators of a project when it is first conceived nor the discipline to define and improve them during execution. Due to that, sometimes we do not know if an ended project was successful or not and we do not have a clear view of which indicator were impacted."

Analysis of the information gathered: "Many projects involve a new business for the company, the market place, which generates a lot of questions and lack of mastery and knowledge of the business. This generates a lot of uncertainties in this derationing. The choices of technical solutions made by the teams need a business base that do not come well defined.

The projects are born without scope, even it is not well detailed with formal project methodologies, they are not clear and create doubts even to analyze periodically where we are and if the goals have already been reached. The OKR metrics do not respond all the questions.

Strategic or portfolio planning would help us not to have an 'ideas" set so big and unstructured that come to be projects. It is important for us to define more widely where we want to arrive as a business, how we want to build this path. Just doing projects without this widely organization gives us a sense of debit because we have an ocean of possibilities and the lack of certainty of which paths to choose. Capacity is not infinite, so we must make good choices, understand strategy, but also always having the option to change.

But change with the understanding of impact and what we are going to gain or lose with the choices made along the way. Changing planning should be a natural thing, and simple to be done. It cannot generate insecurity and demotivation, it cannot be just because one person wants it. The team and all people involved need to understand that it makes sense, that it is an evolution, not a lack of planning or management over the business."

At phase 2, Define Planning Cycles, the Planner defined for both teams the same levels and periodicity of cycles. The items to be planned contained in the cycles

[3] Objectives and Key Results (OKRs) is a framework for defining goals and indicators to measure the results that will help to achieve these goals. A goal to be useful must mention both what you will achieve and how you are going to measure its achievement. This formula is the best way to explain the structure of an OKR: I will (Objective) as measured by (this set of Key Results) (CASTRO, 2016).

macroplan and the details of the standard and instantiated process cannot be present also for containing confidential information, but the remaining information are presented in Table 31.

Table 31. Planning Cycles Details

Planning Level	**Release**	**Iteration**
Granularity	Feature, Story	At iteration level, we define tasks in the format of issues, on Jira tool.
Periodicity of the cycle	Three months, aligned with the beginning and ending of OKRs definitions for the company that happens at each quarter.	Two weeks, starting at a Monday.
People involved	Product Owners of each team, stakeholders of each project/team, technical leaders.	Product Owner and all team members. There is no need for involving stakeholders, they can only have the knowledge of the iterations defined because as granularity is more detailed, what matters to them are the deliverables agreed for the release, which are not necessarily delivered at the end of each iteration.
Deliverable	Delivery in production environment and validated by the stakeholder.	Delivery to production environment Technical study about solution options

The phase Execute Planning Cycles has most of its information confidential, the items in the cycles microplan contain details about product functionalities and business rules that we cannot present in full details, we treated the data to present the most details possible regarding the events, decisions and actions taken. Table 32 presents the Events Registry for Market Place Team and Table 33 presents the Events Registry for Comparator Team.

For the team Market Place three events in the level of release happened, two of them were Long Resolution Events (LRE) events and one was Rapid Resolution Event (RRE). In the iteration level three events occurred, two were RRE and one was LRE.

For the team Comparator one event occurred in the level of release and it was a LRE event. Seven events occurred in the iteration level and they were all RRE events.

The case study took place during a complete iteration but not until the end of a release. Therefore, we only followed the first two weeks of the cycle of release level. From the events collected we noticed that in the release level most of the events were LRE and only one was RRE, this indicates that the higher the level of planning the longer the events to treat. In the iteration level only one event was LRE and all the others were RRE, indicating that the lower the level of planning, the faster the events treatment will be.

Table 32. Events Registry for Market Place Team

Date	Level of the cycle	Event	Type	Decision	Actions
02-08-2017	Release	Questions about prioritization were raised regarding a key indicator of the business that was not performing well and the release had no planned deliveries that would impact it at first.	LRE	Gather ideas of projects until the end of the current iteration cycle and review the release microplan if necessary.	The product owner will gather the ideas with the organization departments and review the release microplan if necessary.
03-08-2017	Release	One of the projects has a pending business definition that will not be defined until the date initially estimated for its delivery.	LRE	Inform the team about waiting for a decision for this project to go on or not until the end of the current iteration cycle. Other events can influence this decision because they can introduce new priorities.	The product owner will define the priority of this project with the stakeholders.
03-08-2017	Release	One of the projects will be moved to another team to work on because they were working on a similar project and can continue this work.	RRE	Redirect this project to the new team.	Present the current state to the stakeholders and plan the proposed changes.
04-08-2017	Iteration	A new functionality was solicited by the Commercial department as a fast reaction to one of the indicators of the business that was not performing very well.	RRE	Include the development of this new functionality in the current iteration cycle, even if it is hardcoded.	Maximum priority for this issue over any other in the iteration (it was delivered). Another issue that was originally planned had to be pushed to the next iteration cycle.
04-08-2017	Iteration	Due the previous event, new functionality had to be included in the iteration microplan and another had to be moved.	LRE	Remove issue from current iteration.	Include the issue for next iteration.
08-08-2017	Iteration	The results from the delivery of the functionality included in the iteration were very good. Due to this fact, an improvement to potentialize the results was solicited.	RRE	Develop the improvement before the end of the current iteration.	The development was successful but the delivery was not on the date planned, still is was delivered during the iteration. Other issues that were initially planned were impacted.
09-08-2017	Iteration	An error was identified and it was already available to users	RRE	A decision was made to prioritize this error's correction over other issues due to its impact on user experience.	Correct the error in the current iteration.
10-08-2017	Iteration	One of the developers had a personal problem and was absent for one and a half day.	RRE	Pass the most important issue being developed to another developer outside the team.	Ask for another developer to develop the issue. It was developed in time but due to technical problems the delivery was postponed.
14-08-2017	Iteration	An error was identified and it was already available to users	RRE	A decision was made to prioritize this error's correction over other issues due to its impact on user experience.	Correct the error in the current iteration.

