Escaping from the Time Box towards Continuous Planning

An Industrial Experience

Breno Bernard Nicolau de França, Rachel Vital Simões, Valéria Silva, Guilherme Horta Travassos
PESC - COPPE
Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brazil
{bfranca, rachelsimoes, valeria, ght}@cos.ufrj.br

***Abstract*—Continuous Planning (CP) is a management practice inspired in principles from the agile and lean software development. In this paper, we present in detail the case of a small Brazilian software company on moving from an agile background towards CP, due to a constant changing environment. Furthermore, we discuss the reported case in the light of the technical literature on CP. Although observing evidence on benefits and challenges, the CP adoption at all levels using a "big-bang" approach may be disastrous. Therefore, we advocate the need for more systematic studies on CP, despite the complexity of observing it into real case environments.**

Keywords - continuous planning; continuous software engineering; agile project management.

I. INTRODUCTION

As a response to the increasing pressure to reduce the time-to-market, software organizations have been adopting different agile development practices to promote faster software deliveries in a reliable way. Regarding the impact on downstream activities, the acceleration of software releases highlighted bottlenecks in the systems operations and revealed the need for additional practices to handle and make feasible the constant and rapid software releases, which have been widely recognized as DevOps [1]. However, continuous software development and operations also require upstream management practices to deal with the development pace.

In this sense, Continuous Planning (CP) has been suggested as a managerial and strategic practice to plan and monitor the software development and operations progress from a holistic perspective. CP involves multiple stakeholders from different roles that should keep an extensive collaboration to foster a tighter integration between planning and execution. This way, leaders/managers would be able to continuously adapt their plans (represented by open and dynamic artifacts) along the product progress and business environment changes [2].

However, the technical literature on CP lacks empirical evidence on its applicability. Moreover, the few existing reports do not offer enough information and details on how CP is performed and could contribute to the development and evolution of software. We were not able to identify any technical work describing how CP could be achieved in the software organizations. Besides, the benefits and challenges faced when adopting CP are not exposed as well.

This paper takes these issues into account and contributes to the detailed description of how OWSE, a Brazilian software company, has adopted CP. It discusses the concepts and current practices, as well as its achievements. Besides, we offer a comparison between the case in OWSE and other three cases reported in [3] intending to strength evidence on CP.

Next sections are organized as follows. Section II presents the conceptual background. Section III presents how we collected information from the OWSE Company regarding the adoption of CP as well as the criteria for comparing the presented case against the ones in the technical literature. Section IV details the case at OWSE, including its motivation for adopting continuous planning, strategy, and daily management practices. Section V presents how the current state at OWSE compares to three other cases. Finally, Section VI presents the final remarks.

II. CONCEPTUAL BACKGROUND

A. Agile Project Management

Management processes considering agility have changed the way software projects can be managed and executed. Part of these changes regards an unavoidable changing environment, the increasing need for continuous innovation, and cost savings [4] [5]. Therefore, approaches offering adaptability to changes during the project lifecycle are considered more important than predictability [5].

Planning for product development can be organized into several levels [6]: daily, iteration, release, product, portfolio, and strategical. In general, agile teams use to plan only at daily, iteration, and release levels, while the other levels tend to be annually [2].

Among the existing management agile methods, Scrum is one of the most adopted in the industry [7]. It evolves on

978-1-5386-0428-1/17 $31.00 © 2017 IEEE
DOI 10.1109/RCoSE.2017.5

short iterations (sprints) with fixed size, usually lasting between two and four weeks. Such characteristic is the so-called *time-box,* i.e., every sprint last the same period.

B. Continuous Planning

Fitzgerald and Stol present the concept of Continuous Software Engineering (CSE), organizing its practices into three main areas: (1) business strategy and planning, (2) development, and (3) operations [2]. Continuous planning and budget are practices associated with the first area. In continuous software development, as in agile project management, plans are dynamic open-ended artifacts and evolve in response to the business environment. One of the main goals of the CSE regards the close collaboration of business and development areas, involving multiple stakeholders and closing the existing gap [2] [8].

In CSE, the notion of continuity comes from activities performed in very short and constant cycles, in which the flow of a given software feature is conducted end-to-end, i.e., from its conception until its availability to use. Thus, several cycles are executed in parallel and associated deliveries are released every time the cycles finish for a given demand.

