



Universidade Federal de Pernambuco

Graduação em Ciência da Computação

Centro de Informática

---

# MAPEAMENTO DAS PRÁTICAS DE SCRUM NAS ÁREAS DE PROCESSO DO CMMI E UMA PROPOSTA PARA SUA ADERÊNCIA

---

Trabalho de Graduação em Qualidade de Software

**Aluno:** Henrique Borges Alencar Siqueira (hbas@cin.ufpe.br)

**Orientador:** Alexandre Marcos Lins de Vasconcelos (amlv@cin.ufpe.br)

Recife, 22 de agosto de 2007

## ASSINATURAS

Este Trabalho de Graduação é resultado dos esforços do aluno Henrique Borges Alencar Siqueira, sob a orientação do professor Alexandre Marcos Lins de Vasconcelos, sob o título inicial de “*Mapeamento das práticas de Scrum nas áreas de processo do CMMI e uma análise de sua aderência*”, posteriormente modificado para “*Mapeamento das práticas de Scrum nas áreas de processo do CMMI e uma proposta para sua aderência*”. Todos abaixo estão de acordo com o conteúdo deste documento e os resultados deste Trabalho de Graduação.

---

Henrique Borges Alencar Siqueira

---

Alexandre Marcos Lins de Vasconcelos

|                                                                                                                                      |                                                                                                  |                      |
|--------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|----------------------|
|  <b>Centro de Informática</b><br><small>UFPE</small> | Mapeamento das práticas de Scrum nas áreas de processo do CMMI e uma proposta para sua aderência | 22 de agosto de 2007 |
|--------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|----------------------|

*“Quanto melhor o processo, mais talentosos e experientes  
precisam ser os funcionários a executá-lo”*  
(Tom DeMarco)

## Dedicatória

---

A meus pais, pelo apoio, incentivo e dedicação. Sem eles jamais teria chegado “tão longe”.

## Agradecimentos

---

À minha família, por propiciar as condições necessárias para a criação deste trabalho.

Ao meu orientador, Prof. Dr. Alexandre Vasconcelos, por sua paciência e conselhos durante todo o processo de definição e orientação do trabalho.

À Érica por todo respeito, carinho e compreensão.

Aos sócios e estagiários da FAST Soluções Tecnológicas, pelo apoio e pela paciência nas horas em que precisei me ausentar para realizar este trabalho.

Aos professores do Centro de Informática, por todo conhecimento que me foi repassado.

Aos meus amigos e amigas, pelo apoio, pelos conselhos, pelas alegrias divididas e experiências vividas.

## Resumo

---

Recentemente a comunidade de software vem se deparando com um grupo de novas metodologias de desenvolvimento de software, classificadas como “metodologias ágeis”. Algumas das metodologias que fazem parte deste grupo são Extreme Programming (XP) e Scrum.

Em paralelo à disseminação das metodologias ágeis, pesquisas indicam um crescimento na adoção de modelos de qualidade como ISO 9000 e *Capability Maturity Model Integration* (CMMI). Com as rápidas mudanças no mercado de software e a busca crescente por agilidade, companhias que têm seus processos baseados em modelos de qualidade como o CMMI percebem uma frustração crescente de seus funcionários devido aos planos, especificações e documentações pesados, muitas vezes impostos para simplificar o processo de certificação e a conformidade com os modelos. Apesar de conter valores aparentemente opostos, o uso de metodologias ágeis como forma de implementação dos requisitos e práticas dos modelos de qualidade é possível.

Este trabalho apresenta uma proposta de extensão do Scrum para torná-lo mais aderente à área de Gerenciamento de Projetos do CMMI. Para realizar este estudo, inicialmente foi feito um mapeamento das práticas de Scrum nas áreas de processo do CMMI e depois, para alguns problemas identificados, soluções baseadas em práticas e atividades de outras metodologias ágeis foram propostas. Este trabalho é útil para empresas que desejam manter seus processos de gerenciamento ágeis ao mesmo tempo em que são avaliadas e certificadas pelo CMMI

**Palavras chaves:** Capability Maturity Model Integration (CMMI), Scrum, Metodologias ágeis, Qualidade de software.

## Abstract

---

Lately, the software community is coming across a new group of software development methodologies, classified as “agile methodologies”. Some of them are Extreme Programming (XP) and Scrum.

In parallel to the dissemination of the agile methodologies, researches indicate a growth in the adoption of quality models as ISO 9000 and Capability Maturity Model Integration (CMMI). With the rapid pace of change in the information technology and the crescent search for agility, companies with process based on quality models like CMMI realize increasing frustration with the heavyweight plans, specifications, and other documentation imposed by maturity model compliance and certification process simplification. Although with apparent opposing values, the use of agile methodologies as a implementation of the requirements and practices of quality models is possible.

This work proposes an extension to Scrum, making it more adherent to the CMMI Project Management area. To carry through this study, initially a mapping of Scrum practices in CMMI process areas was made and later, for some identified problems, solutions based on activities of other agile methodologies had been proposed. This work is useful for companies with the desire to keep its agile management processes while still being able to be evaluated and certified by CMMI.

**Keywords:** Capability Maturity Model Integration (CMMI), Scrum, Agile Methodologies, Software quality.

## Índice

|     |                                                                     |    |
|-----|---------------------------------------------------------------------|----|
| 1.  | Introdução .....                                                    | 11 |
| 1.1 | Motivação e contexto .....                                          | 11 |
| 1.2 | Objetivos .....                                                     | 13 |
| 1.3 | Estrutura do Trabalho .....                                         | 14 |
| 2.  | CMMI .....                                                          | 16 |
| 2.1 | Visão geral e objetivos.....                                        | 16 |
| 2.2 | Representações.....                                                 | 17 |
| 2.3 | Áreas de processo e seus componentes.....                           | 18 |
| 2.4 | Níveis .....                                                        | 19 |
| 2.5 | Categorias .....                                                    | 20 |
| 2.6 | Conclusão do capítulo.....                                          | 21 |
| 3.  | Scrum .....                                                         | 22 |
| 3.1 | Visão geral .....                                                   | 22 |
| 3.2 | Papéis e responsabilidades.....                                     | 23 |
| 3.3 | Ciclo de vida .....                                                 | 24 |
| 3.4 | Fluxo de desenvolvimento .....                                      | 24 |
| 3.5 | Sprints .....                                                       | 26 |
| 3.6 | Múltiplos times em um único projeto .....                           | 27 |
| 3.7 | Conclusão do capítulo.....                                          | 28 |
| 4.  | Mapeando Scrum nas áreas de processo de gerenciamento do CMMI ..... | 29 |
| 4.1 | Planejamento de Projeto (PP) .....                                  | 29 |
| 4.2 | Monitoramento e Controle do Projeto (PMC) .....                     | 33 |
| 4.3 | Gerenciamento Integrado de Projetos (IPM) + IPPD .....              | 36 |
| 4.4 | Gerenciamento de Riscos (RSKM).....                                 | 40 |
| 4.5 | Gerenciamento de Acordos com Fornecedores (SAM).....                | 41 |
| 4.6 | Gerenciamento Quantitativo do Projeto (QPM) .....                   | 41 |
| 4.7 | Análise dos resultados.....                                         | 42 |
| 5.  | Tornando Scrum mais aderente ao CMMI.....                           | 45 |
| 5.1 | Priorização das áreas de processo não-satisfeitas .....             | 45 |
| 5.2 | Estendendo o processo de estimativa em Scrum .....                  | 46 |
| 5.3 | Estendendo o processo de definição do orçamento em Scrum .....      | 47 |
| 5.4 | Estendendo o processo de gerenciamento de riscos em Scrum .....     | 47 |
| 5.5 | Estendendo o processo de resolução de impedimentos em Scrum .....   | 48 |
| 5.6 | Revisão do mapeamento considerando as extensões propostas.....      | 49 |
| 6.  | Conclusões .....                                                    | 51 |
| 6.1 | Objetivos atingidos .....                                           | 51 |
| 6.2 | Trabalhos futuros .....                                             | 52 |
| 6.3 | Considerações finais .....                                          | 53 |
| 7.  | Referências Bibliográficas .....                                    | 54 |

## Índice de Figuras

|                                                                                       |    |
|---------------------------------------------------------------------------------------|----|
| Figura 3.2 – Fluxo de desenvolvimento do Scrum [COCHANGO 2006].....                   | 25 |
| Figura 3.3 – <i>Product Burndown</i> [COCHANGO 2006] .....                            | 26 |
| Figura 3.4 – Atividades da Sprint .....                                               | 27 |
| Figura 4.1 - Satisfação das práticas de Gerenciamento de Projetos.....                | 42 |
| Figura 4.2 - Satisfação das práticas por nível de maturidade .....                    | 43 |
| Figura 5.1 - Satisfação das práticas de Gerenciamento de Projetos após extensões..... | 50 |

## Índice de Tabelas

|                                                                                 |    |
|---------------------------------------------------------------------------------|----|
| Tabela 2.1 - Comparação entre as vantagens do modelo contínuo e estagiado ..... | 17 |
| Tabela 2.2 – Áreas de processo do CMMI .....                                    | 20 |
| Tabela 4.1 - Pontuações do mapeamento e seus significados.....                  | 29 |
| Tabela 4.2 - Satisfação das práticas de PP .....                                | 33 |
| Tabela 4.3 - Satisfação das práticas de PMC .....                               | 36 |
| Tabela 4.4 - Satisfação das práticas de IPM + IPPD .....                        | 39 |
| Tabela 4.5 - Satisfação das práticas de RSKM .....                              | 40 |
| Tabela 4.6 - Satisfação das práticas de SAM.....                                | 41 |
| Tabela 4.7 - Satisfação das práticas de QPM.....                                | 42 |
| Tabela 4.8 – Percentuais de satisfação por áreas de processo .....              | 43 |
| Tabela 4.9 – Percentuais de satisfação por níveis de maturidade .....           | 43 |
| Tabela 5.1 - Satisfação das práticas após extensões .....                       | 49 |
| Tabela 5.2 – Satisfação por nível de maturidade após extensões .....            | 50 |

## 1. Introdução

---

Este capítulo introduz conceitos e apresenta os objetivos deste trabalho estruturado conforme descrito abaixo:

- Seção 1.1 – Motivação e contexto: descreve a motivação para realização deste estudo, incluindo aspectos sobre engenharia de software, qualidade de software, metodologias ágeis e o problema ao qual este trabalho se propõe a resolver;
- Seção 1.2 – Objetivos do Trabalho: descreve os objetivos do trabalho e como o mesmo será conduzido; e
- Seção 1.3 – Estrutura da Dissertação: descreve como este documento está organizado.

### 1.1 Motivação e contexto

---

Hoje, mais do que nunca, as organizações se vêm desenvolvendo e integrando softwares cada vez mais complexos, tendo que produzi-los mais rápido, mais baratos e com melhor qualidade do que concorrentes e predecessores [CHRISSIS et al. 2003].

Desde o fim da década de 1960, a indústria do software procura tornar o seu desenvolvimento mais previsível e eficiente [NAUR et al. 1969]. Na tentativa de organizar as atividades de desenvolvimento de software, surgiu a engenharia de software e as primeiras metodologias de desenvolvimento. Estas primeiras metodologias eram baseadas fortemente em planejamento e documentação extensivos, como em outras engenharias. No entanto, as primeiras metodologias não se mostraram bem sucedidas, uma vez que os problemas de estouro de prazo e orçamento nos projetos continuaram a ocorrer [STANDISH GROUP 2001].

O foco na qualidade é dar suporte à evolução e ao aprimoramento da Engenharia de Software. Segundo Pressman [PRESSMAN 1997], “o gerenciamento da qualidade total e filosofias similares fomentam uma cultura de melhoria contínua de processos, e esta cultura leva ao desenvolvimento de abordagens mais maduras de engenharia de software”. Alcançar competitividade pela qualidade, para as empresas de software, implica tanto na melhoria da qualidade dos produtos de software e serviços correlatos, como dos processos de produção e distribuição de software [SOFTEX 2006].

Qualidade é um termo amplo e controverso, com diferentes estudiosos no assunto, definindo qualidade de diferentes formas:

- W. Edwards Deming [DEMING 1985] define qualidade em termos de precisão do esforço. Neste sentido, a qualidade é maior quando planos e especificações são feitos de acordo com a capacidade da organização;

- Para Philip Crosby [CROSBY 1979] qualidade está relacionada com a conformidade às especificações. Um produto de qualidade é obtido a partir de especificações bem escritas que descrevam as necessidades do cliente;
- Tom Peters [CLIPP 1990] define qualidade como atingir às expectativas dos clientes. A qualidade do produto é medida a partir da conformidade das especificações às reais necessidades dos clientes.

Apesar de ser difícil de definir, a qualidade de software trás diversos benefícios. Dentre eles, Campelo [CAMPELO 2003] ressalta: redução do número de defeitos, confiabilidade, diminuição de re-trabalho, diminuição do ciclo de vida de produção do software, redução de custos, maior valor agregado, satisfação do cliente e maior motivação da equipe de produção.

No intuito de aumentar, medir e garantir a qualidade do software produzido, surgiram alguns modelos de qualidade de software. Dentre os modelos de qualidade propostos, os mais usados atualmente são as normas da série ISO 9000 [MELLO ET AL. 2002], *Capability Maturity Model for Software* (SW-CMM) [SEI 1995] e, recentemente, o *Capability Maturity Model Integrated* (CMMI), uma evolução e combinação do SW-CMM, o SECM, e o IPD-CMM [SEI 2006].

Surgindo paralelamente ao desenvolvimento da qualidade de software, mas com ênfase na colaboração entre o time de programação e os especialistas de negócios, comunicação face-a-face, times enxutos e auto-organizados entre outras características [AGILE ALLIANCE 2007], as metodologias ágeis propõem um meio termo entre “nenhum processo” e “processo demais”, de forma que o desenvolvimento do software não seja obstaculizado pela metodologia.

A primeira impressão de que as metodologias ágeis de desenvolvimento se opõem as metodologias mais tradicionais, simplesmente porque geram menos documentos, não toca a essência dos métodos ágeis. Duas das principais idéias por trás dos métodos ágeis, as quais os diferenciam das demais metodologias, são [FOWLER 2006]:

1. Métodos ágeis são mais adaptativos do que preexistentes. Métodos tradicionais procuram planejar em detalhes longos períodos, desta forma, é da sua natureza a resistência à mudança, uma vez que o planejamento esteja estabelecido. Métodos ágeis, por sua vez, acolhem a mudança a qualquer momento, ao ponto de adaptar a própria metodologia para serem bem sucedidos.
2. Métodos ágeis baseiam-se mais nas pessoas do que nos processos. Enquanto os métodos tradicionais procuram definir processos que vão funcionar com qualquer um que os execute, os métodos ágeis asseguram que nenhum processo pode superar as habilidades de um time. Desta forma, o papel do processo é dar suporte à equipe de desenvolvimento em seu trabalho.