Table 33. Events Registry for Comparator Team

Date	Level of the cycle	Event	Type	Decision	Actions
02-08-2017	Release	A project that is being developed by another team can have new functionalities that we can also develop for our projects.	LRE	Wait for an ongoing study regarding another functionality to decide which one to do first.	Review release microplan before the end of the current iteration.
03-08-2017	Iteration	During the validation of another project that was delivered a new user need was identified.	RRE	Prioritize this need to be treated in the current iteration so we can finish this project.	Develop the improvement necessary.
03-08-2017	Iteration	During the validation of another project that was delivered an error was identified.	RRE	Prioritize this need to be treated in the current iteration so we can finish this project.	Develop the correction in the current iteration.
03-08-2017	Iteration	During the validation of another project that was delivered an error was identified.	RRE	Prioritize this need to be treated in the current iteration so we can finish this project.	Develop the correction in the current iteration.
03-08-2017	Iteration	Front-end issues finished early than expected.	RRE	Prioritize issue from a new project to advance this project.	Issue starts on the iteration but will finish only in the next iteration.
07-08-2017	Iteration	One of the partner stores questioned the data from a report informing that was possible to be an error in our calculations.	RRE	Verify the information immediately.	Verify the report and analyze if there is any problem in the website.
09-08-2017	Iteration	New need for a communicative email to our partner stores.	RRE	Create email	Generate HTML for email.
09-08-2017	Iteration	New need for another communicative email to our partner stores.	RRE	Create email	Generate HTML for email.

From the events registered it is possible to observe that during the execution of the cycles, for iteration and release, changes in the scope occurred. New opportunities were identified, in some cases because of business indicators changing, in other cases changes in clients' needs, errors were identified after delivery, also need for change could be identified during validation of new functionality with stakeholders. The events had impact in the current iteration microplan, adding and excluding items, also identifying items that would be necessary to be executed in the next iteration. Events also introduced the need for reviewing the release microplan. Since we could not carry this study until the end of the release, we could not follow all impacts. But these results indicate the plans for both iteration and release were continuously affected and evolved according to the events.

At the end of both iteration cycles the cycle review took place. Table 34 shows the Planner answers for Market Place team and Table 35 the answers for Comparator team.

Table 34. Cycle Review for Iteration of Market Place Team

Cycle Execution: **Beginning date: 01/08/2017** **Ending date: 15/08/2017**	**Level of planning: ( ) Release (X) Iteration ( ) Day**	
Question	**Answer**	**Evaluation Theme**
Did the cycle microplan evolve according to events identified during the cycle's execution?	(X) Yes ( ) No ( ) No event occurred	If the answer is "No" the Planning and execution theme must be evaluated.
Were all of the identified events treated properly?	( ) Yes (X) No ( ) No event occurred	If the answer is "No" the Events identification and treatment theme must be evaluated.
If there was difference between the scope planned and the deliveries, was the difference related to the events registered?	(X) Yes ( ) No ( ) There was no difference	If the answer is "No" the Events identification and treatment theme must be evaluated.
Was the periodicity of the cycle adequate?	(X) Yes ( ) No	If the answer is "No" the Planning and execution theme must be evaluated.
Did any communication problem happened during the cycle's execution?	(X) Yes ( ) No	If the answer is "Yes" the Communication theme must be evaluated.
Does the current process provide ways to identifying and treating events properly?	(X) Yes () No	If the answer is "No" the Process Improvement theme must be evaluated.
Does the current process provide support for communicating events and changes in the plan?	( ) Yes (X) No	If the answer is "No" the Process Improvement theme must be evaluated.
Is there any problem in the execution of agile practices currently in use?	(X) Yes ( ) No	If the answer is "Yes" the Agile Practices theme must be evaluated.
Is there any agile practice not in use that should be used?	(X) Yes ( ) No	If the answer is "Yes" the Agile Practices theme must be evaluated.

Table 35. Cycle Review for Iteration of Comparator Team

Cycle Execution: **Beginning date: 01/08/2017** **Ending date: 15/08/2017**	**Level of planning: ( ) Release (X) Iteration ( ) Day**	
Question	**Answer**	**Evaluation Theme**
Did the cycle microplan evolve according to events identified during the cycle's execution?	(X) Yes ( ) No ( ) No event occurred	If the answer is "No" the Planning and execution theme must be evaluated.
Were all of the identified events treated properly?	(X) Yes ( ) No ( ) No event occurred	If the answer is "No" the Events identification and treatment theme must be evaluated.
If there was difference between the scope planned and the deliveries, was the difference related to the events registered?	(X) Yes ( ) No ( ) There was no difference	If the answer is "No" the Events identification and treatment theme must be evaluated.
Was the periodicity of the cycle adequate?	(X) Yes ( ) No	If the answer is "No" the Planning and execution theme must be evaluated.

Cycle Execution: **Beginning date: 01/08/2017** **Ending date: 15/08/2017**	**Level of planning: ( ) Release (X) Iteration ( ) Day**	
Question	**Answer**	**Evaluation Theme**
Did any communication problem happened during the cycle's execution?	(X) Yes ( ) No	If the answer is "Yes" the Communication theme must be evaluated.
Does the current process provide ways to identifying and treating events properly?	(X) Yes ( ) No	If the answer is "No" the Process Improvement theme must be evaluated.
Does the current process provide support for communicating events and changes in the plan?	( ) Yes (X) No	If the answer is "No" the Process Improvement theme must be evaluated.
Is there any problem in the execution of agile practices currently in use?	(X) Yes ( ) No	If the answer is "Yes" the Agile Practices theme must be evaluated.
Is there any agile practice not in use that should be used?	(X) Yes ( ) No	If the answer is "Yes" the Agile Practices theme must be evaluated.

Finally, the Evaluate Planning phase was executed, since there was an evaluation theme to be evaluated by both teams, the Planner constructed two questionnaires, one for each team. After gathering the answers from the teams, the Planner analyzed them and listed weaknesses, strengths and lessons learned for each Team and described actions need. Table 36 and Table 37 present these results.

For Market Place Team, the Planner identified 3 strengths, 3 weakness and 1 lesson learned.

Table 36. Weakness, strengths and lessons learned of Market Place Team

Classification	**Description**	**Type of Actions**	**Actions**
Strength	The proximity and the few people in the team made communication work well even not having team meetings daily or weekly.	Improvement in the project process	Promote daily meetings. Do not stop doing meetings at the beginning/ending of each iteration.
Weakness	The Market Place Team O MKP has a lot of projects and we need to pay attention on prioritization of release/iteration to avoid urgent demands. We also need to improve predictability to communicate it to stakeholders.	Improvement in the project process	Prioritize items for release level with in advance and aligned with the stakeholders and the team.
Strength	About communication with other teams, when it comes to the business Market Place, the Market Place team and other teams opine enough and interact.	Improvement in the project process	Stimulate more code review between teams.
Lesson Learned	The team had a good perception about the adaptations made in planning.	Improvement in the project process	Keep the register of the adaptations on Jira tool and make them clearer. Also, involve more the Team on decisions about the adaptations.
Strength	Having a team able to work end-to-end on its projects allowed a bigger focus on actions, this was perceived by the team members.	No action needed	We can try to give more visibility of the user experience work.
Weakness	We do not have integrated testing practices and the team complained about how much of testing is still manual, or automatized but not well managed.	Improvement in the project process	It is not an action for this team only. We need to evolve the automated tests and have a schedule for it.
Weakness	In a general way, the team do not give much opinion about agile practices, we do not know if it is because of lack of	Improvement in the project process	Expose more about practices and improvements in the process to engage the team more in the

Classification	Description	Type of Actions	Actions
	knowledge or if they think it will not be used.		evolution of the process. Retrospectives can be a good practice to help.

