Mapping CMMI Project Management Process Areas to SCRUM Practices

Ana Sofia C. Marçal[1,2], Bruno Celso C. de Freitas[2], Felipe S. Furtado Soares[2], Arnaldo D. Belchior[1]

[1] University of Fortaleza - Master's Degree in Applied Computer Science
Av. Washington Soares 1321, 60811-341, Fortaleza - CE, Brazil

[2] C.E.S.A.R - Recife Center of Advanced Systems and Studies
Rua Bione, nº 220, Cais do Apolo 50.0303-90, Recife - PE, Brazil

Abstract - Over the past years, the Capability Maturity Model (CMM) and Capability Maturity Model Integration (CMMI) have been broadly used for assessing organizational maturity and process capability throughout the world [20]. However, the rapid pace of change in information technology has caused increasing frustration with the heavyweight plans, specifications, and other documentation imposed by contractual inertia and maturity model compliance criteria [4]. In the light of that, agile methodologies have been adopted to tackle this challenge. The aim of our paper is to present a mapping between CMMI to one of these methodologies, the Scrum. It shows how Scrum addresses the Project Management Process Areas of CMMI. This is useful for organizations that have their plan-driven process based on CMMI model and are planning to improve its processes towards agility or to help organizations to define a new project management framework based on both CMMI and Scrum practices.

1. Introduction

Over the past years, the Capability Maturity Model (CMM) and Capability Maturity Model Integration (CMMI) have been broadly used for assessing organizational maturity and process capability throughout the world [20].

Organizations are moved to adopt the CMMI framework because improvement in software quality has been associated with higher levels of CMMI compliance [7,8]. These improvements include: reduced rework, predictable engineering and milestones, measurable improvements of products and services and greater customer satisfaction [2].

According to Boehm [4], so far, the 2000's have seen a continuation of the trend toward rapid application development and an acceleration of the pace of change in information technology, in organizations, in competitive countermeasures, and in the environment. This rapid pace of change has caused increasing frustration with the heavyweight plans, specifications, and other documentation imposed by contractual inertia and maturity model compliance criteria. He complements saying that the late 1990's saw the emergence of a number of agile methods such as Adaptive Software Development, Crystal, Dynamic Systems Development, eXtreme Programming (XP), Feature Driven Development, and Scrum. All of these methods employ agile principles, such as iterative cycles, early delivery of working software and simplicity as defined in AgileManifesto [1] published in 2001.

Scrum attracted significant attention among software practitioners during last five years. Whereas the Extreme Programming method that has been widely accepted as one of the most important agile approaches has a definite programming flavour, Scrum concentrates on managing software projects.

CMM or CMMI and agile methods have also been compared in several studies, and mappings or comparisons between agile and CMMI practices have been proposed [3, 11, 14, 15, 19]. For example, Paulk [15] suggests that XP's use of stories, on site customer and continuous integration fulfill the SW-CMM requirement management goals. On the other hand, Turner and Jain [19] found in their study that several of the CMMI components and agile methods were in conflict, most of them being those addressing organizational processes. Pikkarainen and Mäntyniemi [16] conclude that many of them were also found to be supportive or neutral to each other, especially those focusing on project management.

In this context of Project Management, what can we say about Scrum alignment with CMMI? Do they can live together? How agile project management with Scrum are compliant to CMMI goals and practices? The aim of our paper is to present a mapping between CMMI to Scrum practices, showing how Scrum addresses the Project Management Process Areas of CMMI. This is useful for organizations that have their plan-driven process based on CMMI model and are planning to improve its processes towards agility or to help organizations to define a new project management framework based on both CMMI and Scrum practices.

The paper is divided as follows: Section 2 presents the background overview of CMMI and the Scrum; Section 3 focuses on describing the methodology used to do the mapping between CMMI Project Management Process Areas to Scrum practices, showing the rating, gaps and the strengths existents between them and the results from the overall mapping; The last section concludes the paper with the final remarks.

2. Background overview

2.1. CMMI

According to SEI, CMMI (Capability Maturity Model Integration) is a process improvement maturity model for the development of products and services. It consists of best practices that address development and maintenance activities that cover the product lifecycle from conception through delivery and maintenance [5].

CMMI is available in two representations: staged or continuous. Each representation organizes process areas and the application of the generics to them differently. These two representations are really just different views into the same content. A staged representation may be said to focus on the organization's processes as a whole, to provide a roadmap for process improvement with proven predefined groupings of process areas, and to provide an easy migration path from the SW-CMM. A continuous representation may be said to focus on improvement to individual process areas chosen to align with specific organizational needs and to provide an easy migration path from Electronic Industries Alliance Interim Standard (EIA/IS) 731 [[13]].

The continuous representation groups process areas by affinity categories and designates capability levels for process improvement within each process area. The staged representation organizes process areas into five maturity levels to support and guide process improvement.