Existem diversos processos e metodologias que se consideram ágeis, sendo Extreme Programming [BECK 2000], Scrum [SCHWABER 2004], Crystal [CRYSTAL 2007],

MSF for CMMI [ANDERSON 2005] e OpenUP [OPENUP 2007] apenas alguns exemplos.

Das metodologias ágeis existentes no mercado, Scrum têm se destacado nos últimos anos por focar-se nas práticas e atividades de gerenciamento de projetos reunindo ações de monitoramento e feedback que visam identificar e corrigir deficiências e/ou impedimentos no processo de desenvolvimento.

Desenvolvido pelo *Software Engineering Institute* (SEI) a partir da premissa de que “a qualidade de um sistema ou produto é altamente influenciada pela qualidade do processo usado para desenvolver e manter ele”, o CMMI foca em melhorar os processos de uma organização, dizendo quais processos devem estar presentes para aumentar a qualidade do produto. Apesar da existência de características distintas entre Scrum e o modelo CMMI, ambos possuem planos específicos para o desenvolvimento de software e buscam o melhor para que a organização produza software com qualidade [TURNER et al. 2002].

## 1.2 Objetivos

---

Quão incompatíveis são metodologias ágeis e modelos de qualidade? Paulk, da equipe que criou o CMM, publicou um artigo em que relaciona a metodologia ágil mais difundida atualmente, XP e o modelo para melhoria de processos CMM, predecessor do CMMI [PAULK 2001]. Analisando como XP se comporta em relação ao modelo, ele relata que XP é um processo documentado e disciplinado e que atende a vários processos definidos pelo modelo CMM. Como conclusão, Paulk ressalta que:

- XP foca no trabalho técnico e o CMM nas questões de gerenciamento;
- XP não aborda questões relacionadas à institucionalização das práticas em uma organização, o que é considerado crucial para o CMM;
- XP aborda, até um determinado limite, questões de gerenciamento e acompanhamento de projetos.

Como seria o resultado deste mapeamento se Paulk tivesse se focado numa metodologia ágil voltada para o gerenciamento, como Scrum? Seriam Scrum e CMMI completamente incompatíveis?

Modelos de qualidade e metodologias ágeis talvez possam coexistir na mesma organização. O CMMI pode servir como orientação para dizer quais processos e práticas de gerencia de projetos devem estar presentes e Scrum para dizer como esses processos devem ser executados. Uma abordagem similar é seguida nos diversos estudos, análises e mapeamentos entre outras metodologias ágeis e modelos de qualidade, como em [NAWROCKI et al. 2001], [PAULK 2001], [TURNER et al. 2002], [CAMPELO 2003], [KÄHKÖNEN et al. 2004], [ANDERSON 2005] e [MARÇAL 2007].

Assim, este trabalho tem como objetivo responder as seguintes perguntas:

- Quanto aderente é Scrum à área de gerência de projetos do CMMI? Quais práticas e metas do CMMI são atendidas pelas atividades e práticas do Scrum e de outras metodologias ágeis? Quais não são?
- Scrum e CMMI podem conviver numa mesma organização? Se sim, até que nível de maturidade? Se não, em quais áreas e por que não?
- Scrum precisa ser adaptado para torná-lo compatível com a certificação CMMI na área de gerência de projetos? Se sim, quais adaptações são necessárias e como adaptá-lo sem torná-lo um processo “pesado e engessado” e sem perder sua essência ágil? Existem outras metodologias e processos ágeis que resolvem as deficiências de Scrum?

Para fazer isso, primeiramente será realizado um diagnóstico de satisfação dos processos de gerenciamento de projetos do CMMI por Scrum, relacionando as práticas atendidas e não atendidas. Este diagnóstico será agrupado por área de processo e nível de maturidade. Caso o diagnóstico indique que algumas práticas não são atendidas ou são parcialmente atendidas, processos e práticas de outras metodologias e processos ágeis serão adaptadas às atividades de Scrum e propostas como extensões. Com isso, Scrum fica aderente ao CMMI e o seu uso não compromete organizações que desejam a certificação do SEI.

Portanto, seguindo esta metodologia e respondendo a estas perguntas, este trabalho busca:

- Mostrar como uma empresa que segue as práticas e princípios de Scrum pode se adequar às normas de certificação de processo do CMMI na área de gerenciamento de projetos;
- Mostrar como organizações que têm seus processos de gerência baseados no CMMI podem modificá-los através de uma abordagem ágil mantendo a certificação e as melhorias de processo introduzidas;
- Servir como ponto de partida e guia para empresas que planejam definir um processo de desenvolvimento de software baseados tanto no CMMI quanto nas práticas do Scrum.

### 1.3 Estrutura do Trabalho

---

Este trabalho é organizado da seguinte forma:

- O Capítulo 2 descreve o *Capability Maturity Model Integrated* (CMMI) em detalhes;
- O Capítulo 3 descreve Scrum, uma metodologia ágil focada na área de gerenciamento de projetos;

- O Capítulo 4 apresenta o mapeamento das práticas de Scrum nas áreas de processo do CMMI;
- O Capítulo 5 apresenta as extensões propostas para tornar Scrum aderente ao CMMI; e
- O Capítulo 6 apresenta as conclusões deste trabalho.

## 2. CMMI

---

Nos últimos anos, o *Capability Maturity Model Integration* (CMMI) tem sido amplamente utilizado para avaliar a maturidade organizacional e a capacidade dos processos das empresas por todo o mundo [ZUBROW 2003].

Este capítulo introduz o CMMI e destaca alguns aspectos conforme as seções abaixo:

- Seção 2.1 – Visão geral e objetivos: descreve uma visão geral do CMMI, com seus princípios, objetivos e metas;
- Seção 2.2 – Representações: descreve as duas representações do CMMI: contínua e estagiada;
- Seção 2.3 – Áreas de processo e seus componentes: descreve as áreas de processo, práticas relacionadas, que, quando implementadas coletivamente, satisfazem os objetivos considerados importantes de determinada área;
- Seção 2.3 – Áreas de processo e seus componentes: descreve as áreas de processo, práticas relacionadas, que, quando implementadas coletivamente, satisfazem os objetivos considerados importantes de determinada área;
- Seção 2.4 – Níveis: descreve os níveis que as empresas são avaliadas no CMMI e seus requisitos;
- Seção 2.5 – Categorias: descreve as categorias em que as áreas de processo são agrupadas, focando na área de gerencia de projetos; e
- Seção 2.6 – Conclusão do capítulo: sumariza o capítulo e resume o que do CMMI será o foco deste trabalho.

### 2.1 Visão geral e objetivos

---

Aumentando-se a qualidade de um processo, espera-se aumentar também a qualidade dos produtos gerados através desse processo. Com isso em mente, várias abordagens e modelos de melhoria de processos de software foram surgindo [SEI 2007]. No mercado atual, existem modelos de maturidade, padrões, metodologias e recomendações de boas práticas que permitem que uma organização melhore a maneira como conduz seus negócios. No entanto, a maior parte das alternativas de melhoria focam uma parte específica do negócio e não levam em consideração uma abordagem sistêmica dos problemas encontrados pela maioria das organizações [SEI 2006]. Ao se focar apenas em uma das áreas, estes modelos ajudam a perpetuar as barreiras e impedimentos que existem nas organizações.

Segundo o SEI, o CMMI (Capability Maturity Model Integration) provê uma oportunidade de evitar ou eliminar essas barreiras, através de modelos integrados que transcendem as disciplinas [SEI 2007]. O CMMI consiste em boas práticas de atividades de desenvolvimento e manutenção aplicadas em produtos e serviços. Ele aborda práticas

que vão desde a concepção do produto até a entrega e manutenção, com o propósito de ajudar organizações a construir e manter seus produtos durante todo o seu ciclo de vida.

## 2.2 Representações

O CMMI pode ser visto sob duas representações: contínua ou estagiada. A representação contínua permite que uma organização melhore seus processos em ritmos diferentes. Na representação estagiada, o foco é na melhoria dos processos organizacionais como um todo.

A representação contínua é indicada para as empresas que querem focar seus esforços de melhoria numa área específica, conhecida previamente. Ela também é a representação indicada para as empresas que desejam migrar vindas do padrão Electronic Industries Alliance Interim Standard (EIA/IS) [MENEZES 2002]. Nos melhoramentos seguindo esta representação, no entanto, é preciso atenção às interdependências entre as áreas de processo, já que as áreas não são completamente independentes. [SEI 2006].

A representação estagiada é indicada para empresas que desejam melhorar seus processos seguindo agrupamentos predefinidos com coerência e testados pelo mercado. Ela apresenta uma trilha a ser seguida na melhoria de processos, com níveis consecutivos. Nesta representação, no entanto, é preciso cuidado para que atingir o nível seguinte não se torne o único objetivo da organização [DEMARCO 1998]. Nestes casos, os projetos mais arriscados são frequentemente deixados para depois. São justamente os projetos que trazem as maiores inovações e riscos que costumam trazer os maiores retornos.

A tabela a seguir resume as vantagens de cada modelo.

**Tabela 2.1 - Comparação entre as vantagens do modelo contínuo e estagiado**

| Representação contínua                                                                                                                                                    | Representação estagiada                                                                                                    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| Permite liberdade ao escolher a ordem de melhoria dos processos que mais atendem aos objetivos de negócio da organização e mitigar os riscos das áreas mais problemáticas | Permite que a organização tenha um caminho de melhoria predefinido e testado                                               |
| Permite maior visibilidade das capacidades atingidas em cada área de processo                                                                                             | Foca em conjuntos de processos que provêm a organização das capacidades que são categorizadas com os níveis de maturidade. |
| Permite que melhoramentos em diferentes processos sejam feitos em diferentes velocidades                                                                                  | Sumariza os resultados das melhorias de processo de uma maneira simples – um número com o nível de maturidade              |
| Reflete uma opção que ainda não tem dados suficientes para demonstrar qual retorno de investimento proporciona                                                            | É baseada em longa história de uso que inclui estudos de caso e dados que demonstram retorno no investimento               |

## 2.3 Áreas de processo e seus componentes

---

O CMMI está agrupado em 22 áreas de processo. Áreas de processo são um conjunto de práticas relacionadas, que, quando implementadas coletivamente, satisfazem os objetivos considerados importantes para obter melhoramentos naquela área [SEI 2006]. A cada uma das áreas de processo estão associadas metas genéricas e específicas, componentes considerados obrigatórios para analisar a satisfação da área por uma organização.

As metas genéricas têm este nome porque estão relacionadas a mais de uma área de processo. Elas contêm as características que são necessárias para considerar uma área de processo institucionalizada. Um exemplo de meta genérica é “O processo é institucionalizado como um processo definido.” Este documento não está relacionado a uma única organização, portanto, as metas genéricas e seus sub-componentes, não serão aqui considerados durante a análise de satisfação do CMMI. No caso de uma implementação real das soluções aqui propostas, cabe a cada organização avaliar os aspectos relacionados à institucionalização dos processos.

Já as metas específicas, por sua vez, descrevem as características únicas daquela área de processo e são necessárias para considerar a área satisfeita. Um exemplo de meta específica é “A integridade das *baselines* é estabelecida e mantida”. As metas específicas dividem-se em práticas específicas.

Práticas específicas é a descrição de uma atividade que é considerada importante para atingir determinada meta específica. As práticas específicas de uma meta, quando implementadas conjuntamente, resultam na satisfação da meta específica. Elas são consideradas componentes esperados durante uma avaliação e evidências de satisfação das práticas são coletadas. As práticas específicas dividem-se em sub-práticas.

Sub-práticas são componentes informativos que têm o objetivo de auxiliar o entendimento da prática a que estão associadas. Elas devem ser consideradas apenas para prover idéias que serão úteis na melhoria dos processos.

## 2.4 Níveis

---

A cada uma das representações estão associados níveis que caracterizam melhorias vindas de um estado em que os processos estão mal definidos e executados para um estado que usa dados e informações quantitativas para determinar e gerenciar as melhorias necessárias para atingir os objetivos de negócio da organização.

Para definir o nível de capacidade, relacionados com a representação contínua, as áreas de processo são avaliadas em um número de 0 a 5, seguindo a nomenclatura e regras a seguir:

0. **Incompleto:** uma ou mais das metas específicas da área de processo não são executadas.
1. **Executado:** as metas específicas da área de processo são executadas, mas o processo ainda não está institucionalizado (i.e. as metas genéricas não são executadas).
2. **Gerenciado:** é uma área de processo do nível executado que contem a infraestrutura para garantir o suporte e cumprimento do processo. (i.e. as metas genéricas são executadas).
3. **Definido:** é uma área de processo gerenciada que teve seus processos e planos baseados em um padrão organizacional. Além disso, ela deve contribuir com métricas e outras informações para a modificação e melhoria do padrão.
4. **Gerenciado quantitativamente:** é uma área de processo definida que é controlada usando métodos estatísticos e outras técnicas quantitativas.
5. **Melhorado:** é uma área de processo gerenciada quantitativamente que é continuamente melhorada baseada no entendimento das causas de variação.

Para definir o nível de maturidade de uma organização, as áreas de processo são avaliadas e o diagnóstico é dado seguindo a nomenclatura e regras a seguir:

1. **Inicial:** se a organização não cumpre os requisitos do nível de maturidade 2.
2. **Gerenciado:** se as áreas de processo associadas ao nível de maturidade 2 estão no nível de capacidade 2.
3. **Definido:** se as áreas de processo associadas ao nível de maturidade 2 e 3 estão no nível de capacidade 3.
4. **Gerenciado quantitativamente:** se as áreas de processo associadas ao nível de maturidade 2, 3 e 4 estão no nível de capacidade 3 e uma ou mais áreas selecionadas pela organização a ser certificada estão no nível de capacidade 4.
5. **Melhorado:** se todas as áreas de processo estão no nível de capacidade 3 e uma ou mais áreas selecionadas pela organização a ser certificada estão no nível de capacidade 5.

A tabela a seguir lista todas as áreas de processo, suas siglas e o nível de maturidade associado a cada uma delas.