The lesson learned that originated the action “Keep the register of the adaptations on Jira tool and make them clearer. Also, involve more the Team on decisions about the adaptations” was first classified as “change in the planning levels”, after revision of the research the classification changed to “improvement in the project process”. This confusion happened because the action defined by the Planner was regarding the registering of events, the adaptations cited in her text, and since an event is related to a planning level, she thought it would be a change in the planning levels. But since the planning levels would stay the same, release and iteration, it was understood that it was a change in the process, because it would only affect the Events Registry.

For Comparator team, the Planner identified 3 strengths and 3 weaknesses.

Table 37. Weakness, strengths and lessons learned of Comparator Team

Classification	Description	Type of Actions	Actions
Strength	The team understands that the communication between them is good even not having a formal process of recurrent communication.	No action needed	
Weakness	Although we have examples where communication with stakeholders worked well, we have other issues as some people do not even know who are the stakeholders of the team.	Improvement in the project process	Make clearer who really are the stakeholders of the team. The Product Owner act to allow stakeholders to participate more in the communication with the team. One of the stakeholders is conducting a “Product Committee” that is important to communicate to the team. There is also a doubt about this committee process, because what is discussed there does not become priority immediately.
Weakness	Regarding communication with other teams, the problems are mainly on deliveries that one team does that impacts on the work of the other.	Improvement in the project process	Define a clear process of communication od deliveries of each team to stop generating problems with code commits.
Strength	Most people think the current process is better than the previous one.	Improvement in the project process	The main actions to improve the development process are currently in progress: Change the versioning tool and improve the deployment process. Another important item is the documentation of requirements, we still need to study solutions and evolve this point.
Strength	Most people think the agile practices identified (Code Review, Refactoring, Time end to end) are good.	No action needed	
Weakness	In a general way, the team do not give much opinion about agile	No action needed	Expose more about practices and improvements in the process to engage the

Classification	Description	Type of Actions	Actions
	practices, we do not know if it is because of lack of knowledge or if they think it will not be used.		team more in the evolution of the process. Retrospectives can be a good practice to help.

The actions defined were all improvements on the project process and included new agile practices to use, actions to improve communication and the idea of documenting the events using the Jira tool. Also, when analyzing the answers regarding agile practices, the Planner noticed that answers were very different inside the same team and was confused to what it could be done to have a more aligned vision from the hole team towards the agile practices. We did not consider it to be a problem from the method that generated this doubt, but a challenge that the organization is facing with the use of agile practices. The Planner defined for both teams the action to expose more information about practices and improvements in the process to engage the team more in the evolution of the process and considered retrospectives to be helpful in this matter.

5.5.1. Doubts and Problems collected during the method execution

During the execution of each phase, the researcher documented doubts and problems faced by the Planner and classified them according to their level of complexity. The researcher also registered if each one was and obstacle for the execution of the method (i.e. if the Planner had to stop the execution of COPLAM or if the continuity of the execution was not affected). The levels simple, medium and complex were used to classify the complexity of the explanation the researcher had to provide to the Planner at the time, this classification does not explain the amplitude of any change needed in the method. Changes in COPLAM regarding the results are explained further in the discussion session.

Each doubt/problem was given an identification in the format DP<phase number>.<sequential number>, for example, the 7th doubt/problem identified during the execution of the first phase of COPLAM has the identification DP1.7. The identification of each problem/doubt is used further to link it to improvements on the method.

During the execution of the phase Elicit Context, 10 doubts/problems were identified and among them 4 were obstacles to the continuity of the method execution and 6 were not. Seven of the doubts/problems were simple and required a rapid

explanation, 3 of them were complex and required a more elaborated discussion. Table 38 presents the doubts and problems and their classification.

Table 38. Doubts and Problems in Phase Elicit Context

Identification	Doubt or problem description	Did it affect the continuity of execution?	Level
DP1.1	What are team characteristics?	No	Simple
DP1.2	Which is the level of detail to describe risks?	Yes	Simple
DP1.3	Execute the Elicit Context phase separating the information for each team, but define a unique instantiated process for all of them in the Define Planning Cycles phase?	Yes	Complex
DP1.4	Level of planning was misunderstood for maturity, but when continuing to read the text it became clear.	No	Simple
DP1.5	The Planner suggested that the method description would have an overview of the method phases without the details and figures for each phase.	No	Simple
DP1.6	Continuous Planning characteristics are not clear if they are only an explanation or something to be done in the execution of the method.	No	Complex
DP1.7	Describing risks and restrictions separately is difficult, it is better to describe them in the same section.	No	Simple
DP1.8	The Planner forgot to do the sections main motivation for adopting continuous planning and analysis of the information gathered she was using a printed version of the method and the table was cut into the next page.	Yes	Simple
DP1.9	The Planner did not understand what was information analysis.	Yes	Complex
DP1.10	The Planner asked if had to detail which resource was shared between teams. The researcher explained to detail only the specialty that was shared and not the resources.	No	Simple

Also during the first phase of COPLAM some positive feedbacks were collected from the Planner:

- Elicit risks and restrictions is important for the projects planning and execution.
- Classify risks and restrictions in business or technical is helpful to recall things that are not usually thought about the projects.

When executing Define Planning Cycles phase the Planner had fewer doubts/problems. At total 3 doubts/problems were identified, two of them were obstacles to the continuity of execution of COPLAM. Regarding the level of complexity to treat them, 2 were simple to explain and 1 medium. Table 39 presents the doubts/problems identified during the execution of the second phase of COPLAM.

Table 39. Doubts and Problems in Phase Define Planning Cycles

Identification	Doubt or problem description	Obstacle to the continuity of execution?	Level
DP2.1	The Planner asked which were the deliverables of each activity of the phase.	Yes	Medium
DP2.2	In the Cycles Macroplan, the Planner did not understand that she needed to list the items, instead she first explained what was a	Yes	Simple

Identification	Doubt or problem description	Obstacle to the continuity of execution?	Level
	backlog and a list of the things to be done in the next cycle execution.		
DP2.3	Granularity and Deliverables were not informed accordingly, the researcher had to explain because the Planner described the activities to define what should be done and delivered for the cycles instead of the granularity of the items and the definition of deliverables.	No	Simple

During the execution of the cycles, third phase of COPLAM, only one doubts/problem was identified. It was not an obstacle for the continuity of the method execution and it required a simple explanation.