The continuous environment favors, even more, the constant feedback regarding the software product, making easier the identification of changing opportunities and new features that may add more value to business. However, obtaining feedback is not limited to the iteration end (time-boxed), but it is captured as soon as an independent cycle finishes, or a feature is ready to be released.

Additionally, CP performs planning practices in rapid parallel cycles instead of predefined and regular planning occasions, reacting to changes and triggering planning when needed. For that, the planning level (items to be planned) should go beyond iteration and release planning, considering also higher-level planning such as product, portfolio, and strategy as well as their relationships. The timeframe (periods) of a plan may vary from hours to months according to the planning level.

As far as we are aware, there is no standard method or approach for CP. However, it does have fundamental elements [3]: organizational, strategic and business planning. Organizational planning defines the levels and timeframes of a plan, strategic planning forms the overall plan of an organization and business planning forms the budgeting frame for a plan. Besides, the CP factors include governance, leadership, transparency and competency development.

III. RESEARCH METHOD

This paper reports an experience at a Brazilian software company originally based on agile practices that started the transition towards CP in a development environment. It is not a report of a systematic observation like a Case Study or Action-Research. Rather, it represents the perspective from the involved people from organization along with the critical point of views of researchers.

Our main goal is to present the motivating issues, practices, observed benefits and challenges towards CP. The level of detail of this report differs from the one observed in the literature [3], aiming at improving the understanding of the adoption of CP from a contextualized perspective.

Along with the experience report, we searched for other experiences or case studies approaching CP in software projects, aiming at comparing similarities and differences.

Therefore, the research method included the following steps: (1) Reporting the experience at OWSE; (2) Identifying other experience reports or case studies exploring continuous planning in the technical literature; (3) Defining and extracting relevant information from the identified works (step 2) to support comparison; (4) Comparing OWSE case with the works of the technical literature (step 3).

Three researchers independently performed the steps one and two. The information for the report (step 1) was collected from different OWSE employees, but mainly from the project manager.

For the identification of relevant studies in the technical literature (step 2) we performed and structured the search in the Scopus digital library using the following search string: *("software development" AND "continuous planning" AND ("case study" OR "experience report"))*

We identified only one work [3] discussing continuous planning in three real settings. It reinforces the contribution of this paper (w.r.t. the lack of works) discussing CP issues.

For the step 3, we considered a set of information described in the three cases presented in [3]: Organizational Information (size, domain, provided services/products, software development process) and Continuous Planning (motivations and observed challenges and benefits).

Finally, we performed a comparison among our case and those presented in [3] aiming at observing potential similarities and differences regarding motivation, challenges, and benefits so that we could draw insights from these experiences.

IV. THE OWSE CASE

OWSE is a Brazilian software development company with 40 employees, being 70% software developers. Usually, project teams are composed of three to eight people; varying according to the perceived project complexity, the number of deliveries, and/or the project relevancy. Team roles include at least one system analyst, one or more developers, and one test analyst. Besides, software architects, project managers, and business consultants are shared among projects.

For three years, the company adopted Scrum to support development. However, some features of this method needed to be tailored according to the company and partners' needs.

Before adopting Scrum practices, the organization used to work *ad-hoc*. The team did not define tasks, and they had no planning meetings. Besides, top management missed