**Tabela 2.2 – Áreas de processo do CMMI**

| Sigla | Nome                                     | Nível de maturidade |
|-------|------------------------------------------|---------------------|
| CAR   | Causal Analysis and Resolution           | 5                   |
| CM    | Configuration Management                 | 2                   |
| DAR   | Decision Analysis and Resolution         | 3                   |
| IPM   | Integrated Project Management +IPPD      | 3                   |
| MA    | Measurement and Analysis                 | 2                   |
| OID   | Organizational Innovation and Deployment | 5                   |
| OPD   | Organizational Process Definition+IPPD   | 3                   |
| OPF   | Organizational Process Focus             | 3                   |
| OPP   | Organizational Process Performance       | 4                   |
| OT    | Organizational Training                  | 3                   |
| PI    | Product Integration                      | 3                   |
| PMC   | Project Monitoring and Control           | 2                   |
| PP    | Project Planning                         | 2                   |
| PPQA  | Process and Product Quality Assurance    | 2                   |
| QPM   | Quantitative Project Management          | 4                   |
| RD    | Requirements Development                 | 3                   |
| RM    | Requirements Management                  | 2                   |
| RSKM  | Risk Management                          | 3                   |
| SAM   | Supplier Agreement Management            | 2                   |
| TS    | Technical Solution                       | 3                   |
| VAL   | Validation                               | 3                   |
| VER   | Verification                             | 3                   |

## 2.5 Categorias

As áreas de processo do CMMI estão intimamente relacionadas entre si. Para facilitar a compreensão das dependências e do fluxo de informação, elas são agrupadas em quatro categorias:

- Gerenciamento de Processos;
- Gerenciamento de Projetos;
- Engenharia; e
- Suporte

Neste trabalho, o foco está em prover soluções para atender aos requisitos do CMMI na área de gerência de projetos. Dentre as quatro categorias do CMMI, a categoria de Gerenciamento de Projetos foca especificamente nesta área. Seu escopo corresponde às atividades relacionadas a estabelecer e manter o plano do projeto, estabelecer e manter compromissos, monitorar o progresso comparando-o ao plano, tomar ações corretivas, gerenciar acordos com fornecedores, estabelecer um processo definido adaptado dos processos organizacionais, estabelecer o ambiente de trabalho adaptado dos padrões organizacionais, coordenar e colaborar com os *stakeholders* envolvidos no projeto, formar e manter times integrados para a condução de projetos e gerenciar

quantitativamente o processo do projeto. Assim, as seguintes áreas de processo fazem parte desta categoria:

- Planejamento de Projeto (PP)
- Monitoramento e Controle do Projeto (PMC)
- Gerenciamento Integrado de Projetos (IPM) + IPPD
- Gerenciamento de Riscos (RSKM)
- Gerenciamento de Acordos com Fornecedores (SAM)
- Gerenciamento Quantitativo do Projeto (QPM)

Estas áreas de processo também podem ser agrupadas de acordo com o nível de maturidade em que estão presentes na representação estagiada:

- Nível 2: PP, PMC e SAM
- Nível 3: IPM+IPPD e RSKM
- Nível 4: QPM

## 2.6 Conclusão do capítulo

---

Neste capítulo foi descrito o CMMI, modelo de qualidade escolhido neste trabalho. Seus objetivos, representações, níveis, categorias e áreas de processo foram observados.

Neste trabalho, o objetivo está em prover soluções para atender aos requisitos do CMMI no gerenciamento do projeto. Esta análise é independente de uma organização específica, portanto não considera aspectos de institucionalização. Assim, apenas metas e práticas específicas das áreas de processo da categoria de Gerenciamento de Projetos foram escolhidas para um mapeamento no decorrer deste trabalho.

### 3. Scrum

---

Scrum foi descrito por Ken Schwaber em 1996 [SCHWABER 1996] como um processo que aceita a imprevisibilidade do desenvolvimento de software e a contorna através da adaptação constante, atingindo sucesso com numerosos desenvolvedores. Scrum se destaca das demais metodologias ágeis por dar mais enfoque a área de gerenciamento. O termo Scrum tem origem no esporte conhecido como Rugby [RISING 2007]. Neste esporte, “Scrum” ocorre quando jogadores de cada time colaboram entre si numa tentativa de avançar juntos pelo campo adversário. [HIGHSMITH 2002].

Este capítulo foca na metodologia de Scrum e está dividido conforme as seções a seguir:

- Seção 3.1 – Visão geral e objetivos: descreve uma visão geral do Scrum, com seus princípios, objetivos e metas;
- Seção 3.2 – Papéis e responsabilidades: descreve os papéis presentes em Scrum e suas obrigações;
- Seção 3.3 – Ciclo de vida: descreve as fases do processo de desenvolvimento usando Scrum;
- Seção 3.4 – Fluxo de desenvolvimento: descreve o processo e o fluxo de desenvolvimento em Scrum;
- Seção 3.5 – Sprints: descreve o processo e o fluxo de desenvolvimento dentro de cada iteração;
- Seção 3.6 – Múltiplos times em um único projeto: descreve como Scrum trata projetos maiores, com muitos desenvolvedores.
- Seção 3.7 – Conclusão do capítulo: sumariza o capítulo, resumindo o que foi visto de Scrum.

#### 3.1 Visão geral

---

Nas engenharias, muitos dos processos funcionam porque o grau de imprecisão obtido com eles é aceitável. Em muitos destes casos, esta precisão é limitada por nossos sentidos. Ao construir uma cadeira, por exemplo, eu só preciso fazê-la aceitável aos olhos humanos. Nos casos em que o produto não está com a precisão adequada, aumentamos os detalhes do processo, garantindo que os passos determinem a qualidade do produto. Criar e controlar um processo, que repetidamente produz um produto de qualidade, é chamado de “controle de processos definidos” [OGUNNAIKE et al. 1992]. Usam-se processos definidos sempre que possível, com o objetivo de conseguir controlar a produção para níveis de qualidade aceitáveis.

De acordo com seu criador [SCHWABER 2004], Scrum foi criado a partir do reconhecimento que desenvolvimento de software é muito complexo para ser planejado corretamente desde o início. Baseando-se nos trabalhos de modelagem de processos de Ogunnaike [OGUNNAIKE et al. 1992], Schwaber argumenta que quando o controle de processos definidos não é aplicável, devido a complexidade das atividades intermediárias

e ao alto custo ao corrigir produtos com defeito, processos empíricos devem ser aplicados.

Processos empíricos baseiam-se na visibilidade dos aspectos e resultados do processo e em prover maneiras de inspecionar e corrigir o processo e o produto. Dessa forma, processos empíricos aceitam as falhas como consequência natural da produção e tentam torná-las visíveis e passíveis de correção o mais cedo possível. Assim, argumenta Schwaber, a abordagem empírica é ideal para o ambiente de desenvolvimento de software, onde funcionalidades consideradas “terminadas” muitas vezes não estão suficientemente testadas e aceitas pelo cliente, onde os passos a serem seguidos na criação de um produto só podem ser descritos em alto-nível (como “elicitare requisitos”) e onde correções, seguindo um processo definido, têm um custo inaceitável (“modificar os documentos de requisitos, arquitetura e testes além do código-fonte”).

Seguindo a abordagem empírica, Scrum reúne atividades de monitoramento e *feedback*, em geral, principalmente através de reuniões rápidas e diárias com toda a equipe. Tais reuniões visam a identificação e correção de quaisquer deficiências e/ou impedimentos no processo de desenvolvimento. Além disso, Scrum recomenda equipes pequenas (no máximo, 7 pessoas) supõe que os requisitos são pouco estáveis ou desconhecidos e que o desenvolvimento deve ser feito em iterações curtas, de no máximo 30 dias, também chamadas de *Sprints*. [SCHWABER et al. 2002]

### 3.2 Papéis e responsabilidades

---

Uma das principais idéias do Scrum para atacar a complexidade do desenvolvimento e gerenciamento de software é a implantação de um controle descentralizado, capaz de lidar mais eficientemente com contextos pouco previsíveis [SCHWABER 2004]. Para tanto, o gerenciamento é distribuído através de três agentes independentes [SCHWABER et al. 2002]:

- **Product Owner:** É o responsável pelo retorno sobre o investimento do projeto. Seu papel é garantir que o produto entregue atenda aos anseios do patrocinador do projeto e priorizar quais funcionalidades devem ser entregues e quais agregam mais valor ao projeto.
- **Time:** São todos os integrantes (desenvolvedores, designers, testadores, arquitetos) comprometidos em desenvolver o produto alvo do projeto, de acordo com suas próprias decisões para alcançar os objetivos estabelecidos;
- **Scrum Master:** É responsável por guiar o Time, intermediar negociações entre o *Product Owner* e a equipe e pelo processo de desenvolvimento do produto. Seu principal papel é ensinar e acompanhar a utilização do Scrum.

As pessoas que preenchem estes papéis estão diretamente comprometidas e são responsáveis pelo projeto. Scrum faz uma clara distinção entre os que são responsáveis, chamados de *Pigs*, e os que estão apenas envolvidos, como outros *stakeholders*, clientes e executivos, chamados de *Chickens* [SCHWABER 2004]. Esta distinção é importante para

Scrum, por prover visibilidade e responsabilidade sobre o projeto, deixando claras as obrigações e deveres dos envolvidos. Da mesma maneira que atribui responsabilidade para os *Pigs*, Scrum estabelece que todos os *Chickens* não atrapalhem o Time. Assim, todo e qualquer interesse de pessoas que não façam parte da equipe responsável deve ser filtrado e levado em consideração apenas pelo Product Owner e pelo Scrum Master, deixando o Time livre para prosseguir com o projeto.

### 3.3 Ciclo de vida

---

Scrum possui um ciclo de vida composto por 4 fases [LARMAN 2004]:

- **Planejamento:** Nesta fase se estabelece a visão do projeto e as expectativas dos envolvidos, garantindo recursos para sua execução. São criadas, também, as versões iniciais do *Product Backlog*, o plano de releases, a arquitetura de negócio e a arquitetura técnica em alto nível;
- **Preparação:** Período em que se avalia as várias dimensões do projeto, criando itens adicionais ao *Product Backlog* relacionados com o tipo do sistema, time, ambiente de desenvolvimento e tipo de aplicação. Os times são formados e são construídos os mecanismos de comunicação e coordenação entre eles;
- **Desenvolvimento:** Nela ocorre o desenvolvimento iterativo e incremental das funcionalidades do produto através de múltiplas *Sprints*; e
- **Entrega:** Realiza-se a entrega final do produto ao cliente e suas atividades relacionadas, incluindo treinamento para o usuário final.

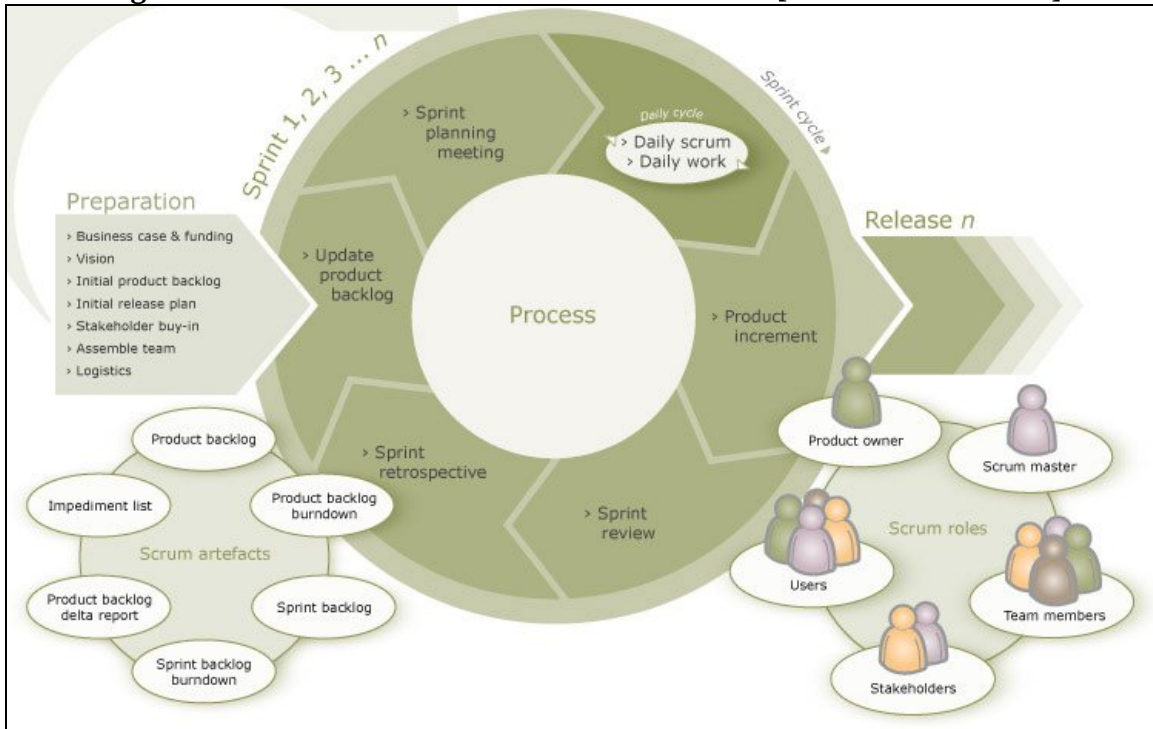
As duas primeiras fases preparam a infra-estrutura e os recursos necessários para o projeto. Na terceira, onde existe a maior alocação de pessoas e recursos, é feito o desenvolvimento do projeto. Por fim, na fase de menor duração é feita a entrega do produto e é obtida a aceitação final.

### 3.4 Fluxo de desenvolvimento

---

O desenvolvimento em Scrum se inicia com uma visão do produto a ser desenvolvido, contendo suas características, premissas e restrições. [SCHWABER 2004]. Em seguida, o *Product Backlog* é criado, contendo os requisitos a serem desenvolvidos em ordem de prioridade. Este documento servirá como entrada para o processo iterativo e incremental de desenvolvimento, como ilustra a figura a seguir:

**Figura 3.2 – Fluxo de desenvolvimento do Scrum [COCHANGO 2006]**



O processo de desenvolvimento se inicia com uma reunião de planejamento (*Sprint Planning Meeting*), na qual o *Product Owner* e o Time decidem em conjunto o que deverá ser implementado do *Product Backlog*. Esta reunião é dividida em duas partes.

Na primeira parte (*Sprint Planning 1*), o *Product Owner* apresenta os requisitos de maior valor e prioriza aqueles que devem ser implementados. O Time então define colaborativamente o que poderá entrar no desenvolvimento da próxima *Sprint*, considerando sua capacidade de produção.

Na segunda parte (*Sprint Planning 2*), o Time planeja seu trabalho, definindo o *Sprint Backlog*, que são as tarefas necessárias para implementar as funcionalidades selecionadas no *Product Backlog*. A lista de tarefas pode ser modificada ao longo da *Sprint* pelo Time e recomenda-se que as tarefas sejam divididas de forma a levar no máximo 16 horas para a sua conclusão.

Depois da reunião de planejamento, o Time realiza o desenvolvimento do produto durante 30 dias (*Sprint*). As atividades da *Sprint* serão melhor detalhadas na seção a seguir.