Table 40. Doubts and Problems in Phase Execute Planning Cycles

Identification	Doubt or problem description	Obstacle to the continuity of execution?	Level
DP3.1	The Planner asked (before reading the method description) what were events. For example, if someone asked a team member to participate in a meeting to help with technical doubts.	No	Simple

At the phase Evaluate Planning feedback from team members was collected, for this phase we translated the questions from English to Portuguese to facilitate the teams understanding since not all member were fluent in English. The translated questions are presented in APPENDIX II. During this phase execution, the doubts/problems that occurred were during the construction of the questionnaire considering the themes chosen in the previous phase.

Table 41. Doubts and Problems in Phase Evaluate Planning

Identification	Doubt or problem description	Obstacle to the continuity of execution?	Level
DP4.1	The Planner forgot to include questions regarding one of the themes in the questionnaire to one of the teams. The researcher had to interfere and point out that questions were missing.	No	Simple

Given the doubts/problems identified during the method execution, actions to improve COPLAM were taken. Table 42 explains the improvement actions for each doubt/problem. Not all doubts/problems originated improvement actions, some of them we considered inherent to organization's nature and its projects context.

Table 42. Improvement Actions for COPLAM

Doubt/ problem	Improvement Action(s)
DP1.1	Include examples of characteristics more explicitly.
DP1.2	Include examples of risks.
DP1.3	No action was needed, the project context analysis can englobe more than one team. In this case the doubt that emerged from the Planner was consequence of the context of the organization.
DP1.4	No action needed.
DP1.5	Include a figure with only the method phases and not the activities and artifacts to give a higher-level vision of the method before going into details.

Doubt/ problem	Improvement Action(s)
	Change colors of the phases to better vision in black and white printing. Include figures with the details of each phase in the beginning of each phase description.
DP1.6	Move the continuous planning characteristics to Chapter 2 to clarify that is the result of a literature review and not a part of the method execution.
DP1.7	Risks and restrictions became the same section and not separate ones.
DP1.8	Avoid dividing tables into more than one page. In some cases, it was not possible due to text structure and the size of the tables, so we explained better in the method description the information to be produced.
DP1.9	Explain better the information analysis.
DP1.10	No action needed.
DP2.1	Explicit each deliverable at the beginning of the phase description.
DP2.2	Review text to make it more explicit that the items planned should be listed.
DP2.3	Change the description to make clearer what a deliverable is and the difference of granularity of planning items and type of deliverables of a planning cycle.
DP3.1	No action needed.
DP4.1	Include a revision of the questionnaire as a step to check if all the evaluation themes were included.

5.5.2. Results from TAM Questionnaire

This section presents the Planner answers to TAM questions regarding perceived usefulness, perceived ease of use and self-predicted future use of COPLAM. The answers are presented in Table 43, Table 44 and Table 45.

Given that the method was executed by only one person, we have only one set of answers and due to this fact quantitative analysis is not possible. In this section, we discuss and explore the collected answers from a qualitative point of view.

Results of Perceived Usefulness: From the six objective questions addressing perceived usefulness, all of them were concentrated in the options “fully agree”, “strongly agree” and “partially agree”, indicating that the Planner was inclined to agreeing with the method usefulness. Table 43 presents the answers for each question. From the answers gathered we conclude that COPLAM is useful for planning in the team level. The Planner partially agreed to the questions regarding accomplish tasks more quickly and increase productivity, this gives a perception that the method helps with that but could be improved to be faster to execute.

Table 43. Answers for Perceived Usefulness

Statement	Answer given
1. Using COPLAM in my job would enable me to accomplish tasks more quickly.	Partially agree
2. Using COPLAM would improve my job performance.	Strongly agree
3. Using COPLAM in my job would increase my productivity.	Partially agree
4. Using COPLAM enhances my effectiveness on the job.	Strongly agree
5. Using COPLAM makes it easier to do my job.	Fully agree
6. I would find COPLAM useful in my job.	Fully agree

After answering the six questions the Planner was asked to comment on the answers given, the comments were "Overall the beginning generates more work, but having a method to organize planning that mainly focus on the changes we make during the way is very useful. The daily registering of events helps us understand the problems and treat them, because the method forces a moment of reflection."

From the researcher's perspective, the phases Elicit Context and Define Planning Cycles are indeed heavier to execute, mostly when the projects and teams' characteristics were not information previously documented by the company and when the default process for software development is not documented either, which was the scenario in this case study. Once the default process is documented it will require less effort to generate the instantiated process. Documenting the projects and teams' characteristics is important to understand the context, during the execution of the method we collected feedback that eliciting and classifying risks and restrictions is important. We believe that once it becomes a habit it can also be quicker to do.

Results of Perceived Ease of Use: From the six objective questions addressing perceived ease of use, all of them were concentrated in the options "fully agree" and "strongly agree", indicating that the Planner was inclined to agreeing that the method is easy to use. Table 44 presents the answers to each question. The questions about getting the method to do what the person wants, and method flexibility were answered with "fully agree". This gives evidence that COPLAM was easily adapted to the organization context. For questions about being easy to learn, being clear and understandable, being easy to become skillful using the method and the method being easy to use, the answers were "strongly agree". This indicates that the method is easy to use, but also relates to the answers about perceived usefulness by showing that there an opportunity for the method to become clearer and easier to use.

Table 44. Answers for Perceived Ease of Use

Statement	Answer given
1. Learning to operate COPLAM would be easy for me.	Strongly agree
2. I would find it easy to get COPLAM to do what I want it to do.	Fully agree
3. My interaction with COPLAM would be clear and understandable.	Strongly agree
4. I would find COPLAM to be flexible to interact with.	Fully agree
5. It would be easy for me to become skillful at using COPLAM.	Strongly agree
6. I would find COPLAM easy to use.	Strongly agree

The Planner was also asked to comment on his/her answers to perceived ease of use and the comments were "Overall the method is simple to be used, but it demands a little discipline. However, by being flexible on the way we plan it is good for adapting the method to our necessity. Also, the fact of demanding reviews of what was defined forces us to always improve and evolve. This makes it smoother to deploy."

These comments suggest that even with the opportunity that the method should be easier and clearer, its flexibility is good and matters in facilitating the use.