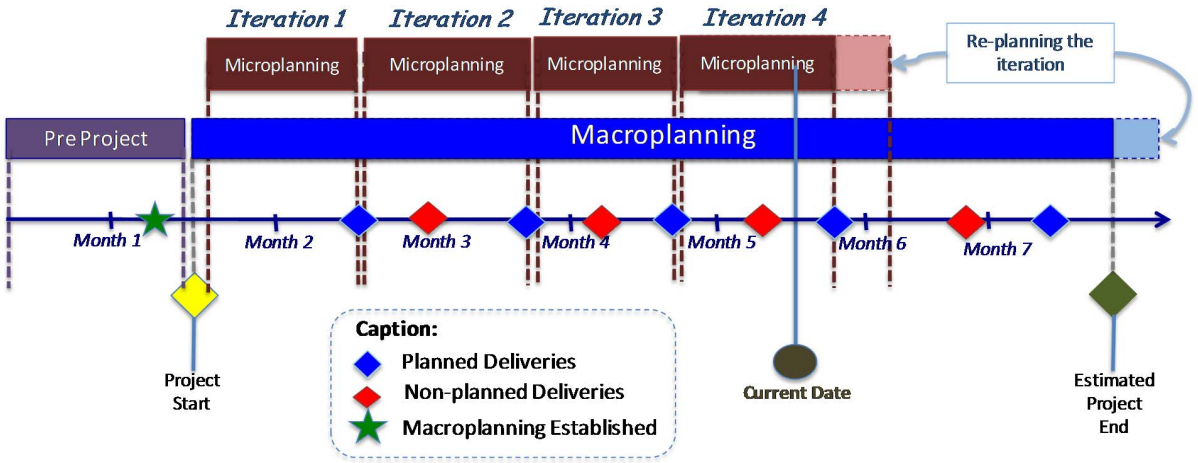

Figure 1: Planning iterations along the project lifecycle

relevant information to monitor a project progress as a whole. This way, the organization used to lack a holistic view of the projects, hindering the portfolio management.

The company also provides maintenance services for legacy systems. In both cases, clients demand a long-term planning. Thus, the company reports to stakeholders every fortnight. Critical projects require progress reporting weekly.

A. Motivation for adopting Continuous Planning

The main reasons for leading OWSE to invest in CP include the constant changing business environment, having the main client in logistics, as well as providing services for large companies. Such dynamic context requires plans not being updated only at discrete phases, i.e., occasionally. They need to evolve continuously to avoid making late and uninformed decisions.

However, clients ask for a long-term plan for the provided services. Furthermore, managers have difficulties in justifying the personnel allocation without a long-term planning. Cost and size of teams, including hiring, should match the project needs. Finally, the achievement of project goals could only be assessed through deliveries.

B. Project Management Lifecycle

Based on its needs, OWSE developed an initial model to shorten the frequency of planning activities, making it closer to CP. In this model, planning activities are split into two phases (Figure 1): Pre-project and Project.

The Pre-project planning aims at defining a holistic view, specifying just the main deliveries. Changes may occur during the development process and are accommodated during the project. Activities in this stage involve:

- Defining the initial product backlog;
- Estimating the product backlog size;
- Prioritizing backlog items with stakeholders;
- Setting the main milestones;
- Performing the initial release planning, when the main releases are defined, and;
- Developing an initial project schedule.

These activities compose the Macroplanning perspective. It is superficially performed based on a general understanding of the project, obtained at meetings with the main stakeholders. In this stage, a team of experts performs the estimations using Delphi technique [9]. Uncertainties in the initial planning are managed and plans updated accordingly. A first project baseline is defined at the end of the Pre-project.

In the Project phase, the management effort focus on planning the iteration scope with the project team, monitoring the progress of tasks, and ensuring the release planning follows the macroplanning. Besides the Scrum activities, the company included additional monitoring activities, such as risk management and the use of performance indicators, which are performed continuously.

C. Planning Perspectives

Two planning perspectives (Figure 1) composes the project management: Macro and Microplanning. Activities in both perspectives last the whole project lifecycle and work coordinated.

In the Macroplanning, the entire project progress is managed in a holistic view; while microplanning considers tasks inside the scope of a planning iteration. Daily updates in the microplanning should be reflected in the macro planning. Updates in the macroplanning are used to evaluate potential impacts on delivery dates.

1) The Concept of Planning Iteration

Although aiming CP in full, the company kept the pace of planning activities inside **planning iterations** under the following conditions:

- The upfront and detailed planning is understood as worthless due to inevitable and constant changes, so the detailed planning is performed only into iterations;
- There is a synchronism between planning and development. However, it does not mean the development outcomes should be delivered

exclusively when the iteration ends. They can occur continuously;

- Planning iterations are not time-boxed. The company experienced shorter and longer planning iterations, but they used to be more efficient when lasting about two weeks. However, they are free to vary according to the project needs and how the team works;
- At least one deliverable associated with the end of each iteration is expected, and;
- Deliveries may occur during the planning iteration. They may and should take place always the work is ready to be released and required to answer business needs.