CMMI for Development (CMMI-DEV) [[5]], Version 1.2, describes 22 process areas. A process area is a group of related activities that are performed collectively to achieve a set of goals. In the context of these models, processes refer to "what to do" rather than "how to do it." A process area specifies goals that describe the result of successful application and practices that describe required (and expected) activities to achieve those goals. Some goals and practices are specific to the process area; others are generic and apply across all process areas. These generics describe essential ways in which a process can be institutionalized. Institutionalization refers to a process's degree of repeatability, standardization, and sophistication of control. Process areas can be grouped into four categories: Process Management, Project Management, Engineering and Support.

2.2. Scrum

Agile methods define how the work should be carried out under agile values and principles to answer the challenges of rapid development and changing requirements. Agility, as characterized by Highsmith [9], is the ability of to both create and respond to change in order to profit in a turbulent business environment.

Ken Schwaber first described Scrum in 1996 [18] as a process that accepts that the development process is unpredictable, formalizing the "do what it takes" mentality, and has found success with numerous independent software vendors. The term is borrowed from Rugby: "[A] Scrum occurs when players from each team huddle closely together . . . in an attempt to advance down the playing field" [10].

According to Schwaber [17], Scrum starts with the premise that software development is too complex and unpredictable to be planned exactly in advance. Instead, empirical process control must be applied to ensure visibility, inspection, and adaptation. The different environmental and technical variables (such as time frame, quality, requirements, resources, implementation technologies and tools, and even development methods) must be controlled constantly in order to be able to adapt to changes flexibly. This is achieved through an iterative and incremental development process.

Scrum implements an iterative, incremental skeleton through three roles: the Product Owner, the Team, and the ScrumMaster [17], as shown in Table 1.

TABLE 1
SCRUM'S ROLES AND RESPONSABILITIES

Role	Responsibilities
Product Owner	Represents the interests of everyone with a stake in the project and its resulting system. Maintains the Product Backlog, i.e., a prioritized list of project requirements with estimated times to turn them into completed product functionality.
Team	Responsible for developing functionality. Teams are self-managing, self-organizing, and cross-functional, and they are responsible for figuring out how to turn Product Backlog into an increment of functionality within an iteration and managing their own work to do so. Team members are collectively responsible for the success of each iteration and of the project as a whole.
ScrumMaster	Responsible for managing the Scrum process, i.e., for teaching Scrum to everyone involved in the project, for implementing Scrum so that it fits within an organization's culture and still delivers the expected benefits, and for ensuring that everyone follows Scrum rules and practices.

Detailed Scrum flow is shown in Fig.1. According to Schwaber [17], a SCRUM-based project starts from a high-level vision of the system will be developed. After that, Product Backlog is created containing a list of known requirements. So, the items of the Product Backlog are prioritized and divided in small time-boxed iterations (called sprints).

Every work in SCRUM is carried out through iterations called Sprints. A Sprint is a 30-day period of development time. Schwaber [17] explains that each Sprint is initiated with a Sprint planning meeting, where the Product Owner and Team get together to collaborate about what will be done for the next Sprint. Selecting from the highest priority Product Backlog, the Product Owner tells the

Team what is desired, and the Team tells the Product Owner how much of what is desired it believes it can turn into functionality over the next Sprint. In the first sprints, most of architecture and infra-structure work is done, so less functionalities are released.

After deciding what has to be done in the next Sprint, the Team develops the Sprint Backlog, i.e., a list of tasks that must be performed to deliver a completed increment of potentially shippable product functionality by the end of the Sprint. The tasks in the list emerge as the Sprint evolves and should be divided so that each takes roughly 4 to 16 hours to finish.

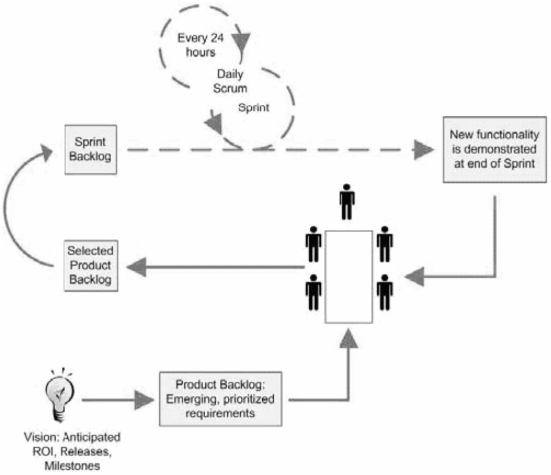

Every 24 hours
Daily Scrum
Sprint
Sprint Backlog
New functionality is demonstrated at end of Sprint
Selected Product Backlog
Product Backlog: Emerging, prioritized requirements
Vision: Anticipated ROI, Releases, Milestones

Fig. 1. Scrum process overview, from Agile Project Management with Scrum[17].

During sprints execution, the team meets dialy in 15-minute meetings to track work progress and schedule other meetings, if necessary. At the Daily Scrum, each Team member answers three questions: What have you done on this project since the last Daily Scrum Meeting? What will you do before the next meeting? Do you have any obstacles? Ate the end of each sprint, the team presents its results to the stakeholders. This presentation will guide the inspection of developed functionalities and possible adjustments in the project.