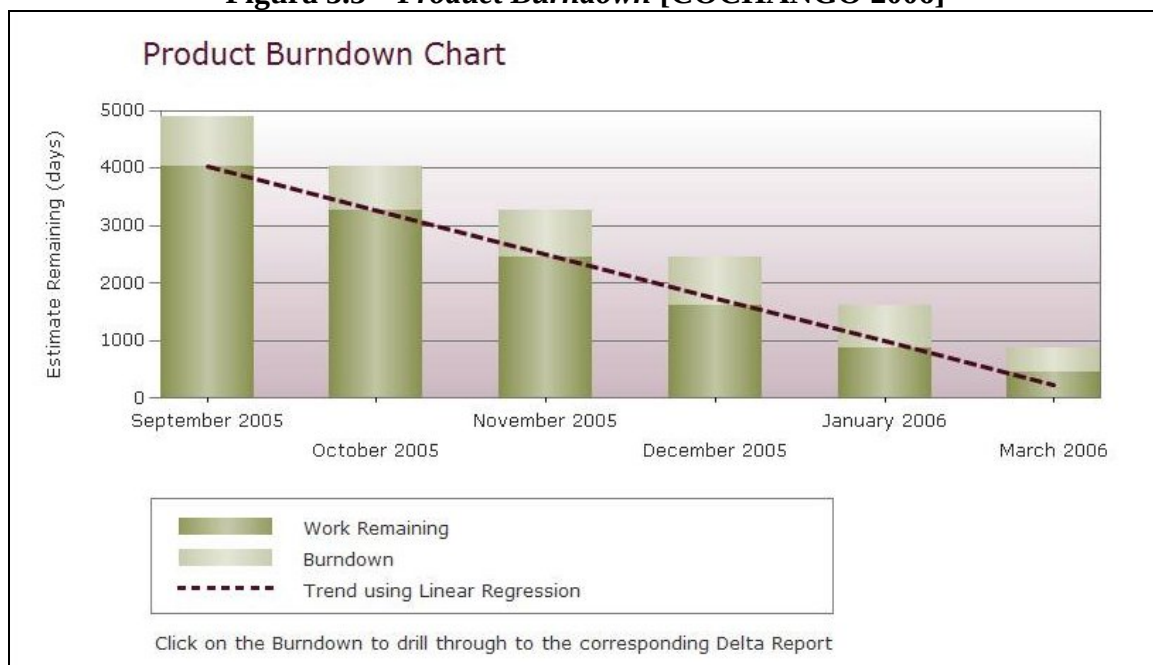
Vale ressaltar que durante a *Sprint* nenhuma interferência por parte do *Product Owner* ou dos outros *stakeholders* é permitida no *Sprint Backlog*. Mudanças, adições e priorização de requisitos só podem ser feitas por parte do *Product Owner* durante as reunião de planejamento. No entanto, caso mudanças no ambiente de negócios forcem mudanças nos requisitos, o *Product Owner*, juntamente com o *ScrumMaster*, pode cancelar a *Sprint*

atual imediatamente (*Abort the Sprint*) e realizar uma nova reunião de planejamento (*Sprint Planning Meeting*). Com isso, Scrum mantém a visibilidade da produtividade realizada pelo Time, que seria seriamente prejudicada caso requisitos pudessem ser modificados e adicionados durante um *Sprint*, ao mesmo tempo que provê um mecanismo de adaptação e correção para os requisitos do produto colocados pelo *Product Owner*.

Ao final da *Sprint*, é realizada a reunião de retrospectiva (*Sprint Retrospective Meeting*), com o objetivo de melhorar o processo, as capacidades e aptidões do Time e o desenvolvimento do produto para a próxima *Sprint*.

O monitoramento do progresso do projeto é realizado através de dois gráficos principais: *Product Burndown* e *Sprint Burndown*. Estes gráficos mostram ao longo do tempo a quantidade de trabalho que ainda resta para realizar para o produto e a iteração, respectivamente. Com isso, observa-se a quantidade de trabalho que falta ser feita (em qualquer ponto) e o progresso do Time do projeto em reduzir este trabalho. A figura a seguir ilustra um gráfico de *Product Burndown*.

**Figura 3.3 – Product Burndown [COCHANGO 2006]**



### 3.5 Sprints

Na execução das *Sprints*, diariamente o Time faz uma reunião de 15 minutos para acompanhar o progresso do trabalho e agendar outras reuniões necessárias. Na reunião diária (*Daily Scrum Meeting*), cada membro do Time responde a três perguntas básicas:

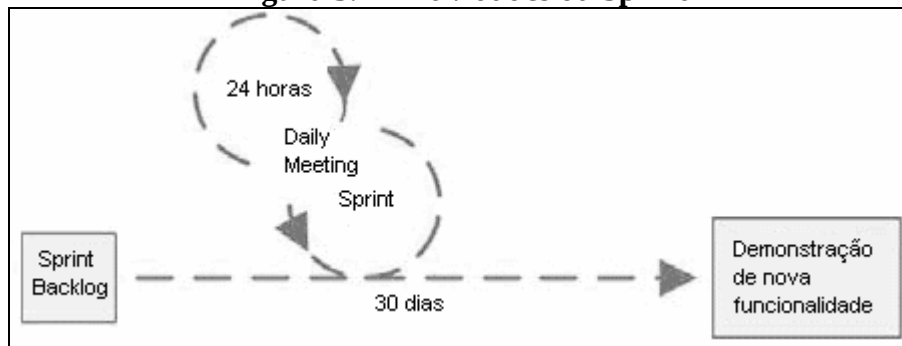
- O que eu fiz no projeto desde a última reunião?

- O que irei fazer até a próxima reunião?
- Quais são os impedimentos e dificuldades encontradas?

Com estas três perguntas busca-se a visibilidade das atividades desenvolvidas pelo Time e o levantamento de problemas relacionados ao desenvolvimento. Os impedimentos são então registrados no *Impediments Backlog*, um artefato que registra todos os entraves encontrados pelo Time. É responsabilidade do Scrum Master trabalhar para resolver estes impedimentos e cuidar para que o Time consiga desenvolver com a máxima eficiência.

Ao final da *Sprint* é realizada a reunião de revisão (*Sprint Review Meeting*) para que o Time apresente o resultado alcançado na iteração ao *Product Owner*. Neste momento as funcionalidades são inspecionadas e adaptações do projeto podem ser realizadas. As atividades da *Sprint* são sumarizadas na figura a seguir.

**Figura 3.4 – Atividades da Sprint**



### 3.6 Múltiplos times em um único projeto

Muitos projetos demandam mais esforço do que o que um único time pode prover. Nestes casos são necessários múltiplos times e mecanismos de coordenação e cooperação entre eles. Em Scrum, quando diversos times trabalham juntos num único projeto, este projeto é chamado de *Scaled Project* (projeto escalonado) [SCHWABER 2004].

Primeiramente, um único Time é criado durante a fase de Preparação do Scrum com o objetivo de desenvolver a infra-estrutura e a arquitetura necessárias para a coordenação. No entanto, mesmo durante o desenvolvimento da arquitetura, este Time deve manter o foco em prover valor de negócio para o cliente. Assim, o desenvolvimento da arquitetura deve ser feito, por exemplo, através da implementação de uma funcionalidade simples que utilize todas as camadas do sistema. No final do desenvolvimento da arquitetura, os integrantes deste Time inicial são divididos e farão parte dos diversos Times, com o objetivo de prover uma visão sistêmica do projeto e guiar os Times individuais.

Após o *Daily Meeting* de cada Time individual é feita uma reunião com representantes de cada Time (*Daily Scrum of Scrums*) também de 15 minutos com os mesmos objetivos, formato e atividades da *Daily Meeting*. A única diferença é que, ao invés de responder

perguntas sobre as atividades e impedimentos individuais, cada participante da *Daily Scrum of Scrums* vai relatar as atividades e os impedimentos do seu Time. Com isso, cria-se um mecanismo ágil de coordenação entre Times.

### 3.7 Conclusão do capítulo

---

Neste capítulo foi descrito o Scrum, processo ágil escolhido neste trabalho. Foram observadas suas premissas, fases, atividades e etapas, além dos papéis presentes e suas responsabilidades. Levando as práticas do Scrum em consideração, será feito a seguir um mapeamento das áreas de processo de gerenciamento do CMMI nestas práticas.

## 4. Mapeando Scrum nas áreas de processo de gerenciamento do CMMI

---

Como já explicado no capítulo do CMMI, os componentes para atender a uma avaliação são as metas, práticas e sub-práticas. Apesar das práticas não serem consideradas obrigatórias para o alcance dos níveis de maturidade e capacidade, as práticas de uma meta são, em geral, o que determinam ou não a satisfação da mesma. Devido à importância das práticas para o entendimento do CMMI, uma análise da satisfação em relação às práticas será realizada.

Neste mapeamento considera-se que uma prática do CMMI pode ser “Satisfeita”, “Parcialmente satisfeita” ou “Não satisfeita”, como sumariza a tabela abaixo:

**Tabela 4.1 - Pontuações do mapeamento e seus significados**

| Pontuação               | Significado                                          |
|-------------------------|------------------------------------------------------|
| Satisfeita              | Scrum atende completamente a prática do CMMI         |
| Parcialmente satisfeita | Scrum atende de forma parcial a prática do CMMI      |
| Não satisfeita          | Não há evidências da satisfação da prática por Scrum |

Realizar esta avaliação da satisfação de cada uma das áreas de processo requer julgamento por parte do avaliador. Em avaliações de certificação do CMMI, este julgamento é realizado por profissionais competentes, treinados e experientes que definem processos. Boa parte destas práticas do CMMI, no entanto, exigem que a equipe de avaliação discuta se está acontecendo a implementação da prática até que se chegue a um consenso. Assim como avaliações oficiais, qualquer diagnóstico é resultado de algumas interpretações e julgamentos, de forma que este trabalho pode diferir de outros diagnósticos, realizados por outras pessoas. No entanto, as premissas utilizadas para determinar a satisfação das práticas estão documentadas, entre outras razões, como forma de justificar o resultado obtido.

Neste capítulo, as seções 4.1 a 4.6 correspondem a cada uma das áreas de processo do CMMI que serão analisadas. Por fim, a seção 4.7 apresenta a conclusão do capítulo e análise sumarizada dos resultados.

### 4.1 Planejamento de Projeto (PP)

---

De acordo com o SEI, o propósito da área de processo de Planejamento de Projeto (PP) é estabelecer e manter os planos que definem as atividades do projeto. [SEI 2006]. Dessa forma, PP envolve criar um plano para o projeto, conseguir apoio e comprometimento a ele e interagir com todos os *stakeholders* de forma apropriada. A seguir, descreveremos como Scrum está relacionado a cada uma das 14 práticas de PP, resumindo os resultados na tabela 4.2.

### SP 1.1 - Estimar o Escopo do Projeto

- **Análise:** Em Scrum, o *Product Backlog* e o documento de visão, criados durante a fase de planejamento, provêm a definição do escopo do projeto. Com este escopo inicial, é possível estimar a quantidade de iterações e o tempo de desenvolvimento necessário para o desenvolvimento do produto.
- **Diagnóstico:** Satisfeita.

### SP 1.2 - Estabelecer Estimativas de Atributos de Produtos de Trabalho e Tarefas

- **Análise:** Em Scrum, não há orientações para medir o tamanho e a complexidade dos produtos e atividades relacionadas ao projeto. Assim, os atributos dos produtos acabam por não serem estimados.
- **Diagnóstico:** Não satisfeita.
- **Observações:** O maior problema da não satisfação da prática 1.2 é que Scrum permite que as estimativas de esforço não sejam derivadas de um entendimento comum por parte dos *stakeholders* dos atributos das tarefas e produtos. Para tornar-se aderente a esta prática, Scrum deveria estimular o entendimento das atividades através de um método formal de estimativa em que os desenvolvedores e outros *stakeholders* envolvidos pudessem discutir os atributos do que deve ser feito. Assim, conseguir-se-ia melhorar o entendimento dos requisitos e, conseqüentemente, aprimorar as estimativas.

### SP 1.3 – Definir o Ciclo de Vida do Projeto

- **Análise:** Em Scrum, o projeto tem um ciclo de vida definido em 4 fases: Planejamento, Preparação e escalonamento, Desenvolvimento e Entrega, conforme descrito na seção 3.3.
- **Diagnóstico:** Satisfeita.

### SP 1.4 – Determinar Estimativas de Esforço e Custo

- **Análise:** As estimativas de esforço em Scrum são realizadas em 2 níveis de detalhamento. As estimativas do *Product Backlog* são pouco precisas e têm como objetivo oferecer uma primeira visão para o *Product Owner* do esforço necessário para cada requisito. Já as estimativas do *Sprint Backlog* são mais precisas, são feitas normalmente em horas, e têm como objetivo prover controle e visibilidade do desenvolvimento das atividades. Como já observado, não há recomendações de métodos formais, como o uso de pontos de função, para a realização de estimativas no Scrum. Além disso, Scrum não realiza estimativas para os custos dos produtos.
- **Diagnóstico:** Parcialmente satisfeita.

### SP 2.1 – Estabelecer o Orçamento e o Cronograma

- **Análise:** Com as estimativas de esforço do Product Backlog e uma idéia da produtividade da equipe, estabelece-se um primeiro cronograma do projeto através da divisão em Sprints de trinta dias. Este cronograma é automaticamente atualizado à medida que o projeto se desenvolve e as funcionalidades são repriorizadas. O orçamento do projeto, no entanto, não está contemplado nas atividades do Scrum e nem sempre pode ser calculado diretamente do cronograma.
- **Diagnóstico:** Parcialmente satisfeita.

### SP 2.2 – Identificar os Riscos do Projeto

- **Análise:** No Scrum não há identificação e monitoramento explícito e sistemático dos riscos do projeto, com categorias e planos de mitigação, como sugere o CMMI. No entanto, riscos são possíveis impedimentos, e, portanto, podem ser levantados durante a *Daily Meeting*, e anotados na impediments list, gerando um levantamento iterativo dos riscos.
- **Diagnóstico:** Parcialmente satisfeita.

### SP 2.3 – Planejar o Gerenciamento de Dados

- **Análise:** Esta prática requer que os requisitos de dados e documentos para o projeto sejam planejados, baseados em um padrão comum, para evitar a coleta desnecessária de dados e a escrita desnecessária de documentos. Em Scrum, a maioria das informações do projeto são comunicadas face-a-face nas reuniões diárias e entre *Sprints*, com o restante das informações sendo comunicadas através de documentos ou relatórios de progresso como o *Burndown chart*. É importante ressaltar, no entanto, que não existe um plano de comunicações que formalize como serão realizadas as coletas, consolidação e divulgação das informações e dados do projeto, baseado em um padrão comum utilizado na organização.
- **Diagnóstico:** Não satisfeita.

### SP 2.4 – Planejar os Recursos do Projeto

- **Análise:** No começo do projeto é realizada a alocação do time e da infra-estrutura necessária para todo o desenvolvimento. O *Scrum Master* e o *Product Owner* são responsáveis por garantir os recursos e a continuação do projeto, através das reuniões ao início de cada iteração e da remoção de impedimentos levantados pelo time.
- **Diagnóstico:** Satisfeita.

## SP 2.5 – Planejar os Conhecimentos e Habilidades Necessárias

- **Análise:** O time, que é um grupo multifuncional e auto-gerenciado, deve, na medida do possível, ser criado contendo os conhecimentos e as habilidades (técnicas e de negócio) necessárias para desenvolver o *Product Backlog*. Caso algum treinamento se torne necessário, no entanto, esta necessidade é incluída no *Product Backlog* ou numa lista de impedimentos.
- **Diagnóstico:** Satisfeita.

