

Estendendo o SCRUM segundo as Áreas de Processo de Gerenciamento de Projetos do CMMI

Ana Sofia Cysneiros Marçal

Universidade de Fortaleza, Mestrado em Informática Aplicada
Fortaleza - CE, Brasil, 60811-341
C.E.S.A.R – Centro de Estudos e Sistemas Avançados do Recife
Recife - PE, Brasil, 50.0303-90
ana.sofia@cesar.org.br

Bruno Celso Cunha de Freitas, Felipe Santana Furtado Soares, Teresa Maria Medeiros Maciel

C.E.S.A.R – Centro de Estudos e Sistemas Avançados do Recife
Recife - PE, Brasil, 50.0303-90
{bruno.freitas, felipe.furtado, teresa.maciел}@cesar.org.br

Arnaldo Dias Belchior

Universidade de Fortaleza, Mestrado em Informática Aplicada
Fortaleza - CE, Brazil, 60811-341
belchior@unifor.br

Abstract

In software development market, there is a strong tend to application agile development due to an accelerated rhythm of changes. This context has caused growing frustrations in organizations because of heavy plans, specifications and documentations, frequently imposed by the compliance criteria of maturity models. Software development organizations that have been employing the Capability Maturity Model such as CMMI for improving their processes are now increasingly interested in the possibility of adopting agile development methods. This work presents an approach of agile methodology Scrum compliant with CMMI project management process areas applied in an organization of innovation and research for developing software projects. This is useful for organizations that have the intention to adopt a new agile project management framework based on both CMMI and Scrum practices.

Keywords: Scrum, Agile Development Methods, CMMI, Agile Project Management.

Resumo

No mercado de desenvolvimento de software há uma forte tendência para o desenvolvimento ágil de aplicações devido a um ritmo acelerado de mudanças. Este contexto tem causado crescentes frustrações nas organizações devido aos planos, especificações e documentações pesados, muitas vezes impostos por critérios de conformidade dos modelos de maturidade. Organizações de desenvolvimento de software que vinham adotando modelos de capacidade de maturidade como o CMMI para a melhoria de seus processos estão cada vez mais interessadas na possibilidade de adoção de métodos ágeis. Este trabalho apresenta uma abordagem do método ágil *Scrum* compatível com áreas de Gerenciamento de Projetos do CMMI aplicada em uma organização de inovação e pesquisa no desenvolvimento de projetos de software. A extensão proposta para o *Scrum* pode contribuir de forma relevante para organizações que têm o propósito de adotar uma metodologia de gerenciamento de projetos ágil e que esteja compatível com práticas do CMMI.

Palavras chave: *Scrum*, Metodologias Ágeis, CMMI, Gerenciamento Ágil de Projetos.

1 Introdução

De acordo com Boehm [6], a partir de 2000 estamos vivendo uma tendência para o desenvolvimento ágil de aplicações devido a um ritmo acelerado de mudanças e inovações na tecnologia da informação, em organizações e no ambiente de negócios. Esse ritmo acelerado de mudanças tem causado frustrações crescentes com planos, especificações e documentações pesados muitas vezes impostos por critérios de conformidade dos modelos de maturidade.

Boehm [6] ainda cita que no final dos anos 90 acompanhamos o surgimento de vários métodos ágeis, entre eles: *Adaptive Software Development*, *Crystal*, *Dynamic Systems Development*, *eXtreme Programming (XP)*, *Feature Driven Development* e *Scrum*. Todos esses métodos empregam princípios ágeis, tais como ciclos iterativos, entrega rápida de software funcionando e simplicidade, como definido no Manifesto para Desenvolvimento Ágil [7] publicado em 2001. A essência desse movimento é a definição de novo enfoque de desenvolvimento de software, calcado na agilidade, na flexibilidade, nas habilidades de comunicação e na capacidade de oferecer novos produtos e serviços de valor ao mercado, em curtos períodos de tempo [14].

Ao lado do manifesto ágil, surge também o APM (*Agile Project Management*) que representa um conjunto de valores, princípios e práticas, que auxiliam a equipe de projeto a entregar produtos ou serviços de valor em um ambiente desafiador [14]. Como agilidade deve-se entender “a habilidade de criar e responder a mudanças, buscando a obtenção de lucro em um ambiente de negócio turbulento”; ou ainda, a capacidade de balancear a flexibilidade e a estabilidade [14]. Métodos, práticas e técnicas ágeis para desenvolvimento de software prometem incrementar a satisfação do cliente [5] produzindo software com maior qualidade e acelerando o tempo de desenvolvimento [3].

Assim sendo, organizações que empregaram um grande esforço na melhoria dos seus processos baseadas no CMMI (*Capability Maturity Model Integration*) [19], agora também acreditam que abordagens ágeis possam prover incrementos de melhorias [15].