At the end of the Sprint, a Sprint review meeting is held at which the Team presents what was developed during the Sprint to the Product Owner and any other stakeholders who want to attend. After the Sprint review and prior to the next Sprint planning meeting, the ScrumMaster also holds a Sprint retrospective meeting in order to encourage the Team to revise, within the Scrum process framework, its development process to make it more effective and enjoyable for the next Sprint.

Schwaber [17] concludes saying that together, the Sprint planning meeting, the Daily Scrum, the Sprint review, and the Sprint retrospective constitute the empirical inspection and adaptation practices of Scrum.

3. Mapping CMMI Project Management Process Areas To Scrum Practices

The mapping between CMMI process areas and Scrum practices considers the staged representation of CMMI-DEV[5] model, version 1.2, released on august, 2006. The assessment just considers the project management process areas (Table 2), because it is the focus of this work.

TABLE 2
CMMI PROJECT MANAGEMENT PROCESS AREAS

Level	Process Area
Level 2	Project Planning Project Monitoring and Control Supplier Agreement Management
Level 3	Integrated Project Management +IPPD Risk Management
Level 4	Quantitative Project Management

For each process area, it was made a mapping between its specific practices and the SCRUM practices. Several considerations were identified in order to establish this mapping, identifying gaps and strengthens. After that, a coverage rating for each practice was established considering the following criteria in Table 3.

TABLE 3
CRITERIA FOR PRACTICES RATING

Rating		Criteria
U	Unsatisfied	The practice is not addressed by SCRUM
PS	Partially Satisfied	There are some evidencies addressing the practice by SCRUM, however the practice is not fully addressed.
S	Satisfied	The practice is fully addressed.

After rating phase, a coverage percentage for each process area was calculated based on the total quantity of specific practices. Afterwards, results were grouped and a complete view of the CMMI project management process areas coverage by SCRUM practices was generated. The individual mappings and the general results are described in next sub-sections.

3.1. Mapping the Process Area Project Planning

According to CMMI-DEV[[5]], the purpose of Project Planning (PP) is to establish and maintain plans that define project activities. PP has 3 specific goals (SG 1 Establish Estimates, SG 2 Develop a Project Plan, SG 3 Obtain Commitment to the Plan) enclosing 14 specific practices. The general rating for this process area is showed in the Fig. 2. The mapping of all specific practices related to this process area is presented below.

SP 1.1 Estimate the Scope of the Project

Its purpose is establish a top-level work breakdown structure (WBS) to estimate the scope of the project. In SCRUM, the initial definition of the project's scope occurs during the Pre-Game Planing phase, when the stakeholders can contribute to Product Backlog creation. In this case, the WBS is composed by the Product Backlog and the set of all pre-defined sprints, providing the necessary resources to estimate the project's scope. Detailed estimates are carried out at the beginning of each sprint, in the second part of Sprint Planning meeting. So, this practice is ***Satisfied***.

SP 1.2 Establish Estimates of Work Product and Task Attributes

Its purpose is to define what is necessary to establish and maintain estimates of the attributes of the work products and tasks. There are not explicit orientations in SCRUM to establish, for instance, size and/or complexity of items of Product Backlog and Sprint Backlog. Beyond that, no method is explicitly mentioned to guide the estimates, such as WideBand Delphi, Function Point Analysis, Use Case Points or Story Points. In this way, this practice is ***Unsatisfied***.

SP 1.3 Define Project Lifecycle

Its purpose is to define a project lifecycle on which to scope the planning effort. This practice is fully addressed by SCRUM because it defines a lifecycle as showed by Larman[12], composed by 4 phases:

- Planning: establishes a project vision and the stakeholders expectations, beyond assuring funding/budgeting for project execution.
- Staging: identifies and prioritizes the requirements (at least, for the next sprint). Break the Product Backlog in Sprints, according to previous priorization, considering the team productivity.
- Development: implements the system in a set of 30-day iterations (sprints), when, at the end of each sprint, a product increment is presented to the stakeholders.
- Release: System deployment.

So, this practice is ***Satisfied***.

SP 1.4 Determine Estimates of Effort and Cost

Its purpose is estimate the project effort and cost for the work products and tasks based on estimation rationale. In SCRUM, estimates are carried out on 2 levels: Product backlog and Sprint Backlog. Product Backlog estimates are high level estimates, so less accurate, providing a visibility of each requirement size (effort). Sprint Backlog estimates are more accurate than the first ones. Team estimation is informed by performance on previous sprints, capacity for the forthcoming sprint and the relative complexity of the tasks required to deliver the Sprint Goal [6]. However, these effort estimates don't follow a formal method or they are derived from size or complexity as required by CMMI model. SCRUM doesn't mention the importance of using a historical base. Cost is not explicitly mentioned by SCRUM, just effort, but it is necessary to calculate project's budget and funding by Product Owner. In this way, this practice is classified ***Partially Satisfied***.

SP 2.1 Establish the Budget and Schedule

Its purpose is to establish and maintain the project's budget and schedule. In SCRUM, project's budget and schedule are obtained from Product Backlog and directly derived from the estimated effort. Product Backlog is prioritized and subdivided in Sprints, considering team allocation and its workload. Schedule is composed by a set of 30-day sprints. On the other hand, SCRUM doesn't provide orientations about establishing budget. Considering this gap, this practice was rated ***Partially Satisfied***.