Besides the use of iterations in the microplanning, some continuous features are preserved like the team updating the task progress daily, from which managerial information can be obtained at any moment; daily standup meetings last 15 minutes when tasks are monitored, and any hindrances are raised, and; any new task defined as important is included in the iteration scope and the project plan as well.

According to Figure 1, macroplanning represents the project overview, while the detailed planning occurs only at the beginning of each planning iteration, as suggested by the detailed planning until the fourth iteration, which represents the current planning iteration.

Re-planning may occur at any moment during the iteration. However, its impact on the macro planning must always be evaluated.

2) Macroplanning

In the macro planning, the items are defined in the product backlog. An initial effort estimation is made (in H/h), and the backlog items are prioritized based on their business value. From this, an initial schedule is generated and approved, and the baseline is evolved.

As the iterations run, task progress information updates the schedule, providing feedback for project managers and top management. The constant schedule updates allow the tracking of the project through the following indicators:

i. Percent of actual progress;
ii. Percent of progress according to the baseline;
iii. Project Performance Index, calculated by the ratio between the percent of actual progress and of progress according to the baseline.

Using tracking information, OWSE developed a board for monitoring projects (Table 1). Project and top managers use it as well as clients.

Semaphores inform the project situation by indicating project delay (in red); project under acceptable delay (in yellow), defined according to the project criticality and by the project sponsor, and; project progress is according to the plan (in green). As the performance index gets closer to 1, it means the project is running according to the plan. If it is equal or greater than 1, it means a favorable situation. Otherwise, it means the performance is below the expected, as indicated by the arrows beside the index values.

3) Microplanning

At the beginning of each iteration, a planning meeting is performed to select the backlog items according to their priorities. Usually, before the meeting, the items are sent back to the stakeholders so they can review the priorities. Then, items to be developed are decomposed into tasks, and their efforts are estimated by the team so that the iteration plan can be defined. The project manager is responsible for verifying the task planning against the macroplanning. Based on this, stakeholders' expectations are aligned with the updated planning information.

D. Challenges and Benefits

The observation of release planning allowed perceiving the unfeasibility of time-based iterations, as deliveries are more important than following the planned dates. This way, first initiatives towards CP started by the need for development activities that could be performed into flexible iterations, enough to answer to business needs, or even in a continuous flow.

Differences between macro and microplanning are considered to be normal, as long as they can be measured and evaluated. Main causes are:

- Lack of understanding of the software requirements at the Pre-Project Phase;
- Scope changes by new or existing features;
- Poor quality releases. The effort spent correcting defects delays the deliveries, and;
- Lack of communication among the team members and management.

Business understanding may be unclear at the Pre-Project Phase. Hence, the planning is detailed at the Project phase, but eventual deviations between the planned and executed are monitored, and such information is shared with stakeholders.

The company developed a Redmine panel, to overcome the controlling and monitoring of planning iterations, which is visualized by the managers, clients and mainly by the project team. This panel is composed by a burndown (Figure 2) and pie charts (Figure 3). Pie charts present the distribution of task types in the iteration.