## SP 2.6 – Planejar o Envolvimento dos Stakeholders

- **Análise:** Considerando o envolvimento dos *stakeholders*, percebe-se que as práticas de Scrum contribuem para garantir e manter uma boa comunicação e colaboração entre o time e os usuários.
- **Diagnóstico:** Satisfeita.

## SP 2.7 - Estabelecer o Plano do Projeto

- **Análise:** Em Scrum, o documento de Visão e o Product Backlog representam o plano em alto-nível do projeto, com os requisitos funcionais e não-funcionais e o porquê do desenvolvimento do projeto.
- **Diagnóstico:** Satisfeita.

## SP 3.1 – Revisar Planos que Afetam o Projeto

- **Análise:** Em Scrum, o Time, *Scrum Master* e *Product Owner* revisam os planos e compromissos no início e no final de cada *Sprint*, durante a reuniões de planejamento (*Sprint Planning*) e revisão (*Sprint Review*).
- **Diagnóstico:** Satisfeita.

## SP 3.2 – Equilibrar Níveis de Trabalho e Recursos

- **Análise:** Durante a reunião de planejamento da Sprint, as atividades a serem desenvolvidas são estimadas e os recursos disponíveis são alocados para os itens mais prioritários, definindo e negociando o escopo da iteração. Além disso, os recursos e o escopo de cada iteração são revisados durante a reunião de retrospectiva (*Sprint Review*).
- **Diagnóstico:** Satisfeita.

### SP 3.3 – Obter Compromissos com o Plano

- **Análise:** O compromisso com o plano é obtido continuamente no início de cada *Sprint* quando as atividades a serem desenvolvidas são negociadas com o *Product Owner* e o time.
- **Diagnóstico:** Satisfeita.

Assim, Scrum atende parcialmente às práticas e metas da área de processo de Planejamento de Projeto, como sumariza a tabela abaixo:

**Tabela 4.2 - Satisfação das práticas de PP**

| Meta                                    | Prática                                                                       | Avaliação               |
|-----------------------------------------|-------------------------------------------------------------------------------|-------------------------|
| SG 1<br>Estabelecer Estimativas         | SP 1.1 Estimar o Escopo do Projeto                                            | Satisfeita              |
|                                         | SP 1.2 Estabelecer Estimativas de Atributos de Produtos de Trabalho e Tarefas | Não satisfeita          |
|                                         | SP 1.3 Definir o Ciclo de Vida do Projeto                                     | Satisfeita              |
|                                         | SP 1.4 Determinar Estimativas de Esforço e Custo                              | Parcialmente satisfeita |
| SG 2<br>Desenvolver um Plano de Projeto | SP 2.1 Estabelecer o Orçamento e o Cronograma                                 | Parcialmente satisfeita |
|                                         | SP 2.2 Identificar os Riscos do Projeto                                       | Parcialmente satisfeita |
|                                         | SP 2.3 Planejar o Gerenciamento de Dados                                      | Não satisfeita          |
|                                         | SP 2.4 Planejar os Recursos do Projeto                                        | Satisfeita              |
|                                         | SP 2.5 Planejar os Conhecimentos e Habilidades Necessárias                    | Satisfeita              |
|                                         | SP 2.6 Planejar o Envolvimento dos Stakeholders                               | Satisfeita              |
|                                         | SP 2.7 Estabelecer o Plano do Projeto                                         | Satisfeita              |
| SG 3 Obter Compromissos com o Plano     | SP 3.1 Revisar Planos que Afetam o Projeto                                    | Satisfeita              |
|                                         | SP 3.2 Equilibrar Níveis de Trabalho e Recursos                               | Satisfeita              |
|                                         | SP 3.3 Obter Compromissos com o Plano                                         | Satisfeita              |

## 4.2 Monitoramento e Controle do Projeto (PMC)

De acordo com o SEI, o propósito da área de processo de Monitoramento e Controle do Projeto (PMC) é prover um entendimento do progresso do projeto possibilitando a tomada de ações corretivas apropriadas quando o desempenho do projeto desviar-se significativamente do planejado. [SEI 2006]. Dessa forma, PMC envolve monitorar o plano do projeto, comunicar status e tomar ações corretivas. PMC está dividida em 10

práticas e 2 metas. O mapeamento destas práticas e meta é apresentado abaixo e os resultados são sumarizados na tabela 4.3.

### **SP 1.1 - Monitorar os Parâmetros de Planejamento do Projeto**

- **Análise:** Em Scrum, o *Product Burndown* e as reuniões diárias (*Daily Meeting*) provêm a visibilidade necessária para monitorar o andamento do projeto e conferir desvios no plano. O *Sprint Burndown* também contribui para isso, mostrando diariamente a velocidade do time e se as funcionalidades que vão ser entregues ao final do período.
- **Diagnóstico:** Satisfeita.

### **SP 1.2 - Monitorar os Compromissos**

- **Análise:** Na reunião de planejamento da Sprint (*Sprint Planning Meeting*) são estabelecidos os compromissos da iteração. Durante o Sprint, as reuniões diárias (*Daily Meeting*) e o *Sprint Burndown* provêm maneiras de monitorar o cumprimento dos compromissos. Por fim, o *Sprint Review* mostra os compromissos assumidos e realizados, possibilitando uma maneira de avaliar o time e descobrir os compromissos que não foram cumpridos.
- **Diagnóstico:** Satisfeita.

### **SP 1.3 - Monitorar os Riscos do Projeto**

- **Análise:** Em Scrum os riscos e impedimentos são levantados nas reuniões diárias (*Daily Meeting*) e vão fazer parte do *Sprint Backlog*, *Impediments Backlog* e/ou serem resolvidos em reuniões posteriores. No entanto, os riscos não são categorizados e planos de mitigação e contingência não são elaborados. Além disso, os riscos são registrados de maneira informal, em quadros brancos, cartões ou na *Impediments Backlog*.
- **Diagnóstico:** Parcialmente satisfeita.

### **SP 1.4 - Monitorar o Gerenciamento de Dados**

- **Análise:** Esta prática tem por objetivo garantir que os dados identificados na prática 2.3 da área de processo Planejamento de Projeto sejam monitorados. Como em Scrum não há qualquer planejamento e análise dos dados que fazem parte do projeto, estes dados não podem ser monitorados
- **Diagnóstico:** Não satisfeita.

### SP 1.5 - Monitorar o Envolvimento dos Stakeholders

- Análise: Em Scrum o envolvimento dos *stakeholders* é garantido pelo *Product Owner* (que os representa na definição do escopo e das prioridades) e pelo *Scrum Master* (que deve garantir o cumprimento das regras do Scrum). Evidências indiretas desta participação, como atualizações no *Product Backlog*, mostram que os *stakeholders* estão envolvidos no projeto.
- Diagnóstico: Satisfeita.

### SP 1.6 - Conduzir Revisões de Progresso

- Análise: Em Scrum revisões freqüentes são realizadas a cada iteração (*Sprint Review*) e diariamente (*Daily Meeting*), inclusive com ajuda de gráficos (*Product Burndown* e *Sprint Burndown*).
- Diagnóstico: Satisfeita.

### SP 1.7 - Conduzir Revisões em Marcos

- Análise: Ao final de cada *Sprint*, o progresso do projeto e o cumprimento dos compromissos é revisado na reunião de *Sprint Review*.
- Diagnóstico: Satisfeita.

### SP 2.1 - Analisar Problemas

- Análise: Durante as reuniões diárias, o time reporta e analisa os problemas encontrados, registrando-os em um quadro branco, cartões de papel ou na *Impediments Backlog*. Além disso, é responsabilidade do *Scrum Master* identificar e resolver impedimentos o mais breve possível.
- Diagnóstico: Satisfeita.

### SP 2.2 - Tomar ações corretivas

- Análise: O *Scrum Master* é responsável por resolver os impedimentos assim que eles são identificados. No entanto, não há registro ou evidências do planejamento e monitoramento dos impedimentos, impedindo um acompanhamento dessas ações de correção.
- Diagnóstico: Parcialmente satisfeita.

### SP 2.3 - Gerenciar Ações Corretivas

- **Análise:** Quando os impedimentos encontrados são resolvidos, eles são excluídos da lista de impedimentos. No entanto, não há registro de quais ações foram tomadas para resolvê-los, impedindo uma análise da eficácia com que foram resolvidos.
- **Diagnóstico:** Parcialmente satisfeita.

Assim, Scrum atende parcialmente às práticas e metas da área de processo de Monitoramento e Controle de Projeto, como sumariza a tabela a seguir:

**Tabela 4.3 - Satisfação das práticas de PMC**

| Meta                                                  | Prática                                                   | Avaliação               |
|-------------------------------------------------------|-----------------------------------------------------------|-------------------------|
| SG 1<br>Monitorar o Projeto em relação ao Plano       | SP 1.1 Monitorar os Parâmetros de Planejamento do Projeto | Satisfeita              |
|                                                       | SP 1.2 Monitorar os Compromissos                          | Satisfeita              |
|                                                       | SP 1.3 Monitorar os Riscos do Projeto                     | Parcialmente satisfeita |
|                                                       | SP 1.4 Monitorar o Gerenciamento de Dados                 | Não satisfeita          |
|                                                       | SP 1.5 Monitorar o Envolvimento dos Stakeholders          | Satisfeita              |
|                                                       | SP 1.6 Conduzir Revisões de Progresso                     | Satisfeita              |
|                                                       | SP 1.7 Conduzir Revisões em Marcos                        | Satisfeita              |
| SG 2<br>Gerenciar Ações Corretivas até o Encerramento | SP 2.1 Analisar Problemas                                 | Satisfeita              |
|                                                       | SP 2.2 Tomar Ações Corretivas                             | Parcialmente satisfeita |
|                                                       | SP 2.3 Gerenciar Ações Corretivas                         | Parcialmente satisfeita |

### 4.3 Gerenciamento Integrado de Projetos (IPM) + IPPD

De acordo com o SEI, o propósito da área de processo de Gerenciamento Integrado de Projetos (IPM) + IPPD (Desenvolvimento Integrado de Produtos e Processos) é estabelecer e gerenciar o projeto e o envolvimento dos *stakeholders* basados em um processo integrado e definido a partir do conjunto de processos padrões da organização. Além disso, com a adição do IPPD, também cobre o estabelecimento de times integrados e uma visão comum para o projeto. O mapeamento destas práticas e meta é apresentado a seguir e os resultados são sumarizados na tabela 4.4.

## SG 1 - Utilizar o Processo Definido do Projeto

- **Análise:** Esta meta tem como objetivo garantir que o projeto é conduzido usando o processo definido que é adaptado do conjunto de processos padrões da organização. No entanto, Scrum não define, e nem mesmo comenta, que deva existir um processo padrão organizacional e que o processo do projeto deva ser derivado dele. Não contém práticas ou recomendações para criar derivados do processo organizacional e nem como os processos de cada um dos projetos possam contribuir com o processo organizacional.
- **Diagnóstico:** Todas as práticas desta meta são consideradas não satisfeitas:
  - o SP 1.1 - Estabelecer o Processo Definido do Projeto: Não satisfeita;
  - o SP 1.2 - Utilizar os Ativos de Processos Organizacionais para o Planejamento das Atividades do Projeto: Não satisfeita;
  - o SP 1.3 - Estabelecer o Ambiente de Trabalho do Projeto: Não satisfeita
  - o SP 1.4 - Integrar os Planos: Não satisfeita;
  - o SP 1.5 - Gerenciar o Projeto Utilizando os Planos Integrados: Não satisfeita; e
  - o SP 1.6 - Contribuir para os Ativos de Processos Organizacionais: Não satisfeita.

- **Observações:**

Em seu livro, Schwaber [SCHWABER 2004] afirma que em 2003, com a criação do Scrum Methodology, Certified Scrum Program (programa de certificação em Scrum) e do Projeto Quickstart, Scrum passou a ser compatível com o CMM (predecessor do CMMI) nível 3, incluindo as áreas de processo de institucionalização. Estes programas têm como objetivo auxiliar organizações a treinarem e implantarem Scrum de uma forma padrão, seguindo as recomendações de uma organização criada para disseminar a metodologia (Scrum Alliance). Dessa forma, Schwaber parece sugerir que o Scrum seja considerado o “processo organizacional” para estas instituições, já que inclui programas de treinamento, uma definição dos processos e ambientes de trabalho padrões, etc.

Este trabalho discorda desta abordagem onde Scrum é considerado o “processo organizacional” e onde cada projeto tem seu processo definido a partir do Scrum “padrão”. Considera-se aqui que as organizações possuem foco de negócio, valores e conhecimentos muito distintos para todas partirem de um processo comum e genérico como o Scrum ensinado por Schwaber. Acredita-se que o Scrum “padrão” deva servir como um ponto de partida para a definição do processo organizacional, mas que adaptações podem ser necessárias para adequá-lo a realidade de cada empresa.

### SP 2.1 - Gerenciar o Envolvimento dos Stakeholders

- **Análise:** As práticas e atividades de Scrum descrevem os papéis e responsabilidades de cada *stakeholder* e garante a participação dos envolvidos.

Além disso, é papel do *Scrum Master* monitorar e garantir o envolvimento dos *stakeholders*.

- Diagnóstico: Satisfeita.

### SP 2.2 - Gerenciar Dependências

- Análise: Dependências críticas podem ser monitoradas junto com os impedimentos nas reuniões diárias. É responsabilidade do *Scrum Master* tratar e corrigir essas dependências e impedimentos. No entanto, não há registro dos planos ou das ações que devem ser tomadas para resolver as dependências, nem são acordados prazos e responsabilidades. Além disso, não há estratégias de verificação ou de monitoramento das dependências e impedimentos.
- Diagnóstico: Parcialmente Satisfeita.

### SP 2.3 - Resolver Questões de Coordenação

- Análise: Durante as reuniões diárias, impedimentos, inclusive devido a dependências, são levantados pelo time e adicionados na *Impediments Backlog*, que serve como forma de comunicação com os *stakeholders* relevantes. É responsabilidade do *Scrum Master* corrigir estes impedimentos, escalando quando necessário, e monitorá-los até que eles estejam resolvidos, excluindo-os da lista de impedimentos quando terminados.
- Diagnóstico: Satisfeita.

### SP 3.1 - Estabelecer a Visão Compartilhada do Projeto

- Análise: Durante a fase de Planejamento, antes do primeiro *Product Backlog* ser criado, é feito um documento de visão, que vai servir como base da idéia do projeto. O documento de visão deve ser breve e acessível, comunicando a essência e o caráter do empreendimento [COCHANGO 2006]. Este documento de visão serve como essência para todos os times que estão realizando o projeto.
- Diagnóstico: Satisfeita.

### SP 3.2 - Estabelecer uma Estrutura Integrada para o Time

- Análise: Em Scrum, ao executar um *Scaled Project* (projeto escalonado) são utilizados mecanismos de escalonamento que garantem uma estrutura integrada para o time, conforme descrito na seção 3.6.
- Diagnóstico: Satisfeita.