SP 2.2 Identify Project Risks

Its purpose is a complete identification and analysis of project risks. SCRUM considers a risk as a possible impediment for the project. The risks identification occurs in a iterative way, during daily meetings and registered in white-boards, flipcharts or impediments list. But, this risk identification doesn't occur in a systematic and parametrized manner, using, for instance, risks categories and sources. So, this practice is ***Partially Satisfied***.

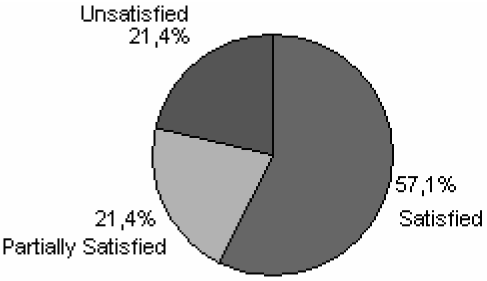


Fig. 2. General coverage for Project Planning process area.

SP 2.3 Plan for Data Management

Its purpose is to plan for the management of project data. The practices and rules defined in SCRUM contribute for good communication and promote collaboration between team and stakeholders, beyond providing visibility of project progress. According to Schwaber [[17]], any data generated by the project must be stored in a public folder, available to everyone. Many project information is communicated through meetings or documents. However, there is not a formal procedure to collect, consolidate and publish these information and data. Data privacy is another weakness, so this practice is ***Unsatisfied***.

SP 2.4 Plan for Project Resources

This practice requires a planning for necessary resources to perform the project. In SCRUM, team allocation and infra-structure availability are carried out at the beginning of the project. During its execution, Scrum Master is responsible to provide new resources when the actual resources aren't enough or new impediments related to resource scarce are reported in daily meetings. This practice is ***Satisfied***.

SP 2.5 Plan for Needed Knowledge and Skills

This practice focus is on the planning for knowledge and skills needed to perform the project. In SCRUM, teams are multi-functional groups, self-managed, composed by 7 skilled people for implementing the Sprint Backlog items. The team is composed by analysts, designers, QA, developers, data administrators, architects, among others. Senior members must mentor, monitor and guide other members. However, SCRUM doesn't mention the necessity of planning the knowledge and skills needed to perform the project activities, such as: identify which knowledge and skills are needed, analyze the knowledge and skills of the available resources and select the mechanisms to provide knowledge and skills not found in the organization. In this way, this practice is considered ***Unsatisfied***.

SP 2.6 Plan Stakeholder Involvement

Its purpose is involve identified stakeholders. SCRUM defines how stakeholders will be involved during the project execution. This involvement is monitored by Scrum Master and registered in a communication plan. In this way, this practice is considered ***Satisfied***.

SP 2.7 Establish the Project Plan

Its purpose is to establish and maintain the overall project plan content. According to Schwaber [17], the minimum plan necessary to start a Scrum project consists of a vision and a Product Backlog. The vision describes why the project is being undertaken and what the desired end state is. The Product Backlog defines the functional and no functional requirements that the system should meet to deliver the vision, prioritized and estimated. Vision document and product backlog compose a base for elaborating a high-level project plan. So, this practice is rated ***Satisfied***.

SP 3.1 Review Plans That Affect the Project

Its purpose is to review all plans that affect the project to understand project commitments. In SCRUM, plans are revised at the beginning of each sprint and possible adaptations are carried out in accordance with change requirements and technologies. CMMI model doesn't explicit which plans need to be revised, such as QA plan, CM plan, Test plan, among others. Therefore, this practice is considered ***Satisfied***.

SP 3.2 Reconcile Work and Resource Levels

Its purpose is to reconcile the project plan to reflect available and estimated resources. This practice is ***Satisfied***, because this work reconciliation occurs during the Sprint Planning meeting. The team, Product Owner and Scrum Master define the functionalities that will be developed at the sprint.

SP 3.3 Obtain Plan Commitment

This practice addresses the commitment from relevant stakeholders responsible for performing and supporting plan execution. The plan commitment occurs continuously at the beginning of each sprint, during the Sprint Planning Meeting. Product Owner, Scrum Master and the team define the priorities of Product Backlog for each sprint and which items will be developed at the next sprint. During the Sprint execution, if the team workload is not enough to develop all items agreed, the Product Owner can decide which sprint backlog item could be removed. On the other hand, if the team workload is greater than the effort necessary to implement the Sprint Backlog items, Product Owner can allocate other items from Product Backlog. Therefore, this practice is ***Satisfied***.

3.2. Mapping the Process Area Project Monitoring and Control

The purpose of Project Monitoring and Control (PMC) is to provide an understanding of the project's progress so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan [[5]]. PMC encloses 10 specific practices grouped in 2 specific goals (SG 1 Monitor Project Against Plans and SG 2 Manage Corrective Action to Closure). The mapping of all specific practices related to this process area is presented below. The general rating for this process area is showed in the Fig. 3.