Apesar da existência de características distintas entre os métodos ágeis 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 [22].

Davis [10] relata que, apesar de existir uma grande controvérsia entre a compatibilidade de ADM (*Agile Development Methods*) e o CMMI, eles não são mutuamente exclusivos. Complementa, explicando que há um lugar para ADM no CMMI e, o mais importante, aqueles que adotaram o CMMI, podem considerar a adição de ADM em seus processos. O caminho para alcançar mais agilidade com o CMMI é perceber que as práticas são primariamente consultivas ou indicativas e, que para corresponder a uma avaliação do CMMI, uma organização deve demonstrar que as metas de uma área de processo estão sendo alcançadas através de evidências de práticas [4].

Inserido neste contexto de controvérsias e compatibilidades entre CMMI e métodos ágeis, este trabalho apresenta uma extensão do *Scrum* segundo as áreas de Gerenciamento de Projetos do CMMI. Este estudo se inicia com um mapeamento entre o *Scrum* e o CMMI, avaliando-se o atendimento das práticas específicas do modelo restrito ao escopo de gestão de projetos. A partir desta avaliação, sugere-se a adoção de práticas complementares visando deixar o *Scrum* mais compatível com o CMMI, sem, no entanto, perder seus princípios ágeis. A extensão proposta para o *Scrum* pode contribuir de forma relevante para organizações que têm o propósito de adotar uma metodologia de gerenciamento de projetos ágil e que esteja compatível com práticas do CMMI.

Este trabalho está organizado da seguinte maneira: a Seção 2 apresenta uma visão geral CMMI; na Seção 3, é provida uma visão geral do *Scrum*; a Seção 4 compreende a análise realizada entre o *Scrum* e as práticas de gerenciamento de projetos do CMMI; na Seção 5, propõem-se extensões para o *Scrum* visando um maior grau de atendimento às práticas do CMMI; na Seção 6, realiza-se uma aplicação prática dessa proposta em um ambiente real de desenvolvimento de software. Finalmente, a Seção 7 apresenta as conclusões deste trabalho.

2 Enfoques sobre o CMMI

O CMMI-DEV [19] é uma abordagem de melhoria de processos que provê elementos essenciais para um processo efetivo. Reúne melhores práticas que endereçam atividades de desenvolvimento e manutenção, cobrindo todo o ciclo de vida de produto desde a sua concepção até a sua entrega e manutenção. É apresentado em duas representações: em estágios ou contínua. Cada representação organiza diferentemente as áreas de processo.

A representação contínua agrupa as áreas de processo por categorias de afinidade, com atribuição de níveis de capacidade para a melhoria de processos em cada área de processo. A representação em estágios, por sua vez, organiza as áreas de processo em 5 níveis de maturidade para suportar e guiar a melhoria dos processos. As áreas de processos podem ser agrupadas também em quatro categorias: Gerenciamento de Processos, Gerenciamento de Projetos, Engenharia, e Suporte.

Na categoria de Gerenciamento de Projetos, são englobadas atividades de gestão relacionadas com planejamento, monitoração e controle do projeto sendo composta pelas áreas de processo: nível 2 - Planejamento do Projeto (PP), Monitoramento e Controle do Projeto (PMC), Gerenciamento de Acordos com Fornecedores (SAM); nível 3 - Gerenciamento de Riscos (RSKM) e Gerenciamento Integrado do Projeto (IPM) + Desenvolvimento Integrado do Produto e do Processo (IPPD); nível 4- Gerenciamento Quantitativo do Projeto (QPM).

3 Enfoques sobre o SCRUM

O *Scrum* é um método que aceita que o desenvolvimento de software é imprevisível e formaliza a abstração, sendo aplicável a ambientes voláteis [1]. Ele se destaca dos demais métodos ágeis pela maior ênfase dada ao gerenciamento do projeto. Reúne atividades de monitoramento e feedback, em geral, reuniões rápidas e diárias com toda a equipe, visando à identificação e correção de quaisquer deficiências e/ou impedimentos no processo de desenvolvimento [21].

O *Scrum* assume a premissa de que o desenvolvimento de software é muito complexo e imprevisível para ser planejado totalmente inicialmente. Ao invés disso, deve-se usar controle do processo empírico para garantir a visibilidade, inspeção e adaptação [20].

O método baseia-se ainda, em princípios como: equipes pequenas de, no máximo, 7 pessoas; requisitos que são pouco estáveis ou desconhecidos; e iterações curtas. Divide o desenvolvimento em intervalos de tempos de no máximo 30 dias, também chamados de *Sprints*.