### SP 3.3 - Alocar Requisitos para Times Integrados

- **Análise:** Ao executar um *Scaled Project* algumas práticas garantem que requisitos de infra-estrutura e definição da arquitetura sejam alocados para o time integrado, antes mesmo da criação e divisão dos times. Além disso, a composição dos times individuais deve conter pessoas que os representam no time integrado.
- **Diagnóstico:** Satisfeita.

### SP 3.4 - Estabelecer Times Integrados

- **Análise:** Os mecanismos de escalonamento garantem o estabelecimento de times integrados.
- **Diagnóstico:** Satisfeita.

### SP 3.5 - Garantir Colaboração entre as Interfaces dos Times

- **Análise:** A presença de um representante dos times individuais no time integrado garante a colaboração entre as interfaces dos times.
- **Diagnóstico:** Satisfeita.

Assim, Scrum atende parcialmente às práticas e metas da área de processo de Gerenciamento Integrado de Projetos + IPPD, como sumariza a tabela a seguir:

**Tabela 4.4 - Satisfação das práticas de IPM + IPPD**

| Meta                                                                          | Prática                                                                                              | Satisfação              |
|-------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|-------------------------|
| SG 1 Utilizar o Processo Definido do Projeto                                  | SP 1.1 Estabelecer o Processo Definido do Projeto                                                    | Não satisfeita          |
|                                                                               | SP 1.2 Utilizar os Ativos de Processos Organizacionais para o Planejamento das Atividades do Projeto | Não satisfeita          |
|                                                                               | SP 1.3 Estabelecer o Ambiente de Trabalho do Projeto                                                 | Não satisfeita          |
|                                                                               | SP 1.4 Integrar os Planos                                                                            | Não satisfeita          |
|                                                                               | SP 1.5 Gerenciar o Projeto Utilizando os Planos Integrados                                           | Não satisfeita          |
|                                                                               | SP 1.6 Contribuir para os Ativos de Processos Organizacionais                                        | Não satisfeita          |
| SG 2 Coordenar e Colaborar com os Stakeholders Relevantes                     | SP 2.1 Gerenciar o Envolvimento dos Stakeholders                                                     | Satisfeita              |
|                                                                               | SP 2.2 Gerenciar Dependências                                                                        | Parcialmente satisfeita |
|                                                                               | SP 2.3 Resolver Questões de Coordenação                                                              | Satisfeita              |
| SG 3 Aplicar Princípios do Desenvolvimento Integrado do Produto e do Processo | SP 3.1 Estabelecer a Visão Compartilhada do Projeto                                                  | Satisfeita              |
|                                                                               | SP 3.2 Estabelecer uma Estrutura Integrada para o Time                                               | Satisfeita              |
|                                                                               | SP 3.3 Alocar Requisitos para Times Integrados                                                       | Satisfeita              |
|                                                                               | SP 3.4 Estabelecer Times Integrados                                                                  | Satisfeita              |
|                                                                               | SP 3.5 Garantir colaboração entre as interfaces dos times                                            | Satisfeita              |

#### 4.4 Gerenciamento de Riscos (RSKM)

De acordo com o SEI, o propósito da área de processo de Gerenciamento de Riscos (RSKM) é identificar problemas em potencial antes que eles ocorram, de forma a permitir o planejamento e a execução de atividades que possam mitigar ou controlar o risco durante o ciclo de vida do projeto [SEI 2006]. Ainda segundo o SEI, o gerenciamento de riscos deve ser dividido em três partes: definir uma estratégia de gerenciamento de riscos, identificar e analisar os riscos e tratar os riscos identificados, incluindo a execução de planos de mitigação quando necessários.

Em Scrum os riscos podem ser identificados durante as reuniões diárias e, nesse caso, fariam parte da lista de impedimentos. No entanto, para serem considerados identificados segundo o CMMI, deveria haver uma avaliação, priorização e categorização dos mesmos. Em Scrum, este processo de determinação e documentação dos parâmetros do risco não ocorre.

Além disso, planos de mitigação e contingência não são definidos e não há usos de bases históricas para ajudar o processo de identificação, comprometendo seriamente a estratégia de gerenciamento dos riscos.

Assim, Scrum só atende parcialmente a uma das práticas da área de processo de Gerenciamento de Riscos (RSKM) relacionada a identificação dos riscos e não satisfaz nenhuma das outras práticas, como sumariza a tabela a seguir:

**Tabela 4.5 - Satisfação das práticas de RSKM**

| Meta                                     | Prática                                                     | Satisfação              |
|------------------------------------------|-------------------------------------------------------------|-------------------------|
| SG 1 Preparar o Gerenciamento dos Riscos | SP 1.1 Determinar as fontes e categorias do risco           | Não satisfeita          |
|                                          | SP 1.2 Determinar os parâmetros do risco                    | Não satisfeita          |
|                                          | SP 1.3 Estabelecer a estratégia de gerenciamento dos riscos | Não satisfeita          |
| SG 2 Identificar e Analisar Riscos       | SP 2.1 Identificar Riscos                                   | Parcialmente satisfeita |
|                                          | SP 2.2 Avaliar, categorizar e priorizar riscos              | Não satisfeita          |
| SG 3 Mitigar Riscos                      | SP 3.1 Desenvolver planos de mitigação de riscos            | Não satisfeita          |
|                                          | SP 3.2 Implementar planos de mitigação de riscos            | Não satisfeita          |

## 4.5 Gerenciamento de Acordos com Fornecedores (SAM)

O propósito da área de processo de Gerenciamento de Acordos com Fornecedores (SAM) é gerenciar a aquisição de produtos de terceiros. Dentro desta área de processo estão inclusas a identificação das necessidades de aquisição, a escolha e o estabelecimento de acordos com fornecedores, o monitoramento dos processos do fornecedor e a aceitação do produto adquirido.

Em Scrum não há qualquer prática ou recomendação para tratar com fornecedores e módulos adquiridos. Assim, a metodologia não atende a nenhuma das práticas da área de processo de Gerenciamento de Acordos com Fornecedores (SAM), como sumariza a tabela a seguir:

**Tabela 4.6 - Satisfação das práticas de SAM**

| Meta                                         | Prática                                                           | Satisfação     |
|----------------------------------------------|-------------------------------------------------------------------|----------------|
| SG 1<br>Estabelecer Acordos com Fornecedores | SP 1.1 Determinar tipo de aquisição                               | Não satisfeita |
|                                              | SP 1.2 Escolher fornecedores                                      | Não satisfeita |
|                                              | SP 1.3 Estabelecer acordos com fornecedores                       | Não satisfeita |
|                                              | SP 2.1 Executar o acordo com o fornecedor                         | Não satisfeita |
| SG 2 Satisfazer Acordos com Fornecedores     | SP 2.2 Monitorar os processos selecionados do fornecedor          | Não satisfeita |
|                                              | SP 2.3 Avaliar os produtos de trabalho selecionados do fornecedor | Não satisfeita |
|                                              | SP 2.4 Aceitar o produto adquirido                                | Não satisfeita |
|                                              | SP 2.5 Produtos de transição                                      | Não satisfeita |

## 4.6 Gerenciamento Quantitativo do Projeto (QPM)

O propósito da área de Gerenciamento Quantitativo do Projeto (QPM) é gerenciar quantitativamente o processo definido para o projeto de forma a atingir os objetivos de qualidade e desempenho definidos [SEI 2006]. Para atingir este objetivo, sub-processos do projeto são identificados e medidos para monitorar sua conformidade com os objetivos de qualidade e desempenho definidos. Além disso, os dados de tais medições são gravados e farão parte do acervo organizacional com o objetivo de melhorar os processos padrões da empresa.

Em Scrum, não há nenhuma prática que esteja aderente ou sequer inclinada para satisfazer esta prática. Processos organizacionais não são alvos da metodologia e apenas as métricas necessárias para monitorar o cumprimento dos compromissos do projeto são definidas. Assim, Scrum não atende a nenhuma das práticas da área de processo de Gerenciamento Quantitativo do Projeto (QPM), como sumariza a tabela a seguir:

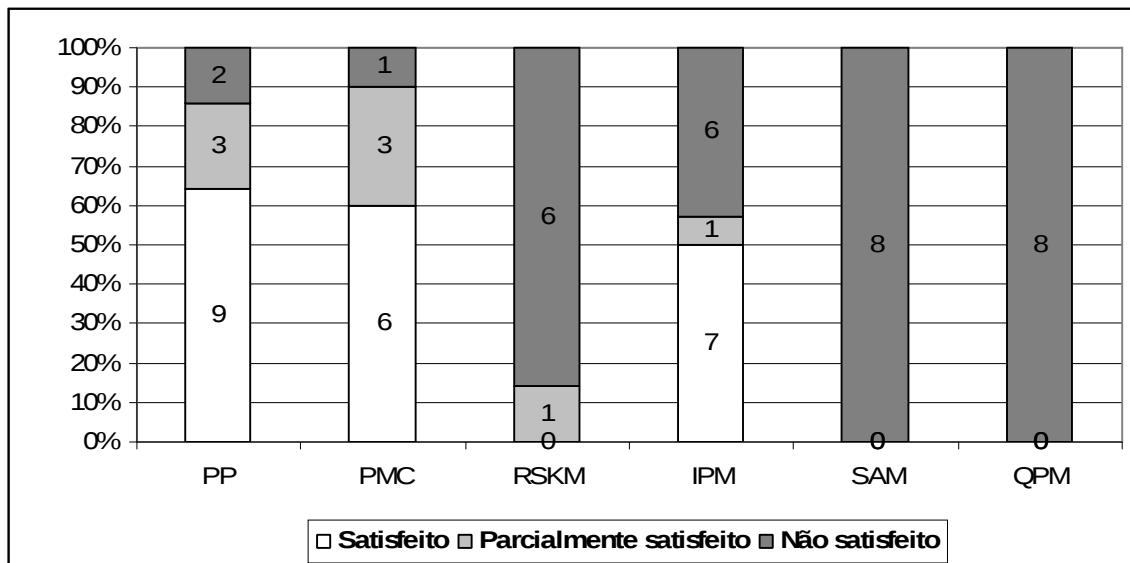
**Tabela 4.7 - Satisfação das práticas de QPM**

| Objetivo                                                       | Prática                                                             | Satisfação     |
|----------------------------------------------------------------|---------------------------------------------------------------------|----------------|
| SG 1 Gerenciar quantitativamente o projeto                     | SP 1.1 Estabelecer Objetivos do Projeto                             | Não satisfeita |
|                                                                | SP 1.2 Compor o Processo Definido                                   | Não satisfeita |
|                                                                | SP 1.3 Escolher subprocessos que serão gerenciados estatisticamente | Não satisfeita |
|                                                                | SP 1.4 Gerenciar a performance do projeto                           | Não satisfeita |
| SG 2 Gerenciar estatisticamente a performance dos subprocessos | SP 2.1 Selecionar as medidas e técnicas analíticas                  | Não satisfeita |
|                                                                | SP 2.2 Aplicar métodos estatísticos para entender variação          | Não satisfeita |
|                                                                | SP 2.3 Monitorar performance dos subprocessos selecionados          | Não satisfeita |
|                                                                | SP 2.4 Gravar gerenciamento estatístico dos dados                   | Não satisfeita |

## 4.7 Análise dos resultados

De acordo com o diagnóstico e as justificativas apresentadas percebe-se que Scrum satisfaz completamente 36,07% das práticas do CMMI na categoria de Gerenciamento de Projetos, satisfaz parcialmente a 13,11% das práticas e não satisfaz a 50,82%. A figura a seguir ilustra a satisfação das práticas por áreas de processo.

**Figura 4.1 - Satisfação das práticas de Gerenciamento de Projetos**



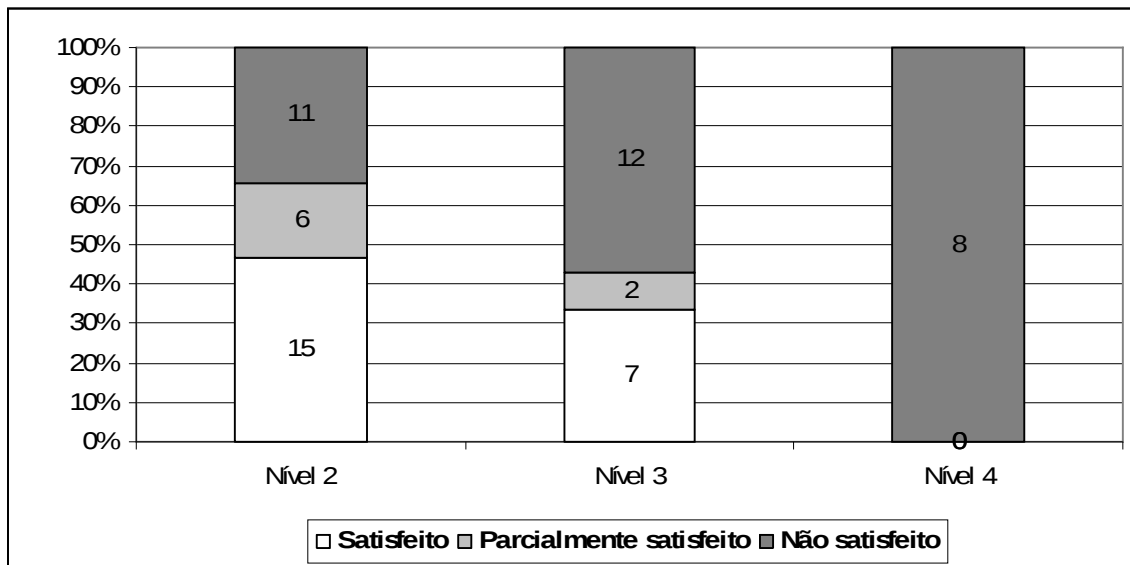
A tabela a seguir demonstra os percentuais de satisfação das práticas por áreas de processo.

**Tabela 4.8 – Percentuais de satisfação por áreas de processo**

| Diagnóstico             | PP     | PMC    | RSKM   | IPM+IPPD | SAM  | QPM  | Total  |
|-------------------------|--------|--------|--------|----------|------|------|--------|
| Satisfeito              | 64,29% | 60,00% | 0,00%  | 50,00%   | 0%   | 0%   | 36,07% |
| Parcialmente satisfeito | 21,43% | 30,00% | 14,29% | 7,14%    | 0%   | 0%   | 13,11% |
| Não satisfeito          | 14,29% | 10,00% | 85,71% | 42,86%   | 100% | 100% | 50,82% |

Além disso, mesmo considerando apenas as práticas dos níveis de maturidade mais baixos, Scrum não satisfaz completamente a todas as áreas de processo. De fato, somente 46,88% das práticas de nível de maturidade 2 são completamente satisfeitas e 33,33% das práticas de nível 3. As práticas não satisfeitas de nível 2 e 3 somam 57,14%. A figura a seguir sumariza a satisfação das práticas de Gerenciamento de Projetos do CMMI agrupadas por nível de maturidade.

**Figura 4.2 - Satisfação das práticas por nível de maturidade**



A tabela a seguir demonstra os percentuais de satisfação das práticas de Gerenciamento de Projetos por níveis de maturidade.