SP 1.1 Monitor Project Planning Parameters

Its purpose is to monitor the actual values of the project planning parameters against the project plan. In SCRUM, project tracking occurs through Burndown graphs and project meetings previously mentioned. Product Burndown shows the speed of releasing Product Backlog items by the team. It is analyzed at the end of each Sprint. It helps to monitor the planning of functionalities releasing, providing visibility if the sprint goal will be successfully achieved or if is necessary re-planning of the sprint scope to attend the planned date. Sprint Burndown dialy shows the speed of the team and the progress of its activities in a Sprint. Sprint Burndows provides a supporting tool for planning necessary corrective actions.

Tracking meetings, mainly daily meetings, allows a day-by-day tracking of the team progress and realize the existent difficulties to carry out the planned activities. These difficulties must be quickly resolved by Scrum Master in order to the team doesn't lost its focus or Sprint goal.

In spite of it, like the cost, size and effort estimates are not carried out in a sistematic manner, there is not a formal tracking of them as required by CMMI model. So, trainings tracking is also informal, because there is not a formal planning of them. In this way, this practice is rated ***Partially Satisfied***.

SP 1.2 Monitor Commitments

Its purpose is to monitor against those identified in the project plan. In SCRUM, commitments of each sprint are established during the Sprint Planning Meeting and monitored through Sprint Burndown and daily meetings and, finally, reviewed in Sprint Retrospective Meeting. During a sprint, the team cannot receive any additional work from stakeholders or Product Owner. Just team itself can update Sprint Backlog in order to maintain uninterrupted focus on Sprint activities. So, this practice is rated ***Satisfied***.

SP 1.3 Monitor Project Risks

Its purpose is to monitor risks against those identified in the project plan. In SCRUM, daily meeting can help to identify impediments (issues, dependencies, risks etc), so risks are identified but they are not analyzed properly. Risks are registered in whiteboards, flipcharts or impediment lists and monitored by Scrum Master, so they are tracked in a informal way. In other words, this practice is considered ***Partially Satisfied***.

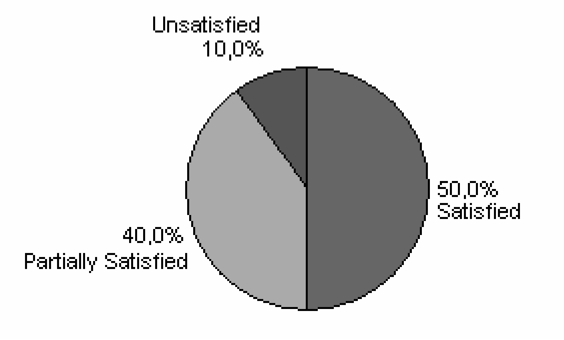


Fig. 3. General coverage for Project Monitoring and Control process area.

SP 1.4 Monitor Data Management

Its purpose is to monitor the management of project data against the project plan. This practice is ***Unsatisfied*** because SCRUM doesn't adopt any procedure for planning and tracking data management, as required by CMMI model.

SP 1.5 Monitor Stakeholder Involvement

Its purpose is to monitor stakeholder involvement against the project plan. In SCRUM, stakeholders involvement tracking is carried out during project meetings by Scrum Master, responsible to assure that all stakeholders understand and respect the rules and practices defined in SCRUM. In spite of there is not any register about this monitoring, this practice is ***Satisfied*** because it is possible found indirect evidences such as impediment list update and Product Backlog and Sprint Backlog update.

SP 1.6 Conduct Progress Reviews

Its purpose is periodically reviewing the project's progress, performance, and issues. In SCRUM, project control is carried out by frequent inspections and progress review meetings (Daily Meeting and Sprint Review Meeting). In this way, this practice is fully ***Satisfied***.

SP 1.7 Conduct Milestone Reviews

Its purpose is to review the accomplishments and results of the project at selected project milestones. As commented in SP 1.6, milestone reviews occur at the end of each Sprint. In Sprint Review Meetings, the project progress is inspected, providing visibility of the accomplishment of commitments. Therefore, this practice is considered ***Satisfied***.

SP 2.1 Analyze Issues

Its purpose is to collect and analyze the issues and determine the corrective actions necessary to address the issues. During the Daily Scrum Meetings, team reports all impediments against expected quality or performance levels. Impediments are registered in whiteboard, flipchart or impediment list and they are erased when resolved. Beyond that, Scrum Master is in charge of resolving the impediments as soon as possible, taking appropriate corrective actions. So, this practice is ***Satisfied***.

SP 2.2 Take Corrective Action

Its purpose is to take corrective action on identified issues. As mentioned before, corrective actions are taken for the impediments found. However, there is not any register of how these actions are planned and monitored. So, this practice is ***Partially Satisfied***.

SP 2.3 Manage Corrective Action

Its purpose is to manage corrective actions to closure. As mentioned before, impediments registered in whiteboard, flipchart or impediment list are erased when resolved. So, all corrective actions are monitored to closure. However, the results of these actions are not analyzed to determine its efficacy. Therefore, this practice is ***Partially Satisfied***.