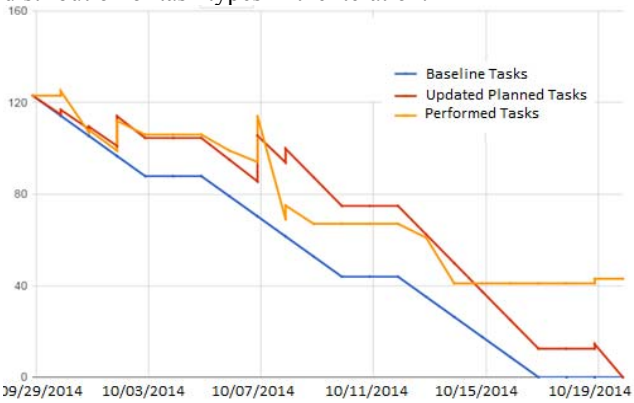

Figure 2: Burndown chart to track planning iterations

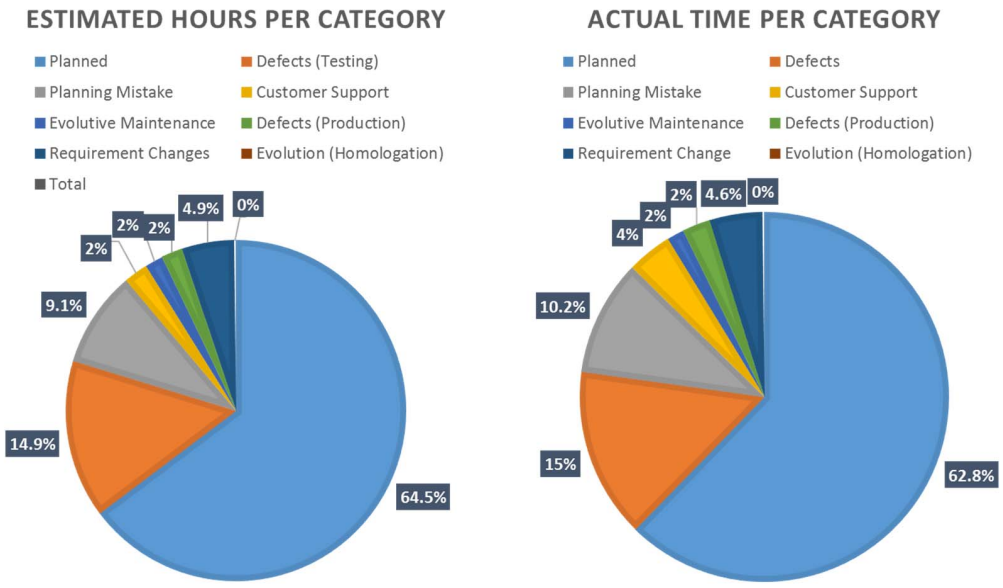

Figure 3: Panel to track planning iterations

TABLE 1. GENERAL PROJECT OVERVIEW

Phase	Project Name	Leader	GO-LIVE Date	Situation	Perform. Index
Project	SISPLAN – Phase 2	Cristine	30/01/2015		1.00
Pre-Project	SISCCON – Phase 2	Cristine			
Project	PTVV - Retirada	Souza	30/04/2013		1.00
Project	EDI – Clientes	Macedo			2.00
Pre-Project	Portal RFB	Souza			
Project	Integração	Macedo			0.45

Changes and quality control were implemented to justify the differences between the initial and current planning. Besides the need for a flexible workflow to accommodate changes, these modifications should be documented and their impact evaluated. If a change impacts on the schedule, deliveries, or the required effort, the project burndown chart shows such deviation.

The burndown chart (Figure 2) helps to monitor tasks planned for the iteration. It is possible to observe deviations from the initial baseline (blue line) in both the actual effort (red line) and the planned/estimated (yellow line).

Such deviations emerge from changes as the iterations progress, like unplanned tasks and defect corrections. These new tasks are recorded as the behavior shown in the red line, which represents the cumulative estimated effort for both tasks defined in the baseline and unplanned tasks.

For each new unplanned task, the inclination of the red line tends to increase according to the new estimated effort. The yellow line represents the actual effort spent on complete tasks.

The pie charts (Figure 3) highlight the unplanned tasks for the iterations of one project, in terms of effort and time. Planned tasks defined from the baseline account for almost two thirds of the spent effort and time. Other tasks include:

- Defects (Testing) – Additional tasks to correct defects identified during test stages (including automated tests);
- Planning Mistakes – Additional tasks due to planning mistakes occurring by communication issues, for instance.
- Customer Support – Additional tasks to customer requests, like urgent policy or regulation changes for a system already in production.
- Evolutive Maintenance – Unplanned tasks for evolving a software feature already in production environment.
- Evolution (Homologation) – Similar to the previous one, but the software feature is in the homologation tests phase.
- Defects (Production) – Additional tasks for correcting defects identified in the production environment.
- Requirement Change – Unplanned tasks for requirements changed during the software development.

During the CP adoption, practitioners in the company demanded more operational and organizational visibility. It influenced the model to have information regarding the tasks progress updated in the microplanning (operational information) and such updates feeding the macro planning

tools (organizational information). For a better understanding regarding the proposed indicators, project managers got more involved in project team daily activities. This way, not only the project managers started to engage development meetings, but also the team members got involved in project planning activities as they had relevant knowledge about all backlog items.

Even after changing the management model, it is possible to observe important Scrum practices influencing and supporting CP: (1) planning meetings at the beginning of each iteration, (2) daily meetings (stand-up meetings), (3) review, (4) retrospective, besides the Kanban board.