**Tabela 4.9 – Percentuais de satisfação por níveis de maturidade**

| Diagnóstico             | 2      | 3      | 4    | Total  |
|-------------------------|--------|--------|------|--------|
| Satisfeito              | 46,88% | 33,33% | 0%   | 36,07% |
| Parcialmente satisfeito | 18,75% | 9,52%  | 0%   | 13,11% |
| Não satisfeito          | 34,38% | 57,14% | 100% | 50,82% |

Assim, demonstra-se que Scrum é parcialmente aderente ao CMMI mesmo considerando apenas as áreas de processo de Gerenciamento de Projetos. As extensões a serem feitas para tornar Scrum aderente nesta categoria são:

- Melhorar o processo de estimativas com o uso de técnicas ou práticas alternativas para adequar Scrum às práticas de PP (1.2 e 1.4);

- Melhorar o gerenciamento de riscos, adequando Scrum às práticas de RSKM (todas), PP (2.2) e PMC (1.3);
- Melhorar a definição e o gerenciamento do orçamento do projeto, adequando Scrum às práticas de PP (1.4 e 2.1);
- Melhorar a definição e a documentação das ações corretivas tomadas pelo *Scrum Master*, adequando a metodologia às práticas de PMC (2.2 e 2.3) e IPM (2.2);
- Utilizar padrões de dados e processos padrões organizacionais, modificando-os para cada projeto, adequando Scrum às práticas de PP (2.3), PMC (1.4) e IPM (1.1 até 1.6);
- Definir um processo para o gerenciamento de fornecedores, adequando Scrum às práticas de SAM (todas); e
- Definir um processo para o planejamento de objetivos de qualidade e desempenho, além de um processo para a medição e coleta dos dados que vão indicar se estes objetivos estão sendo cumpridos. Com isso, espera-se gerenciar quantitativamente o processo, adequando Scrum às práticas de QPM (todas).

## 5. Tornando Scrum mais aderente ao CMMI

---

Neste estudo foram identificadas as áreas de processo e práticas do CMMI que não são cobertas por Scrum. Estas áreas de processo não aderentes tornam o Scrum “padrão” uma opção inválida para empresas que implantaram ou estão implantando o CMMI e, portanto, desejam ser aderentes a esse modelo de qualidade. No entanto, durante a revisão bibliográfica para compor este trabalho, foram encontradas diversas práticas de outras metodologias e processos ágeis que poderiam tornar Scrum mais aderente ao CMMI.

A seguir, na seção 5.1, serão revistos os resultados do mapeamento apresentado, priorizando-se as deficiências de Scrum com relação ao CMMI. Das seções 5.2 a 5.5 serão feitas propostas para tornar algumas áreas mais aderentes, através da incorporação de algumas práticas de outras metodologias e processos. Por fim, na seção 5.6 o mapeamento e as percentagens de aderências serão revisados, considerando as extensões acrescidas.

### 5.1 Priorização das áreas de processo não-satisfeitas

---

Como vimos no capítulo anterior, Scrum não é aderente a todas as metas e práticas do CMMI. Dentro das áreas não aderentes da categoria de gerenciamento de projetos identificadas, algumas áreas de processo, metas e práticas foram consideradas menos relevantes para as atividades de tornar Scrum aderente ao CMMI:

- Definição dos padrões de dados e processos organizacionais: Entende-se que a definição dos padrões e processos organizacionais deva ser executada baseada na realidade da empresa, do estado e maturidade de seus processos e das pessoas e capacidades da organização. Assim, práticas e recomendações aplicáveis a uma empresa podem não servir para outra, diminuindo a contribuição de um trabalho que tente definir práticas nesta linha;
- Gerenciamento de Acordo com Fornecedores (SAM): Esta área de processo só é aplicável a organizações que tercerizam parte do seu desenvolvimento de software ou adquirem no mercado componentes e bibliotecas de terceiros. Entende-se que esta área de processo não é relevante para muitas organizações e que o gerenciamento de fornecedores não deveria fazer parte do escopo de Scrum, devendo ser tratado separadamente, se aplicável, no processo organizacional; e
- Gerenciamento Quantitativo do Projeto (QPM): Esta área de processo do CMMI nível 4 requer o uso muitas métricas, coleta de dados e análise de sub-processos com o objetivo de prover dados para melhorar o processo organizacional. Se realizado de forma errônea, adaptações para adequar uma metodologia ágil ao QPM podem engessá-la e diminuir a eficiência do processo ágil. O processo ágil que mais se aproxima de cumprir as práticas do QPM é o MSF for CMMI

[ANDERSON 2005] que segue os princípios de Edwards Deming [DEMING 1982] para criar uma metodologia que prega o fluxo contínuo de valor para o cliente através de ciclos curtos e muitas iterações, mantendo a redução e o controle de variações através de práticas ágeis e do uso extensivo de ferramentas de apoio como o Visual Studio Team System [MICROSOFT 2006].

## 5.2 Estendendo o processo de estimativa em Scrum

---

Em seu livro sobre planejamento ágil, Cohn [COHN 2006] sugere o uso de Story Points e do Planning Poker como um processo de estimativa. A idéia por trás dos Story Points é prover aos *stakeholders* envolvidos uma idéia da complexidade das atividades através do uso de estimativas relativas. Com o uso de Story Points diz-se que uma atividade custa 2 pontos, outra 3 e outra 7. Dessa maneira, o cliente tem idéia de que a primeira atividade é a mais fácil de ser realizada, a segunda vai levar aproximadamente 1,5 vezes o tempo da primeira e a terceira mais de duas vezes. Com o uso de bases históricas para cada time é possível realizar uma estimativa de quantos Story Points poderão ser desenvolvidos em uma iteração. Nesta seção, adaptaremos o uso deste método para os artefatos e atividades de Scrum.

Para utilizar o processo de estimativa por Story Points em Scrum, inicialmente o Product Owner deve atribuir valores de negócio para cada um dos itens do Product Backlog, priorizando quantitativamente todos os itens. Estes valores podem ser números arbitrários, de 1 a 1000, por exemplo, ou o retorno esperado, em reais, de cada item.

Em seguida, o time deve identificar o item do Product Backlog mais simples de implementar e atribuir para este item um número de Story Points de baixo de custo, normalmente 1 ou 2. A seguir, o time deve estimar todos os itens do Product Backlog tomando este item como referencial e responder perguntas como “Se X custa 2 para implementar, quanto custa Y?”. Para os custos do Product Backlog em Story Points recomenda-se usar os números de Fibonnaci (1,2,3,5,8,13,21,34) Assim, o time converge mais rápido para um resultado e não fica discutindo sobre valores altos e muito próximos (“Este item custa 23 ou 25 pontos?”). O Product Owner deve estar presente nesta seção de estimativa para tirar dúvidas e esclarecer melhor as funcionalidades que vão ser implementadas.

Com o Product Backlog estimado, no primeiro Sprint utiliza-se de bases históricas (ou da experiência do time, quando dados não estão disponíveis) para definir quantos Story Points o time desenvolverá na Sprint. A partir da segunda Sprint, usa-se o mesmo número que foi efetivamente desenvolvido na Sprint anterior. Não se dá “crédito parcial” a itens do Product Backlog. Ou eles foram completamente desenvolvidos, testados e aceitos pelo cliente ou eles não são considerados para a velocidade de desenvolvimento do time.

### 5.3 Estendendo o processo de definição do orçamento em Scrum

---

No MSF for CMMI [MICROSOFT 2006], no OpenUP [OPENUP 2007] e em metodologias tradicionais como o RUP [JACOBSON 1999], o processo da definição dos custos é feito paralelamente às estimativas do tempo de desenvolvimento. Com base nestas outras metodologias, uma prática alternativa para o processo de custos que se adapta as atividades do Scrum é descrita a seguir.

Juntamente com o primeiro *Product Backlog* o *Product Owner*, ou o *Scrum Master* nos casos em que o *Product Owner* não faz parte da organização em que o software está sendo desenvolvido, deve criar o *Cost Backlog*, uma planilha com os custos estimados de cada recurso alocado para o desenvolvimento. Nesta planilha devem estar, por exemplo, o custo mensal dos desenvolvedores, das máquinas necessárias e até os percentuais das comissões de venda e do rateio dos custos da alta gerência.

Esta planilha deve estar ligada à planilha do *Product Backlog*, tendo como entrada as estimativas de duração do projeto em *Sprints*. Dessa forma, à medida que o *Product Backlog* vai sendo atualizado, modificando a duração do projeto, o *Cost Backlog* vai se ajustando. Além disso, no início de cada Sprint, a pessoa responsável pelo *Cost Backlog* deve atualizá-lo, adicionando, se necessário, novos custos ou removendo custos que não são mais adequados.

### 5.4 Estendendo o processo de gerenciamento de riscos em Scrum

---

Um risco é certo evento ou condição que, se ocorrer, terá um efeito negativo ou positivo em um ou mais objetivos do projeto [PMI 2004]. Riscos do projeto podem ser vistos como ameaças ou oportunidades. Oportunidade significa que tomar um risco calculado pode, por exemplo, trazer vantagens competitivas para um produto ou organização. Se há benefícios associados a uma oportunidade, então é possível aceitar certos graus de risco para um projeto ser bem sucedido. [SEI 1999].

No OpenUP [OPENUP 2007], um framework para processos ágeis da equipe de processos do Eclipse, a área de Gerência de Riscos tem como propósito aumentar a probabilidade e o impacto de eventos positivos (oportunidades) e minimizar o impacto de eventos negativos ao projeto (ameaças). Com base neste framework, uma prática alternativa para o processo de gerenciamento de riscos que se adapta as atividades do Scrum é descrita a seguir.

Durante a primeira reunião de planejamento da iteração (*Sprint Planning 1*) os riscos para as atividades a serem desenvolvidas devem ser identificados, com foco nos itens de maior valor. Além disso, eles devem ser analisados quanto a sua probabilidade (alta, média ou baixa), impacto (severo, médio ou leve), categoria (técnico, gerenciamento, organizacional ou externo), e a magnitude do risco, sendo a ultima uma combinação da

probabilidade com o impacto. Estes riscos farão parte do *Risk Backlog* um repositório de riscos que pode estar guardado numa planilha, quadro branco ou cartões de papel. Na próxima iteração, estes riscos devem ser reavaliados e, se necessário, atualizados.

Durante a segunda reunião de planejamento da iteração (*Sprint Planning 2*) a estratégia de resposta aos riscos mais prioritários (os 5 ou 10 primeiros) deve ser planejada, seguindo uma das seguintes opções:

- Explorar/Evitar: Colocar ações no *Sprint Backlog* para fazer com que o projeto não seja afetado, caso o risco seja negativo, ou para garantir que o risco ocorra, caso ele seja positivo.
- Melhorar/Mitigar: Colocar ações no *Sprint Backlog* para diminuir a probabilidade ou impacto do risco, caso ele seja negativo, ou para aumentar a probabilidade ou impacto, caso seja positivo.
- Transferir/Compartilhar: Reorganizar o projeto de forma a transferir a responsabilidade do risco para outra organização ou indivíduo. Isto ocorre porque, às vezes, terceiros podem mitigar (caso seja negativo) ou explorar (caso seja positivo) melhor o risco.
- Aceitar: Aceitar um risco é “planejar não planejar”, ou seja, decidir seguir com o projeto e não executar nenhuma ação para modificar a probabilidade ou impacto do risco. No caso de ameaças a estratégia pode incluir um plano para remediá-la, ou seja, diminuir o efeito do dano depois que o risco ocorre.

As ações criadas pela estratégia de gerenciamento dos riscos farão parte do *Sprint Backlog* juntamente com as outras atividades da iteração. Durante a *Daily Meeting*, os riscos podem ser monitorados e, caso sejam concretizados, farão parte do *Impediments Backlog*, e serão tratados pelo *Scrum Master*.

## 5.5 Estendendo o processo de resolução de impedimentos em Scrum

---

Em Scrum, o *Impediments Backlog* normalmente contém apenas uma breve descrição do impedimento a ser resolvido pelo *Scrum Master*. Infelizmente, essa forma não provê evidências de quais ações corretivas serão tomadas, impedindo o seu monitoramento.

Para melhorar o controle e monitoramento das ações corretivas, sugere-se que o *Impediments Backlog* contenha, além da descrição do impedimento, a sua probabilidade (alta, média ou baixa), impacto (severo, médio ou leve), categoria (técnico, gerenciamento, organizacional ou externo), as ações a serem tomadas pelo *Scrum Master* e uma estimativa de sua complexidade através, por exemplo, de uma data alvo para eliminação do impedimento. Além disso, os impedimentos resolvidos não devem ser

excluídos, sendo apenas marcados como resolvidos, deixando evidências dos problemas encontrados e das ações tomadas.