3.3 Mapping the Process Area Supplier Agreement Management

The purpose of Supplier Agreement Management (SAM) is to manage the acquisition of products from suppliers [5]. In SCRUM, there is not any practice addressing the acquisition of products and product components that are delivered to the project's customer. So, all of its specific practices are ***Unsatisfied***.

3.4 Mapping the Process Area Integrated Project Management + IPPD

According to CMMI-DEV[5], the purpose of Integrated Project Management (IPM) is to establish and manage the project and the involvement of the relevant stakeholders according to an integrated and defined process that is tailored from the organization's set of standard processes. For Integrated Product and Process Development (IPPD), IPM +IPPD also covers the establishment of a shared vision for the project and the establishment of integrated teams that will carry out objectives of the project.

IPM +IPPD is composed by 3 specific goals (SG 1 Use the Project's Defined Process, SG 2 Coordinate and Collaborate with Relevant Stakeholders and SG 3 Apply IPPD - Integrated Product and Process Development Principles) enclosing 14 specific practices.

In this process area, all of 6 specific practices related to SG1 ("the project is conducted using a defined process that is tailored from the organization's set of standard processes") are assessed ***Unsatisfied***. Because SCRUM doesn't define a set of organizational standard processes, but it just establishes a set of practices and rules defined for the project. In other words, the project's defined process is not derived from a set of organizational processes.

The mapping of other specific practices related to this process area is presented below. The general rating for this process area is showed in the Fig. 4.

SP 2.1 Manage Stakeholder Involvement

Its purpose is to manage the involvement of the relevant stakeholders in the project. As commented before, in PMC SP 1.5, SCRUM practices and rules implicitly define how stakeholders will be involved in the project tracking. This involvement is monitored by Scrum Master. So, in this way, this practice is ***Satisfied***.

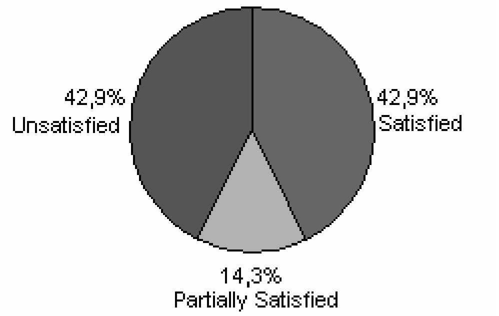


Fig. 4. General coverage for IPM+IPPD process area.

SP 2.2 Manage Dependencies

Its purpose is to participate with relevant stakeholders to identify, negotiate, and track critical dependencies. In SCRUM, dependencies and risks can be managed as impediments, being identified through Scrum Daily Meetings. Scrum Master is in charge to resolve any identified problem as soon as possible, so dependencies as well. However, there is not registers of negotiation, meeting minutes or agreed dates to remove such dependencies. Beyond that, there is not a planning of tracking strategies or verification of the dependencies. Therefore, this practice is ***Partially Satisfied***.

SP 2.3 Resolve Coordination Issues

Its purpose is to resolve issues with relevant stakeholders. This practice is ***Partially Satisfied***, due to the same reasons presented for SP 2.2.

SP 3.1 Establish the Project's Shared Vision

Its purpose is to establish and maintain a shared vision for the project. The Scrum planning process sets the stakeholders' expectations. The plan is a way of syncronizing stakeholders' expectations with the team's expectations [17]. It is also important that the whole team understands the essence of what the project or product is trying to achieve. This is where the Project Vision comes in. It should be as short and as accessible as possible but communicate the substance and character of the undertaking [6]. Therefore, this practice is ***Satisfied***.

SP 3.2 Establish the Integrated Team Structure

Its purpose is to establish and maintain the integrated team structure for the project. In SCRUM, when several teams work in a collaborative environment, this project is called scaled project and the mechanisms used to coordinate their activities are called scaling mechanisms [17]. When scaling Scrum to larger projects few guidelines must be followed:

- Attempt to keep the team size to eight.
- Do not engage all the teams until the infrastructure is built. This may require the first few sprints being done by only a single team and primarily non-functional requirements being built.
- Divide the work into teams that logically represent the work and minimize the need for multiple assignments of people.
- Create a daily scrum meeting of representatives of each scrum, held after the daily scrum.

So, these guides help to establish a integrated team structure and this practice is ***Satisfied***.

SP 3.3 Allocate Requirements to Integrated Teams

Its purpose is to allocate requirements, responsibilities, tasks, and interfaces to teams in the integrated team structure. In SCRUM, some practices are critical for the scaled project success, such as [17]:

- Build a scalable infra-structure before scaling the project. This structure must supports just one team initially and grows during successive sprints. Non functional requirements to build the scaling infra-structure must to be added to Product Backlog and prioritized joint with other business functionalities in the SCRUM staging phase, before the first sprint;

- Always release business value in the sprints carried out for building the infra-structure;
- Optimize the initial team capabilities and prepare additional teams. Additional teams must be composed by, at least, 1 member of initial team, playing a infra-structure and architect expert role.

All of these practices establish the necessary requirements to integrated teams. So, this practice is ***Satisfied***.

SP 3.4 Establish Integrated Teams

Its purpose is to establish and maintain integrated teams in the structure. This practice is ***Satisfied***, due to the same reasons presented for SP 3.2 e SP3.3.