Results of Self-Predicted Future Usage: Both questions addressing self-predicted future usage were answered with "fully agree", indicating that COPLAM was well-accepted by the user, the Planner, and can continue to be applied in practice.

Table 45. Answers for Self-predicted Future Usage

Statement	Answer given
1. Assuming COPLAM would be available on my job, I predict that I will use it on a regular basis in the future.	Fully agree
2. I would prefer using COPLAM to the previous way of planning and executing projects that I used.	Fully agree

For each of the questions regarding self-predicted future usage the Planner was asked to comment on the answer. For the first question, the comments were "Yes, using some of the things from the method will be important in the future, but it is important to make it easier to understand to use it better.".

The researchers believe that this is linked to the answers given in the predicted usefulness questions regarding accomplish tasks more quickly and increase productivity because to use COPLAM in a regular basis the method needs to be faster to execute.

The comments to the second question were "Basically the method besides forcing us to reflect upon what was planned and what happened, the support to change by means of registering events makes all difference. The only problem is the discipline necessary

to use the method if we use it in spreadsheets. Think about a way of making it simpler and direct day-by-day using Jira would be interesting."

The researchers believe this corroborates the central point of COPLAM that is the events and the adaptation of plans according to them. The easier way of registering it can be with Jira tool but we did not focus on recommending a tool at first, neither Jira nor spreadsheets, to let it open to who is executing the method.

5.6. Findings

TAM questionnaire results were positive. No answer of disagreement was given in any of the questions. For perceived usefulness, the most important improvement need identified was related to the velocity of COPLAM execution, primarily on the first two phases. Regarding perceived ease of use we conclude that even with the improvement need of being easier and clearer on its description, the flexibility of the method was good and played an important part on its use. Finally, on the self-predicted future usage the results were also excellent, indicating that the method has future usage on a daily basis for the organization, but also linking with the previous results, it needs a clearer and easier description and improve velocity of execution. We believe that improving the description to facilitate the understanding of the method will also affect the velocity of execution because, given the doubts and problems presented during the execution, a lot of effort were related to the understanding of the method and not on its execution itself. As presented early, some actions to improve the description were already taken and we believe that future uses of COPLAM will be easier and faster.

At the beginning of the case study planning four research questions were defined, we explore each one as follows.

RQ1: Does COPLAM support Continuous Planning Adoption?

From the feedback collected and the results from the TAM questionnaires, we conclude that the Planner could define short and parallel cycles of planning, that plans evolved according to internal and external events. Also, the Planner pointed out the events registry and impacts on planning as a very important part of the method. She also pointed out that there was more proximity between planning and execution.

RQ2: Was the Planner able to execute the method independently, without external help, using only the method content (phases description, templates and documents)?

The Planner was not able to execute all the method without asking for help, but we identified that all the information needed was in the method description.

RQ3: Is the content of the method adequate?

This question relates to RQ2. We identified that when the Planner asked for help sometimes the problem was that the Planner had read the descriptions partially and asked for help before finishing and other times it was because she read more than one activity description before executing and forgot the information when the time for executing it came. For this matter, we considered the results and reviewed the method description to make information clearer and easier to understand for future usage.

RQ4: Is the content suitable?

None of the problems/doubts/feedbacks collected were related to any activity described outside its phase, activity associated with the wrong role, stakeholders not mentioned/described, so we conclude that the content of the method was suitable.

5.7. Limitations and Threats to Validity

This study was subjected to four types of threats: internal, external, construct and reliability (RUNESON et al., 2012). We here discuss these threats, their influence in this study and the actions taken to mitigate each one of them.

Internal Validity: first internal validity threat was that the Planner would not understand the method description, this was mitigated by having one of the researchers to collect Planner's doubts and problems and give explanations when needed. Second, due to project's schedule there was a time pressure that could impact in the quality of information gathered. The researcher reviewed the information produced by the Planner to check if it was understandable. Also, TAM questionnaires' answers did not have a neutral option that would not inform if the participant was more inclined to agree or disagree. Moreover, open-ended questions asking the participant to comment was intended to force the respond tend to reflect about the answers and provide more valuable feedback to researchers.

Besides the threats already discussed, the fact that one researcher is a planning team member of the company, which brings the threat to participants tending to omit negative aspects of the process. This was mitigated by having the Planner to be someone

higher than the researcher in the organization hierarchy, in this way not having pressure from a superior to execute and evaluate the method.

External Validity: This study was limited to only an organization and two of its teams, limiting its findings to this context. COPLAM description, templates and examples along with the results of this study facilitates the execution of the method in other cases.

Construct Validity: this regards the willingness and ability of the teams to adopt continuous planning. The organization had recently changed the configuration of the development teams and needed a new way for planning projects and fast response to changes. Given the constant change in the organization environment and very dynamic business context the organization. The projects and teams were suitable for the use continuous planning. The use of the method was proposed by the researcher but the need for changing the planning process in the given context was clear for the Planner before the start of the study.

Reliability Validity: the study was executed with only one Planner, but she was not familiarized with the method, providing a point of view from a first contact with the method and its execution. No statistical analysis was executed because there was no suitable sample for this type of analysis. The study used the Technology Acceptance Model (TAM), a method used in several studies in the literature, even with few participants, to evaluate perceived usefulness, perceived ease of use and self-predicted future usage and the results were positive for all three evaluations. The planner was the researcher's manager on the company, having a close relationship with the researcher could also affect the results. Also, the study was carried in the same organization as the previous action research study, but the Planner was not in charge of continuous planning adoption in the action research and teams' context and projects' context were different in this case study.

5.8. Final Considerations

The present chapter presented the planning, results, limitations and threats to validity of a case study that evaluated COPLAM using TAM. The results from perceived usefulness, perceived ease of use and self-predicted future usage are very positive. The Planner have chosen agreeing responses for all questions, differing only in the level of agreement (fully agree, strongly agree and partially agree). The improvement needs

identified were regarding the velocity of execution of the method, indicating that it would be better if execution was quicker, and the clarity of the description.

Actions taken to improve the method were executed based on the doubts/problems collected during the method execution and mainly address the problem of clarity in method description. We believe that it also addresses the velocity of execution because the doubts encountered slowed down the execution of the method and less doubts would have made the execution quicker. Also, the longest points of the execution are the phase Elicit Context and the activity Review Project Process. Elicit Context can be longer for organizations executing many projects at the same time and not used to document teams and projects characteristics and risks/restrictions. Review Project Process is also a heavy activity if the default process is not documented, but once it is, it will be smoother when executed again.

Overall, we have evidences that COPLAM was suitable for this case study execution and can be used in different contexts in the future. Next chapter presents our final considerations about this work, the contributions, limitations and further work.