The use of performance indicators and monitoring boards allowed the project managers to allocate resources more efficiently. Furthermore, continuous measurement promoted a better tracking of the project progress, as well as to provide faster feedback for stakeholders. Finally, these initiatives were perceived as important by different company members during the retrospective meetings.

Still, there are some open challenges, such as (a) to implement CP in strategic level, providing managerial and up-to-date visualization for portfolios; (b) to implement mechanisms to make easier tasks monitoring and control; (d) to adopt tool support capable of automatically synchronize planned and actual effort data.

V. Discussion

Suomalainen et al. [19] described a multiple case study from three global information and communication technology (ICT) companies, which were selected due to their adoption of agile organizational practices, evolving towards a lean approach. Those case companies provide IT products and services for different domains and are referred to as Case A, B, and C, having about 1.800, 1.000, 18.000 employees, respectively.

A. Case A

Case A is an IT provider of products and services based on solutions for the wireless and automotive industries, having about 1.800 employees.

The main motivation for adopting CP concerns the constant need for changes on plans, considering its business environment is in frequent changing. Besides, people at Case A identified the need for transparency, so that a continuous visibility could be achieved for both software development and operations, contributing to share information among employees.

This way, the planning activity moved to an iterative approach, in which plans were continuously updated based on clients and market demands. Also, the actual expenditures turned out to be considered continuously and compared to estimated budget. The strategy frame started being reviewed annually and management reviews the strategy goals quarterly or when needed. Roadmaps are updated quarterly, as well as plans, and strategy-related outcomes are reviewed quarterly.

Main perceived challenges for CP at Case A regards human aspects, for instance, the unwillingness to engage or involve. Moreover, information transparency is also a challenge, like making the goals visible to everyone and how to set them in a continuous environment.

The main perceived benefits are associated with more accurate financial estimations and, from the strategical perspective, the team became capable of presenting the current state of their work and where they expect to be in a medium and long term. Besides, people feel more confident about the management, as well as increased competitiveness and reduced costs.

B. Case B

Case B Works in the security domain, providing antivirus and other security software products, and has about 1.000 employees.

Some of the motivations for adopting CP include the fact that business and R&D unities were not working together as expected, and they perceived the need for shorter planning cycles.

By introducing CP, the company started to equalize scope, schedule, and resources, which facilitated on establishing priorities regarding scope items. For that, the team now discusses the features, organized into a board, and filling their descriptions and assigning priorities.

The planning frequency at project level started to be in three-month intervals, with a foresight for six months, in which risks drive planning and updates. Besides, the feature-level planning is performed weekly to keep priorities and status visibility up to date.

Case B faced some challenges regarding which unities would rather work with long-term than three-month plans and make the current work visible to everyone at any time.

Among the perceived benefits, the project management for R&D reduced the work in progress (WIP), as well as improved product quality. The CP allowed more effective communication and involved more stakeholders in the planning process.

At the portfolio perspective, responsible managers could visualize, in a centralized way, the current work progress performed by the team, as it was presented on war-room walls.

C. Case C

Case C provides services and consultancy on IT and R&D and has about 18.000 employees.

The team understood that practices like time-box and static backlog were not working suitably. Urgent tasks arrived continuously for the development team, and that jeopardized the time-boxed planning. It encouraged the team towards a CP approach.

The company implemented practices like business iteration, consumer value analysis, and project management by a project steering group. Business iteration activities involved a series of sprints, and the steering group met quarterly to accepts/rejects the outcomes from the other two practices.

Regarding the benefits, the feature planning became more reliable as they were not constantly modified during the product development anymore. Moreover, there is a greater involvement of team members into activities concerning the business understanding.

As challenges, the authors highlight the definition of a planning process performed integrally by the team, the understanding by the team regarding incremental development practices, and the need for making the work visible to stakeholders.

D. Similarities

Although the four cases present different contexts and different levels of CP, similar issues can be observed.

All companies adopted organizational agile practices before moving towards continuous planning or lean approaches. The dominant motivation for introducing CP concerns the constant changes in the business environment.