SP 3.5 Ensure Collaboration among Interfacing Teams

This practice ensures collaboration among interfacing teams. This practice is ***Satisfied***, due to the same reasons presented for SP 3.2 e SP3.3.

3.5 Mapping the Process Area Risk Management

The purpose of Risk Management (RSKM) is to identify potential problems before they occur so that risk-handling activities can be planned and invoked as needed across the life of the product or project to mitigate adverse impacts on achieving objectives [5]. As previously mentioned, in SCRUM, risks are identified, but it doesn't mention practices to define sources, parameters or categories to analyze and control the risk management effort. In SCRUM, there are not strategies to risk response or a mitigation plan for the critical risks based on historical bases or something like that. In this way, the assessment, categorization and priorization of these risks occur in a informal manner. Therefore, all of specific practices of RSKM are ***Unsatisfied***, except SP 2.1 Identify Risks, because it is ***Partially Satisfied***. The general rating for this process area is showed in the Fig. 5.

3.6 Mapping the Process Area Quantitative Project Management

The purpose of Quantitative Project Management (QPM) is to quantitatively manage the project's defined process to achieve the project's established quality and process-performance objectives [5]. According to CMMI-DEV, in this process area, the quality and process-performance objectives, measures, and baselines identified are developed as described in the Organizational Process Performance process area. Subsequently, the results of performing the processes associated with the Quantitative Project Management process area (e.g., measurement definitions and measurement data) become part of the organizational process assets referred to in the Organizational Process Performance process area. In SCRUM, there is not any practice addressing this process area. Therefore, all of its practices are ***Unsatisfied***.

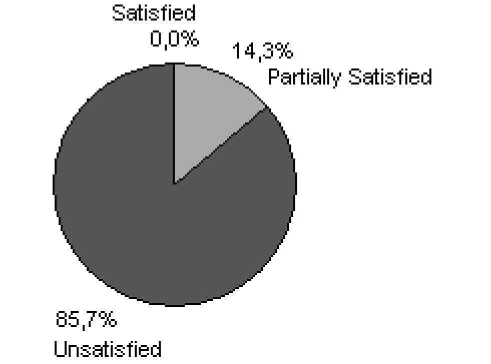
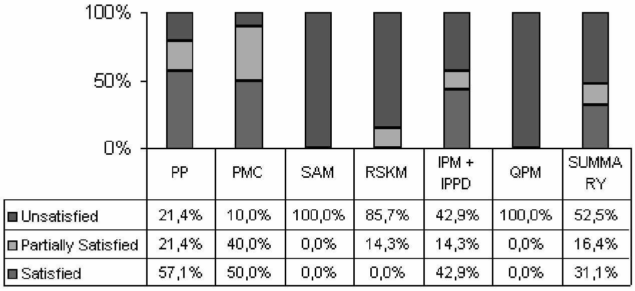


Fig. 5. General coverage for Risk Management process area.

3.7 Overall Results from the Mapping

General results of the mapping are showed in the Fig. 6, a consolidated view of the coverage of CMMI project management process areas by SCRUM practices.

This result shows that 31,1% of specific practices of CMMI project management process areas are Satisfied, 16,4% are Partially Satisfied and 52,5% are Unsatisfied. In other words, SCRUM is not fully complaint with CMMI project management process areas, mainly related to SAM, RSKM and QPM process areas. If SAM and QPM are not considered, 42,2% of the specific practices of CMMI project management process areas are Satisfied, 22,2% are Partially Satisfied and 35,6% are Unsatisfied.



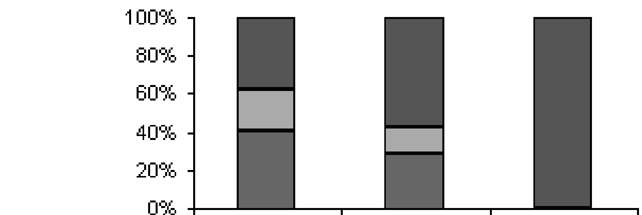
	PP	PMC	SAM	RSKM	IPM + IPPD	QPM	SUMMARY
■ Unsatisfied	21,4%	10,0%	100,0%	85,7%	42,9%	100,0%	52,5%
▨ Partially Satisfied	21,4%	40,0%	0,0%	14,3%	14,3%	0,0%	16,4%
■ Satisfied	57,1%	50,0%	0,0%	0,0%	42,9%	0,0%	31,1%

Fig. 6. CMMI Project Management Process Areas covered by SCRUM.

Considering each CMMI project management process area according to its maturity level, another analysis can be made as showed in the Fig. 7.

In this way, process areas related to maturity level 2 (PP, PMC and SAM) have 40,6% of its specific practices Satisfied by SCRUM, 21,9% are Partially Satisfied and 37,5% are Unsatisfied. If SAM is not applicable to the organization context, SCRUM becomes more attractive (54,2% Satisfied, 29,2% Partially Satisfied and 16,7% Unsatisfied).