6. FINAL CONSIDERATIONS

This chapter presents the final considerations, contributions of this work, the limitations and future works.

6.1. Final Considerations

At the start of this work five requirements and a research question were defined, now we explore each one and comment on the results found and their implications.

R1 – The method shall allow plans to evolve at any time given an internal or external event: COPLAM aimed at fulfilling this requirement by defining the event management subprocess as a part of any planning cycle execution. This was not only well executed in the case study but also was pointed out by the Planner as the main contribution of the method.

R2 – The method shall guide the definition of planning in short and parallel cycles according to the organization needs: during the case study execution, we observed that the Planner was able to define planning cycles for release and iteration for both teams in a satisfying manner.

R3 – The method shall support continuous planning in the levels of release, iteration and day: the activity of Define Planning Levels supports the choice of planning levels to be release, iteration and/or day. This activity is part of the Define Planning Cycles phase, which also supports the definition of periodicity of execution and macroplant for each level of planning chosen by the Planner.

R4 – The method shall foster the alignment between planning and execution of projects: as seen in the case study, events identification and treatment allowed plans to evolve when needed and the documentation of events helped visibility of their impact on projects, therefore, planning and execution alignment is fostered by the Event Management subprocess of COPLAM.

R5 – The method shall support the identification of strengths and weaknesses of the planning process and its improvement: the activity of Review Cycle and the phase Evaluate Planning of COPLAM support this requirement. As feedback, the Planner also

pointed out the reflection moments of the method as one of its strengths and said: “the fact of demanding reviews of what was defined forces us to always improve and evolve”.

Finally, this work aimed at answering the research question “How to support continuous planning adoption at the team level of agile software development?”, COPLAM was created to give this support and its application in industry was successful.

6.2. Contributions

This work scientific contributions are:

- The literature review and continuous planning characteristics identified.
- The Continuous Planning Adoption Method (COPLAM).
- An action research study on continuous planning adoption.
- A case study of COPLAM’s application in industry.

6.3. Limitations

COPLAM is limited to team level planning, which englobes release, iteration and day planning. Also, the method was created to support only software development planning in organizations using agile methods. The focus of the method does not cover traditional software development; however, it can be tailored to this context if the development process allows rapid and parallel cycles of planning and adjustments in plans during the development.

The method was used only in one organization and by one Planner, this limits our findings to this context but we believe that with the descriptions provided in this work and the improvements executed after the case study COPLAM can be applied in other contexts and organizations successfully as well.

6.4. Future Work

Some ideas for future works regarding COPLAM are:

- COPLAM was applied in only one organization and by one Planner. As further work we intent to apply it in other organizations and contexts, with different Planners.

- Also, the method is focused on team level planning but we believe it can be tailored to be used in different levels of planning as product, portfolio and strategic planning.
- As means to facilitate COPLAM execution, the information to be gathered and produced during each phase could be formatted inside a proper tool, the development of a tool or the study to find an existent tool to accommodate this information can also be done in the future. Jira is a candidate identified during the case study that can be used for this aim, especially for registering the events.
- Publications about continuous planning have increased recently, indicating that the topic is becoming more popular. With the increase of the literature on this topic, a systematic mapping or systematic literature review on continuous planning use for software development can be further executed to give a more structured vision of this field of research.
- The Planner is the main role of the method and can be overloaded with the work of executing the method and participating actively on the project, this role can be reviewed to analyze if any of its responsibilities can be distributed to another new role or to the Team.

BIBLIOGRAPHIC REFERENCES

Ameller, D., Farré, C., Franch, X., Valerio, D., and Cassarino, A., 2017, Towards Continuous Software Release Planning: *IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*,.

Ayed, H., Vanderose, B., and Habra, N., 2012, A Metamodel-Based Approach for Customizing and Assessing Agile Methods, *in* 2012 Eighth International Conference on the Quality of Information and Communications Technology, IEEE, p. 66–74.

Basili, V.R., Caldiera, G., and Rombach, H.D., 1994, Goal Question Metric Paradigm, Encyclopedia of Software Engineering, JJ Marciniak:

Baskerville, R.L., and Wood-Harper, A.T., 1996, A critical perspective on action research as a method for information systems research: *Journal of information Technology*, v. 11, no. 3, p. 235–246.

BOGSNES, B., 2008, Implementing Beyond Budgeting: Unlocking the Performance Potential: John Wiley & Sons, Ltd., Hoboken, New Jersey.

Brenner, M., and Nebel, B., 2009, Continual planning and acting in dynamic multiagent environments: *Autonomous Agents and Multi-Agent Systems*, v. 19, no. 3, p. 297–331.

Castro, F., 2016, OKR Secrets - Tips, examples and templates for Objetives and Key Results:

Cohn, M., 2006, Agile estimating and planning: *VTT Symposium (Valtion Teknillinen Tutkimuskeskus)*, , no. 241, p. 37–39.

Davis, F., 1989, Perceived usefulness, perceived ease of use, and user acceptance of information technology: *MIS Quarterly*, v. 13, no. 3, p. 319–340.

Davis, F.D., Bagozzi, R.P., and Warshaw, P.R., 1989, User Acceptance of Computer Technology : a Comparison of Two Theoretical Models *: *Management*, v. 35, no. 8, p. 982–1003.

Dingsøyr, T., Nerur, S., Balijepally, V., and Moe, N.B., 2012, A decade of agile methodologies: Towards explaining agile software development:

Fitzgerald, B., and Stol, K.J., 2015, Continuous software engineering: A roadmap and agenda: *Journal of Systems and Software*, v. 123, p. 176–189.

Fitzgerald, B., and Stol, K.-J., 2014, Continuous software engineering and beyond: trends and challenges, *in* Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering - RCoSE 2014, ACM Press, New York, New York, USA, p. 1–9.

de França, B.B.N., Simões, R. V., Silva, V., and Travassos, G.H., 2017, Escaping from the Time Box towards Continuous Planning, *in* Proceedings of the 3rd International Workshop on Rapid Continuous Software Engineering, p. 43–49.

Gregor, S., and Jones, D., 2007, The Anatomy of a Design Theory: v. 8, no. 2, p. 312–335.

Heikkilä, V.T., Paasivaara, M., Lassenius, C., and Engblom, C., 2013, Continuous Release Planning in a Large-Scale Scrum Development Organization at Ericsson: *Agile Processes in Software Engineering and Extreme Programming*,, p. 195–209.

Hevner, A.R., 2007, A Three Cycle View of Design Science Research: *Scandinavian Journal of Information Systems*, v. 19, no. 2, p. 87–92.