Regarding the CP implementation, the planning activities now occur in shorter cycles at all companies. Case A started to review the strategy frame annually and the strategy goals quarterly or when needed. Roadmaps are updated quarterly, as well as plans, and strategy-related outcomes are reviewed quarterly. Case B shortened the planning frequency at the project level to three-month intervals, with a foresight for six months. Besides, the feature-level planning is performed weekly to keep priorities and status visibility up to date. Case C works in business iterations encompassing sprints, and the steering met quarterly to accepts/rejects the management outcomes. At OWSE, planning usually occurs every two weeks.

Transparency also highlights as a similar concern and achieved benefit among the cases, which seems to be crucial for information sharing, one of the major characteristics of CP. Both cases B and C reported the involvement of development teams in managerial meetings, as well as the participation of managers into the team's daily activities. Case A reports similar practices for controlling "planned vs. actual" deviations. Other benefits include more accurate estimations and goal definitions at Case A and OWSE.

The most recurrent challenge concerns the difficulty of making the tasks progress visible for all stakeholders, including clients. In Section IV, we present visual panels available for stakeholders through the management systems, similar to Case B. Moreover, OWSE and Case B shared common difficulties on long-term plans.

E. Differences

The four cases vary significantly regarding scale and business domain. Case B stands out by pointing the lack of collaboration between business unities and R&D as the main reason for adopting CP.

The planning levels (organizational, strategic and business) also vary from case to case, and that influences the different time intervals for the planning activities. In cases B and C, the timeframe for CP at project level was three months, whereas OWSE adopts a timeframe for the macroplanning, but flexible in the microplanning.

Maybe influenced by scale issues, OWSE neither reported challenges concerning people engagement, as in case A, nor the lack of understanding of incremental development practices by the team, as in case C.

The improvement of product quality levels and reduction of work in progress in the case company B differentiated as a benefit. Another difference was an increase in the management confidence as well as an improvement in the competitiveness and lower costs in the case A.

VI. Final Remarks

From these experiences, we understand CP as a feasible approach for such dynamic scenarios in which organizations need to adapt to constant changes in market conditions.

Concerns about transparency and social aspects such as collaboration and people involvement appear as key factors when introducing CP. Furthermore, we understand from these four experiences that CP has no standardized way of performing the planning activities, requiring the tailoring for specific organizational contexts.

Adopting CP from traditional iterative processes may represent an even greater challenge. Here, we presented a new case evidencing that CP through agile management is a feasible and, maybe, smoother path than introducing CP into organizations used to perform longer iterations. However, it is also not free of risks, as it represents organizational changes. In addition, The CP adoption at all levels using a "big-bang" approach may be disastrous. It seems to be a common approach starting from the operational level and scale up to strategical management as each intermediate level is achieved.

The potential benefits of CP need more systematic investigations to increase the understanding of how different sets of practices can promote benefits at reduced risks for the myriad of existing organizational contexts.

Acknowledgment

Authors thank OWSE collaborators, and CAPES and CNPq for research grants.

References

[1] de França, B. B. N., Jeronimo Junior, H., & Travassos, G. H. (2016). Characterizing DevOps by Hearing Multiple Voices. In *Proceedings of the 30th SBES* (pp. 53-62). ACM.

[2] Fitzgerald, B., & Stol, K. J. (2015). Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software*.

[3] Suomalainen, T., Kuusela, R., Tihinen, M., 2015. Continuous planning: an important aspect of agile and lean development. Int. J. Agile Syst. Manag. 8, 132–162.

[4] Conforto, E.C., Amaral, D.C., 2010. Evaluating an agile method for planning and controlling innovative projects. Proj. Manag. J. 41, 73–80.

[5] Špundak, M., 2014. Mixed Agile/Traditional Project Management Methodology – Reality or Illusion? Procedia - Soc. Behav. Sci. 119, 939–948.

[6] Cohn, M., 2005. Agile estimating and planning. Pearson Education.

[7] Dingsøyr, T., Nerur, S., Balijepally, V., Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, *85*(6), 1213-1221.

[8] Forbrig, P., 2016. Continuous Software Engineering with Special Emphasis on Continuous Business-Process Modeling and Human-Centered Design, in: Proceedings of the 8th S-BPM'16. ACM, New York, NY, USA, pp. 11:1–11:4.

[9] McConnell, S. (2006). Software estimation: demystifying the black art. Microsoft press.