But the things don't sound so good if considering process areas related to maturity level 3 (IPM+IPPD and RSKM). These process areas have 28,6% of its specific practices Satisfied by SCRUM, 14,3% are Partially Satisfied and 57,1% are Unsatisfied, because a weak adherence of SCRUM to RSKM practices, lack of organizational process and systematic use of historical bases, as required by IPM process area. Finally, considering process areas related to maturity level 4, these specific practices are 100% Unsatisfied by SCRUM.



	PM Category Level 2	PM Category Level 3	PM Category Level 4
■ Unsatisfied	37,5%	57,1%	100,0%
▨ Partially Satisfied	21,9%	14,3%	0,0%
■ Satisfied	40,6%	28,6%	0,0%

Fig. 7. CMMI Project Management Process Areas covered by SCRUM. Rating grouped by maturity level, considering only the project management process area category.

4. Final Considerations

The goal of this research was to compare the agile method Scrum in relation to the Project Management process areas of the CMMI model, showing the gaps and the strengths existents between them. From this mapping we conclude that Scrum does not cover all the specific practices of the project management process area, but it could be tailored to be more compliant with CMMI. On the other hand, we can conclude that plan-driven process based on CMMI model can be improved by adding some Scrum agile practices on their activities.

Therefore, SCRUM is a recommended starting point to organizations with small teams and without defined processes, because most of necessary fundamentals for institutionalizing CMMI project management process areas related to maturity level 2 are created without compromising the agility needed by these organizations. However, organizations searching higher maturity levels are not fully attended by SCRUM practices. Other alternative practices are necessary to complement SCRUM and address CMMI requirements. The great challenge is identify how alternative practices can complement SCRUM without losing agility.

In the light of that, it is suggested as future works: extension of this mapping with CMMI generic practices; inclusion of mapping with other agile methodologies in order to establish a complete view about how other methodologies can address CMMI project management practices; and the definition of a new agile project management methodology considering the best practices found in the previous work mapping and some alternative practices of Agile Project Management [9], if necessary.

References

[1] AgileManifesto, Manifesto for Agile Software Development http://agilemanifesto.org/ (December 2006)

[2] Alleman G., Blending Agile Development Methods with CMMI, Cutter IT Journal, Vol. 17, No. 6, June 2004.

[3] Anderson D. J., "Stretching Agile to fit CMMI Level 3 the story of creating MSF for CMMI Process Improvement at Microsoft Corporation" presented at Agile2005Conference, http://www.agilemanagement.net/Articles/Papers/Agile_2005_Paper_DJA_v1_5.pdf. (December 2006)

[4] Boehm B., "A View of 20th and 21st Century Software Engineering", ICSE 2006, May 2006.

[5] CMMI-DEV, CMMI for Development, V1.2 model, CMU/SEI-2006-TR-008. Software Engineering Institute, 2006.

[6] Cochango, Scrum for team systems, http://www.scrumforteamsystem.com, (December 2006).

[7] Goldenson D., and Diane L. Gibson. Demonstrating the Impact and Benefits of CMMI: An Update and Preliminary Results, CMU/SEI-2003-SR-009. Software Engineering Institute, 2003.

[8] Herbsleb J., Anita Carleton, James Rozum, Jane Siegel, and David Zubrow. Benefits of CMM-Based Software Process Improvement: Initial Results, CMU/SEI-94-TR-013. Software Engineering Institute, 1994.

[9] Highsmith J., Agile Project Management, Creating innovative products, AddisonWesley, 2004.

[10] Highsmith J., Agile Software Development Ecosystems, Addison–Wesley, Boston, MA, 2002.

[11] Kähkönen T. and P. Abrahamsson, "Achieving CMMI Level 2 with Enhanced Extreme Programming Approach," In proceedings of the 5th International Conference on Product Focused Software Process Improvement, pp. 378392, 2004

[12] Larman C., Agile & Iterative Development, A Manager's Guide, Addison-Wesley, 2004.

[13] Menezes W., "To CMMI or Not to CMMI: Issues to Think About." Crosstalk 15,2 (February 2002): 9-11

[14] Nawrocki J., W. Bartosz, and A. Wojciechowski, "Toward Maturity Model for eXtreme Programming," In proceedings of the 27th Euromicro Conference, pp. 233239, 2001.

[15] Paulk M. C., "Extreme Programming from a CMM Perspective," Software, vol. 18, issue 6, pp. 1926, 2001.

[16] Pikkarainen M., and Annukka Mäntyniemi, An Approach for Using CMMI in Agile Software Development Assessments: Experiences from Three Case Studies, accepted for SPICE 2006 conference, that will be in Luxemburg at 45th at May 2006.

[17] Schwaber K., Agile Project Management With Scrum, Microsoft, 2004.

[18] Schwaber K., "Controlled chaos: living on the edge", http://www.controlchaos.com/old-site/ap.htm, (December 2006).

[19] Turner R., and A. Jain, "Agile Meets CMMI: Culture Clash or Common Cause," In proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods XP/Agile Universe, pp. 153165, 2002.

[20] Zubrow D., "Current Trends in the Adoption of the CMMI Product Suite," In Proceedings of the 27th Annual International Computer Software and Applications Conference, pp. 126129, 2003.