Hevner, A.R., March, S.T., Park, J., and Ram, S., 2004, Design Science in Information Systems Research: *MIS Quarterly*, v. 28, no. 1, p. 75–105.

Hope, J., and Fraser, R., 2003, Beyond budgeting: how managers can break free from the annual performance trap: Harvard Business Press.

Knight, S., Rabideau, G., Chien, S., Engelhardt, B., and Sherwood, R., 2001, Casper: space exploration through continuous planning: *IEEE Intelligent Systems*, v. 16, no. 5, p. 70–75.

Kurapati, N., Manyam, V.S.C., and Petersen, K., 2012, Agile Software Development Practice Adoption Survey, *in* Wohlin, C. ed., Agile Processes in Software Engineering and Extreme Programming: 13th International Conference, XP 2012, Malm{ö}, Sweden, May 21-25, 2012. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, p. 16–30.

Laitenberger, O., and Dreyer, H.M., 1998, Evaluating the usefulness and the ease of use of a Web-based inspection data collection tool: *Proceedings Fifth International Software Metrics Symposium. Metrics (Cat. No.98TB100262)*,, p. 1–13.

Lanubile, F., Mallardo, T., and Calefato, F., 2003, Tool support for geographically dispersed inspection teams: *Software Process Improvement and Practice*, v. 8, no. 4, p. 217–231.

Leffingwell, D., 2011, Agile software requirements: lean requirements practices for teams, programs, and the enterprise.: Pearson Education, Inc.

Lehtola, L., Kauppinen, M., and Vahaniitty, J., 2007, Strengthening the link between business decisions and RE: Long-term product planning in software product companies, *in* Requirements Engineering Conference, 2007. RE'07. 15th IEEE International, IEEE, p. 153–162.

Lehtola, L., Kauppinen, M., Vähäniitty, J., and Komssi, M., 2009, Linking business and requirements engineering: is solution planning a missing activity in software product companies? *Requirements Engineering*, v. 14, no. 2, p. 113–128.

Mishra, D., and Mishra, A., 2011, Complex software project development: agile methods adoption: *Journal of Software Maintenance and Evolution: Research and Practice*, v. 23, no. 8, p. 549–564.

Myers, K.L., 1999, CPEF: A Continuous Planning and Execution Framework: *AI Magazine*, v. 20, no. 4, p. 63.

OHNO, T., 1988, Toyota production system: beyond large-scale production: crc Press.

Olsson, H.H., Bosch, J., and Alahyari, H., 2013, Towards R&D as Innovation Experiment Systems: A Framework for Moving Beyond Agile Software Development, *in* Artificial Intelligence and Applications / 794: Modelling, Identification and Control / 795: Parallel and Distributed Computing and Networks / 796: Software Engineering / 792: Web-based Education, ACTAPRESS, Calgary,AB,Canada.

Papatheocharous, E., and Andreou, A.S., 2014, Empirical evidence and state of practice of software agile teams: *Journal of Software: Evolution and Process*, v. 26, no. 9, p. 855–866.

Petersen, K., and Wohlin, C., 2010, The effect of moving from a plan-driven to an incremental software development approach with agile practices: An industrial case study: *Empirical Software Engineering*, v. 15, no. 6, p. 654–693.

Rickards, R.C., and Ritsert, R., 2012, Rediscovering Rolling Planning: Controller's

Roadmap for Implementing Rolling Instruments in SMEs: *Procedia Economics and Finance*, v. 2, no. Af, p. 135–144.

Runeson, P., Host, M., Rainer, A., and Regnell, B., 2012, Case Study Research in Software Engineering:

Shalloway, A., Beaver, G., and Trott, J., 2009, Lean-Agile Software Development: Achieving Enterprise Agility: Pearson Education.

Shull, F., Singer, J., and Sjøberg, D.I.K., 2008, Guide to Advanced Empirical Software Engineering:

Špundak, M., 2014, Mixed agile/traditional project management methodology–reality or illusion? *Procedia-Social and Behavioral Sciences*, v. 119, p. 939–948.

Suomalainen, T., Kuusela, R., Teppola, S., and Huomo, and T., 2015, Challenges of ICT Companies in Lean Transformation: *Advances in Computer Science: an International Journal*, v. 4, no. 2, p. 49–56.

Suomalainen, T., Kuusela, R., and Tihinen, M., 2015, Continuous planning: An important aspect of agile and lean development: *International Journal of Agile Systems and Management*, v. 8, no. 2, p. 132–162.

Sutherland, J., 2010, Scrum handbook: *Scrum Training Institute*, , no. May, p. 464.

van de Weerd, I., Bekkers, W., and Brinkkemper, S., 2010, Developing a Maturity Matrix for Software Product Management, *in* Springer, Berlin, Heidelberg, p. 76–89.

WOMACK, J., and JONES, D.T., 2003, Lean Thinking: Banish Waste and Create Wealth in Your Corporation: Productivity Press.

APPENDIX I

Consent Terms for Action Research and Case Study Execution

This appendix presents the Consent Terms required for the execution of the action research and case study on COPLAM's construction and evaluation. The terms are written in Portuguese because the studies were executed inside a Brazilian organization and the native language of the participants were Portuguese.

Consent Term for the Action Research study:

Universidade Federal do Estado do Rio de Janeiro (UNIRIO)

Programa de Pós-Graduação em Informática

Planejamento Contínuo no Desenvolvimento de Software Ágil

Estudo de Caso

Termo de Consentimento

Você está sendo convidado(a) a participar da pesquisa **"Planejamento Contínuo no Desenvolvimento de Software Ágil"**. Você foi selecionado por conveniência e sua participação não é obrigatória. A qualquer momento você pode desistir de participar e retirar seu consentimento. Sua recusa não trará nenhum prejuízo em sua relação com os pesquisadores ou com a instituição.

O **objetivo** da entrevista é capturar sua percepção sobre o planejamento e a execução de projetos, pontos positivos, pontos negativos, desafios e oportunidades melhoria.

Se concordar em participar deste estudo você será solicitado a responder questões sobre os projetos em que participou, a organização desenvolvedora de software, o planejamento e a execução dos projetos, pontos positivos e negativos, desafios encontrados e oportunidades de melhoria.

Destaca-se que **o nome do participante e da organização serão estritamente confidenciais**, sendo omitidos em que qualquer trabalho que venha a ser publicado. Entretanto, as demais informações serão utilizadas e publicadas em trabalhos científicos.