## 5.6 Revisão do mapeamento considerando as extensões propostas

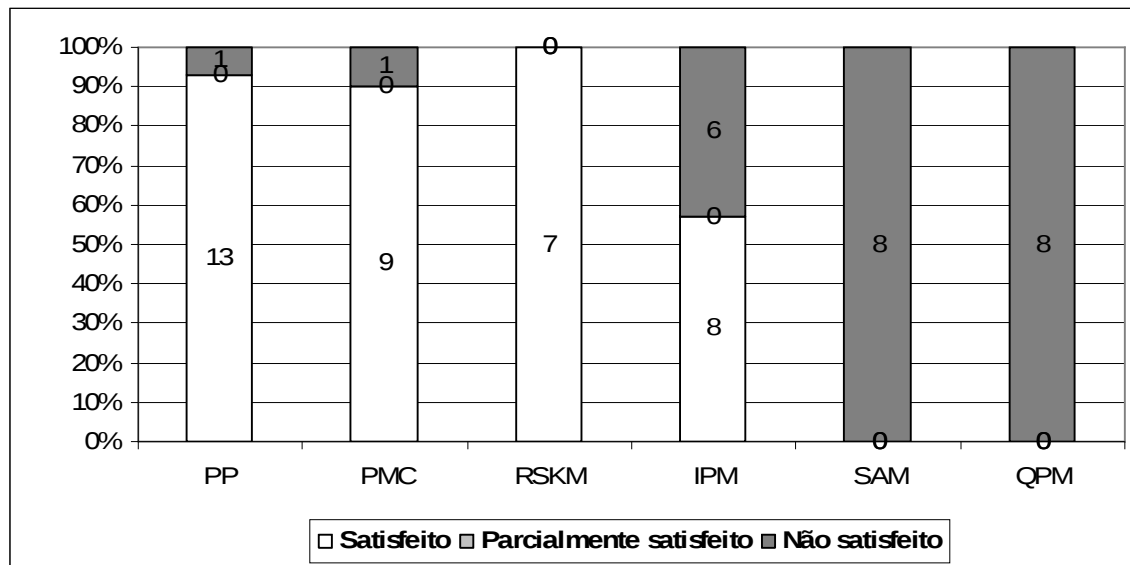
Considerando as melhorias necessárias para tornar Scrum aderente observadas no final do capítulo 4 (Mapeando Scrum nas áreas de processo de gerenciamento do CMMI) e desconsiderando (conforme explicado na seção 5.1) as áreas de processo SAM, QPM e as práticas relacionadas aos padrões e processos organizacionais, com o uso das extensões propostas Scrum torna-se 100% aderente às práticas da categoria de Gerenciamento de Projetos do CMMI. A tabela a seguir ilustra o diagnóstico das práticas consideradas relevantes para este trabalho antes e depois das extensões propostas:

**Tabela 5.1 - Satisfação das práticas após extensões**

| Área | Prática                                                                       | Sem extensões           | Com extensões |
|------|-------------------------------------------------------------------------------|-------------------------|---------------|
| RSKM | SP 1.1 Determinar as fontes e categorias do risco                             | Não satisfeita          | Satisfeita    |
| RSKM | SP 1.2 Determinar os parâmetros do risco                                      | Não satisfeita          | Satisfeita    |
| RSKM | SP 1.3 Estabelecer a estratégia de gerenciamento dos riscos                   | Não satisfeita          | Satisfeita    |
| RSKM | SP 2.1 Identificar Riscos                                                     | Parcialmente satisfeita | Satisfeita    |
| RSKM | SP 2.2 Avaliar, categorizar e priorizar riscos                                | Não satisfeita          | Satisfeita    |
| RSKM | SP 3.1 Desenvolver planos de mitigação de riscos                              | Não satisfeita          | Satisfeita    |
| RSKM | SP 3.2 Implementar planos de mitigação de riscos                              | Não satisfeita          | Satisfeita    |
| IPM  | SP 2.2 Gerenciar Dependências                                                 | Parcialmente satisfeita | Satisfeita    |
| PMC  | SP 2.2 Tomar ações corretivas                                                 | Parcialmente satisfeita | Satisfeita    |
| PMC  | SP 2.3 Gerenciar ações corretivas                                             | Parcialmente satisfeita | Satisfeita    |
| PP   | SP 1.2 Estabelecer Estimativas de Atributos de Produtos de Trabalho e Tarefas | Não satisfeita          | Satisfeita    |
| PP   | SP 1.4 Determinar Estimativas de Esforço e Custo                              | Parcialmente satisfeita | Satisfeita    |
| PP   | SP 2.1 Estabelecer o Orçamento e o Cronograma                                 | Parcialmente satisfeita | Satisfeita    |
| PP   | SP 2.2 Identificar os Riscos do Projeto                                       | Parcialmente satisfeita | Satisfeita    |
| PP   | SP 2.3 Planejar o Gerenciamento de Dados                                      | Não satisfeita          | Satisfeita    |

A figura a seguir ilustra a satisfação das práticas por áreas de processo depois das extensões propostas:

**Figura 5.1 - Satisfação das práticas de Gerenciamento de Projetos após extensões**



Como já explicado, as únicas práticas de PP, PMC e IPM não satisfeitas estão relacionadas aos padrões e processos organizacionais. Além disso, SAM e QPM foram considerados de baixa relevância para este trabalho e fora do escopo de Scrum. A tabela a seguir ilustra os percentuais de satisfação das práticas, agrupadas por níveis de maturidade depois das extensões propostas:

**Tabela 5.2 – Satisfação por nível de maturidade após extensões**

| Diagnóstico             | 2      | 3      | 4    | Total  |
|-------------------------|--------|--------|------|--------|
| Satisfeito              | 68,75% | 71,43% | 0%   | 60,66% |
| Parcialmente satisfeito | 0,00%  | 0,00%  | 0%   | 0,00%  |
| Não satisfeito          | 31,25% | 28,57% | 100% | 39,34% |

## 6. Conclusões

---

Este trabalho foi motivado especialmente pela busca de respostas em relação aos mundos das metodologias ágeis e dos modelos de qualidade: se eles poderiam conviver num mesmo ambiente e, em especial, se seria possível obter benefícios destes dois mundos.

Primeiramente, um estudo detalhado foi feito de Scrum, uma das mais difundidas metodologias ágeis, do CMMI, um dos principais modelos de qualidade do mundo.

Neste trabalho reiteramos que modelos de qualidade e metodologias ágeis não são incompatíveis, analisando a aderência de Scrum às áreas de processo da categoria de Gerenciamento do CMMI, através de um mapeamento. Demonstramos que Scrum é parcialmente aderente ao nível três de maturidade (Definido) do CMMI, satisfazendo diversas práticas e não-satisfazendo a outras. Por fim, após o diagnóstico, algumas extensões baseadas em outras metodologias e processos ágeis foram propostas para tornar Scrum completamente aderente ao CMMI.

As conclusões e os resultados obtidos são apresentados da seguinte forma:

- Seção 6.1 - Objetivos atingidos: relaciona os objetivos propostos e alcançados;
- Seção 6.2 - Trabalhos futuros: descreve estudos que podem ser realizados tendo como base este documento; e
- Seção 6.3 - Considerações finais: apresenta algumas considerações sobre os assuntos abordados

### 6.1 Objetivos atingidos

---

Como descrito na introdução, este trabalho tem como objetivo responder as seguintes perguntas:

1. Quão aderente é Scrum à área de gerencia de projetos do CMMI? Quais práticas e metas do CMMI são atendidas pelas atividades e práticas do Scrum e de outras metodologias ágeis? Quais não são?
2. Scrum e CMMI podem conviver numa mesma organização? Se sim, até que nível de maturidade? Se não, em quais áreas e por que não?
3. Scrum precisa ser adaptado para torná-lo compatível com a certificação CMMI na área de gerencia de projetos? Se sim, como adaptá-lo sem torná-lo um processo “pesado e engessado” e sem perder sua essência ágil? Existem outras metodologias e processos ágeis que resolvem as deficiências de Scrum?

Através das informações apresentadas neste trabalho, foram obtidas as seguintes respostas:

1. Scrum é parcialmente aderente à área de gerencia de projetos do CMMI. De fato, no diagnóstico realizado, 22 das práticas do CMMI foram consideradas completamente satisfeitas, enquanto que 39 práticas foram consideradas parcialmente satisfeitas ou não-satisfeitas.
2. Scrum e CMMI podem conviver numa mesma organização. Com as extensões propostas no Capítulo 5, através da incorporação de práticas de outras metodologias ágeis, Scrum torna-se aderente ao CMMI nível três de maturidade (Definido). Além dessas extensões, as organizações que terceirizam parte do seu desenvolvimento de software precisam complementar Scrum com práticas de Gerência de Fornecedores, para serem completamente aderentes.
3. Scrum precisou ser adaptado para torná-lo compatível com a certificação CMMI na área de gerencia de projetos. Neste trabalho, práticas de outras metodologias e processos ágeis foram incorporadas, adaptando Scrum sem torná-lo um processo “pesado e engessado” e sem perder sua essência ágil.

## 6.2 Trabalhos futuros

---

Este estudo pode ser estendido de várias formas, visando aumentar a contribuição para as áreas de engenharia e qualidade de software, além beneficiar empresas e organizações que desejam implantar paralelamente metodologias ágeis e modelos de qualidade.

São sugeridos os seguintes trabalhos futuros:

- Aplicar as extensões propostas a um projeto real de desenvolvimento e manutenção de software utilizando Scrum, com o objetivo de avaliar o impacto da solução. A partir desta aplicação pode-se obter um estudo de caso;
- Analisar a aderência de Scrum para outras categorias de processo do CMMI, como Engenharia e Suporte, com o objetivo de prover uma visão e avaliação mais abrangente;
- Propor extensões para as áreas de processo de Gerenciamento de Projetos consideradas menos relevantes por este trabalho, como Gerenciamento de Acordos com Fornecedores (SAM) e Gerenciamento Quantitativo do Projeto (QPM); e
- Identificar práticas alternativas (como aceitas pelo CMMI) para aumentar a satisfação de Scrum ao modelo de qualidade.

### 6.3 Considerações finais

---

Cada vez mais a comunidade de software vem aceitando, entendendo e aplicando metodologias ágeis. Elas se mostram uma alternativa viável especialmente para grupos e organizações pequenas, por serem muito fáceis de implantar e de ter seus conceitos disseminados por toda a equipe.

Apesar disto é notória a falta de tratamento de aspectos relevantes do desenvolvimento de software por grande parte destas metodologias. A combinação das metodologias ágeis com estes aspectos, seja com base em um modelo de qualidade como o CMMI, seja com base nos estudos e resultados obtidos pela engenharia de software, se mostra uma solução factível e viável para diversos ambientes de desenvolvimento, trazendo ganhos de produtividade e melhorando ainda mais os sistemas e produtos desenvolvidos.

## 7. Referências Bibliográficas

---

[AGILE ALLIANCE 2007] Agile Alliance. What Is Agile Software Development? Disponível em <<http://www.agilealliance.org/intro>>. Acesso em maio 2007.

[ANDERSON 2005] Anderson D. J., "Stretching Agile to fit CMMI Level 3 the story of creating MSF for CMMI Process Improvement at Microsoft Corporation", Agile2005Conference, Disponível em: [http://www.agilemanagement.net/Articles/Papers/Agile\\_2005\\_Paper\\_DJA\\_v1\\_5.pdf](http://www.agilemanagement.net/Articles/Papers/Agile_2005_Paper_DJA_v1_5.pdf). Acesso em maio de 2007.

[BECK 2000] Beck, K. Extreme Programming Explained: Embrace Change, Addison-Wesley, 1ª Edição, 2000.

[BEEDLE 2004] Beedle, M. Agile Software Development with Scrum. Prentice Hall. 1ª Edição, 2004.

[CAMPELO 2003] Campelo, Renata E. C. XP-CMM2: Um Guia para Utilização de Extreme Programming em um Ambiente Nível 2 do CMM, Dissertação de Mestrado, Centro de Informática, Universidade Federal de Pernambuco, 2003.

[CHRISSIS et al. 2003] Chrissis, M. B.; Konrad, M.; Shrun, Sandy. CMMI Guidelines for process integration and product improvement, Addison-Wesley, 2003.

[CLIPP 1990] F. Paul Clipp III, Total Quality Management, The National Management Association, 1990.

[COCHANGO 2006] Cochango, Scrum for team systems, <http://www.scrumforteamssystem.com>. Acesso em maio de 2007.

[COHN 2006] Cohn, M., Agile Estimating and Planning, Prentice Hall, 2006

[CROSBY 1979] Crosby, Philip B. Quality Is Free: The Art of Making Quality Certain. New York: McGraw-Hill, 1979.

[CRYSTAL 2007] Web site Crystal Methodologies, Disponível em: <<http://crystalmethodologies.org/>>. Acesso em junho de 2007.

[DEMARCO 1998] DeMarco, Tom; Lister, Timothy. Peopleware: productive projects and teams. Dorset House Publishing, 2ª Edição, 1998

[DEMING 1982] Deming, W. Edwards, Out of Crisis, The MIT Press, Cambridge, MA 2000

[FOWLER 2006] Fowler, Martin. The New Methodology. Disponível em: <<http://martinfowler.com/articles/newMethodology.html>>. Acesso em maio 2007

[HIGHSMITH 2002] Highsmith J., Agile Software Development Ecosystems, Addison-Wesley, 1ª Edição, 2002.

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

[JACOBSON 1999] Jacobson I.; et. al. The Unified Software Development Process. Addison-Wesley Press. 1ª Edição, 1999.

[KÄHKÖNEN et al. 2004] 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

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

[MARÇAL 2007] Marçal, A., Freitas, B., Soares, F., Belchior, A., Mapping CMMI Project Management Process Areas to SCRUM Practices, 31st Annual Software Engineering Workshop, Loyola College, Baltimore, USA, 2007.

[MELLO ET AL. 2002] C.H.P. Mello et al, ISO 9001:2000: Sistema de Gestão da Qualidade para Operações de Produção e Serviços, São Paulo, Atlas, 2002.

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

[MICROSOFT 2006] Microsoft Corporation, Microsoft Solutions Framework Homepage, Disponível em: <http://go.microsoft.com/fwlink/?linkid=40663> Acesso em agosto de 2006.

[NAUR et al 1969] Naur, Peter (Ed.); Randell, Brian (Ed.). Software Engineering: report on a conference... In: NATO Conference on Software Engineering, Oct. 1968, Garmish, Germany. NATO Science Committee Garmish, Germany, Jan. 1969.

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

[OGUNNAIKE et al. 1992] Ogunnaike, B.A. & Ray, W. H., Process Dynamics, Modeling and Control, Oxford University Press, 1ª Edição, 1992

[OPENUP 2007] Eclipse process framework team, OpenUP: A lightweight process framework, Disponível em <<http://openup.epfwiki.net/>>. Acesso em agosto de 2007.

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

[PMI 2004] Project Management Institute, A Guide to the Project Management Body of Knowledge, 3ª Edição, 2004.

[PRESSMAN 1997] Pressman, R. S. Software Engineering: A practitioner's approach. McGraw-Hill. 4ª Edição, 1997.

[RISING 2007] Rising, L. The Scrum Software Development Process. Disponível em: <http://members.cox.net/risingl1/articles/IEEEScrum.pdf>. Acesso em março 2007.

[SCHWABER 1996] Schwaber, Ken. "Controlled chaos: living on the edge". Disponível em: <http://www.controlchaos.com/old-site/ap.htm>. Acesso em maio de 2007.

[SCHWABER 2004] Schwaber, K. Agile Project Management with Scrum. Microsoft Press. 1ª Edição, 2004.

[SEI 1995] Software Engineering Institute. The Capability Maturity Model: Guidelines for Improving the Software Process. Reading, MA: Addison-Wesley, 1995.

[SEI 1999] SEI, Software Risk Evaluation (SRE) Method Description, v2.0, 1999, Disponível em <<http://www.sei.cmu.edu/pub/documents/99.reports/pdf/99tr029-body.pdf#search=%22software%20risk%20evaluation%22>>. Acesso em abril de 2007.

[SEI 2006] CMMI® for Development, Software Engineering Institute (SEI), Universidade de Carnegie Mellon, Versão 1.2, Agosto 2006. Disponível em: <<http://www.sei.cmu.edu/cmmi/>>. Acesso em maio 2007.

[SEI 2007] CMMI Website. Software Engineering Institute (SEI), Universidade de Carnegie Mellon, 2005. Disponível em: <<http://www.sei.cmu.edu/cmmi/>>. Acesso em maio 2007.

[SOFTEX 2006] Softex, MPS.BR - Melhoria de Processo do Software Brasileiro, Guia Geral, versão 1.1, 2006, ISBN 978-85-99334-03-4

[STANDISH GROUP 2001] Standish Group International, Inc., The, Extreme Chaos. Disponível em: <[http://www.standishgroup.com/sample\\_research/PDFpages/extreme\\_chaos.pdf](http://www.standishgroup.com/sample_research/PDFpages/extreme_chaos.pdf)>. Acesso em Outubro 2005.

[TURNER et al. 2002] Turner, R., and Jain, A., 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.

[ZUBROW 2003] 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.