É importante que você esteja consciente de que a participação neste estudo de pesquisa **é completamente voluntária** e de que você pode recusar-se a participar ou sair do estudo a qualquer momento sem penalidades. Em caso de você decidir retirar-se do estudo, deverá notificar ao pesquisador que esteja realizando a entrevista. A recusa em participar ou a saída do estudo não trará nenhum prejuízo com esta instituição.

Declaro que li as informações contidas neste documento antes de assinar este termo de consentimento. Declaro que tive tempo suficiente para ler e entender as informações acima. Confirmo também que recebi uma cópia deste formulário de consentimento. **Dou meu consentimento de livre e espontânea vontade** e sem reservas para participar como entrevistado deste estudo.

Nome do Participante: _____

Data: _____/_____/_____

Assinatura: _____

Nome do Pesquisador: _____

Data: _____/_____/_____

Assinatura: _____

Consent Term for the Case Study:

Universidade Federal do Estado do Rio de Janeiro (UNIRIO)

Programa de Pós-Graduação em Informática

Aplicação do método para adoção de planejamento contínuo COPLAM - Estudo de Caso

Termo de Consentimento

Você está sendo convidado(a) a participar da pesquisa **"Aplicação do método para adoção de planejamento contínuo COPLAM"**. Você foi selecionado por conveniência e sua participação não é obrigatória. A qualquer momento você pode desistir de participar e retirar seu consentimento. Sua recusa não trará nenhum prejuízo em sua relação com os pesquisadores ou com a instituição.

O **objetivo** do estudo é capturar sua percepção sobre a utilidade, facilidade de uso e possibilidade de uso futuro do COPLAM bem como identificar oportunidades de melhoria no método.

Se concordar em participar deste estudo você será solicitado a utilizar o COPLAM em projetos e times que atua, fornecendo informações sobre o planejamento e a execução dos projetos, características dos times e da organização em que trabalha. Será necessário também responder questões sobre sua percepção de utilidade, facilidade de uso e possibilidade de uso futuro do método.

Destaca-se que **o nome do participante, da organização e detalhes sobre funcionalidades desenvolvidas nos projetos serão estritamente confidenciais**, sendo omitidos em que qualquer trabalho que venha a ser publicado. Entretanto, as demais informações serão utilizadas e publicadas em trabalhos científicos.

É importante que você esteja consciente de que a participação neste estudo de pesquisa **é completamente voluntária** e de que você pode recusar-se a participar ou sair do estudo a qualquer momento sem penalidades. Em caso de você decidir retirar-se do estudo, deverá notificar ao pesquisador que esteja realizando a entrevista. A recusa em participar ou a saída do estudo não trará nenhum prejuízo com esta instituição.

Declaro que li as informações contidas neste documento antes de assinar este termo de consentimento. Declaro que tive tempo suficiente para ler e entender as informações acima. Confirmo também que recebi uma cópia deste formulário de consentimento. **Dou meu consentimento de livre e espontânea vontade** e sem reservas para participar como entrevistado deste estudo.

Nome do Participante: _____

Data: _____/_____/_____

Assinatura: _____

Nome do Pesquisador: _____

Data: _____/_____/_____

Assinatura: _____

APPENDIX II

Translated Questions of the Evaluate Planning Phase

This appendix presents the questions from the Evaluate Planning Phase translated to Portuguese. This translation was used during the data collection of the case study for COPLAM evaluation. The translated questions for each evaluation theme are presented in tables Table 46 to Table 50.

Table 46. Translated Questions for Communication Theme

Pergunta	**Tipo de resposta**
Como você classifica a qualidade da comunicação entre o seu time e outros times?	( ) Insuficiente ( ) Regular ( ) Boa ( ) Excelente
Você sugere alguma melhoria na comunicação entre times?	Texto Livre
Como você classifica a qualidade da comunicação entre o seu time e stakeholders de outros departamentos?	( ) Insuficiente ( ) Regular ( ) Boa ( ) Excelente
Você sugere alguma melhoria na comunicação entre o seu time e os stakeholders de outros departamentos?	Texto Livre
Como você classifica a comunicação dentro do seu time?	( ) Insuficiente ( ) Regular ( ) Boa ( ) Excelente
Você sugere alguma melhoria na comunicação dentro do seu time?	Texto Livre

Table 47. Translated Questions for Events Identification and Treatment Theme

Pergunta	**Tipo de resposta**
Na sua opinião, o planejamento foi adaptado de acordo com eventos internos e externos? Exemplos:<O Executor do método deve listar aqui exemplos de eventos que aconteceram durante a execução do ciclo>	( ) Nunca ( ) Às vezes ( ) Na maioria das vezes ( ) Sempre
Você sugere alguma melhoria ao tratar eventos durante a execução do que foi planejado?	Texto Livre

Table 48. Translated Questions for Planning and Execution Theme

Pergunta	**Tipo de resposta**
Na sua opinião, a participação de <O Executor do Método deve dar exemplos de especialistas de planejamento como gerente de projetos ou *product owner*> ajudou em que?	Texto Livre

Pergunta	Tipo de resposta
Na sua opinião, como os <O Executor do Método deve dar exemplos de especialistas de planejamento como gerente de projetos ou *product owner*> podem ajudar a melhorar o planejamento?	Texto Livre
Na sua opinião, como é o alinhamento entre o planejamento e a execução dos projetos?	( ) Insuficiente ( ) Regular ( ) Boa ( ) Excelente
Na sua opinião, o planejamento e a execução tornaram-se mais próximos recentemente?	( ) Sim ( ) Não
Como você classifica a frequência dos ciclos de planejamento, ou seja, a frequência atual em que as entregas são planejadas?	( ) Insuficiente ( ) Regular ( ) Boa ( ) Excelente
Na sua opinião, alguma mudança deveria ser feita em relação ao planejamento de projetos? Se sim, qual(is)?	Texto Livre

Table 49. Translated Questions for Process Improvement Theme

Pergunta	Tipo de resposta
Qual parte do processo de desenvolvimento atual você acredita que é a melhor?	Texto Livre
Qual parte é a pior? Se você fosse mudar algo, o que seria a primeira coisa que mudaria?	Texto Livre
Comparado com o processo de desenvolvimento usado anteriormente, quais são as vantagens e desvantagens do atual?	Texto Livre
Na sua opinião, alguma melhoria deveria ser feita no processo? Se sim, qual(is)?	Texto Livre

Table 50. Translated Questions for Agile Practices Theme

Pergunta	Tipo de resposta
Para cada pratica ágil utilizada pelo time faça a seguinte pergunta: Como você classifica o uso de <nome da prática>?	( ) Insuficiente ( ) Regular ( ) Boa ( ) Excelente