3.1 Papéis e Responsabilidades

O *Scrum* implementa um esqueleto iterativo e incremental através de três papéis principais [20]:

- *Product Owner*: representa os interesses de todos no projeto; define os fundamentos do projeto criando requisitos iniciais e gerais (*Product Backlog*), retorno do investimento (ROI), objetivos e planos de entregas; prioriza o *Product Backlog* a cada *Sprint*, garantindo que as funcionalidades de maior valor sejam construídas prioritariamente.
- *ScrumMaster*: Gerencia o processo do *Scrum*, ensinando o *Scrum* a todos os envolvidos no projeto e implementando o *Scrum* de modo que esteja adequado a cultura da organização; deve garantir que todos sigam as regras e práticas do *Scrum*; é responsável por remover os impedimentos do projeto.
- *Time*: desenvolve as funcionalidades do produto; define como transformar o *Product Backlog* em incremento de funcionalidades numa iteração gerenciando seu próprio trabalho sendo responsáveis coletivamente pelo sucesso da iteração e conseqüentemente pelo projeto como um todo.

3.2 Fases do Scrum

O *Scrum* possui um ciclo de vida composto por 4 fases [16] [2]:

- *Planejamento*: estabelecer a visão do projeto e expectativas garantindo recursos para a sua execução. Nesta fase são criadas as versões iniciais do *Product Backlog* e o plano de release, arquitetura de negócio e técnica em alto nível.
- *Stagging*: avaliar as várias dimensões do projeto criando itens adicionais ao *Product Backlog* relacionados com o tipo do sistema, time, ambiente de desenvolvimento, tipo de aplicação. Nesta fase os Times são formados e são construídos os mecanismos de comunicação e coordenação entre eles.
- *Desenvolvimento*: consiste de múltiplas *Sprints* para o desenvolvimento dos incrementos de funcionalidade do produto.
- *Releasing*: realizar a entrega do produto ao cliente.

3.3 Fluxo de Desenvolvimento

No *Scrum*, um projeto se inicia com uma visão do produto que será desenvolvido [20]. A visão contém a lista das características do produto estabelecidas pelo cliente, além de algumas premissas e restrições. Em seguida, o *Product Backlog* é criado contendo a lista de todos os requisitos conhecidos. O *Product Backlog* é então priorizado e dividido em *releases*. O fluxo de desenvolvimento detalhado do *Scrum* é mostrado na Figura 1.

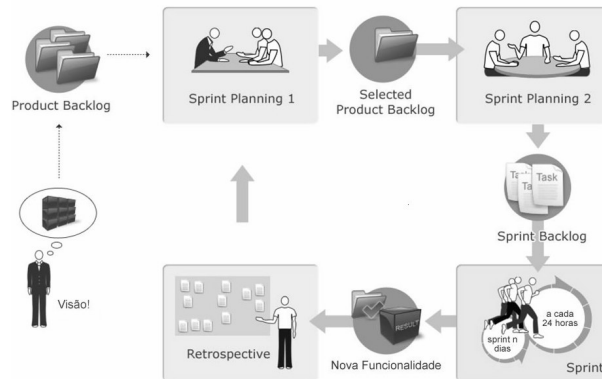


Figura 1. Visão geral do processo do Scrum (adaptada de [12])

No *Scrum*, são realizadas iterações chamadas de *Sprints*. Schwaber [20] explica que cada *Sprint* inicia-se com uma reunião de planejamento (*Sprint Planning Meeting*), na qual o *Product Owner* e o Time decidem em conjunto o que deverá ser implementado (*Selected Product Backlog*). A 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*. Nas primeiras *Sprints*, é realizada a maioria dos trabalhos de arquitetura e de infra-estrutura. A lista de tarefas pode ser modificada ao longo da *Sprint* pelo Time e as tarefas podem variar entre 4 a 16 horas para a sua conclusão.

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? 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. Em seguida o *ScrumMaster* conduz a reunião de retrospectiva (*Sprint Retrospective Meeting*), com o objetivo de melhorar o processo/time e/ou 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 ser feito, sendo um excelente mecanismo para visualizar a correlação entre a quantidade de trabalho que falta ser feita (em qualquer ponto) e o progresso do time do projeto em reduzir este trabalho.

4 Scrum versus Áreas de Gerenciamento de Projetos do CMMI

Neste trabalho, a avaliação do *Scrum* nas práticas do modelo CMMI corresponde a: Planejamento do Projeto (PP), Monitoramento e Controle do Projeto (PMC), Gerenciamento Integrado do Projeto (IPM) + Desenvolvimento Integrado do Produto e do Processo (IPPD) e Gerenciamento de Riscos (RSKM). Estas áreas de processo compõem os níveis 2 e 3 de maturidade da representação por estágios do modelo, versão 1.2, na categoria de Gerenciamento de Projetos.

Foram excluídas do escopo deste trabalho Gerenciamento de Acordos com Fornecedores (SAM) e Gerenciamento Quantitativo do Projeto (QPM), porque estas áreas de processo nem sempre são aplicadas a todas as organizações e têm uma importância menor dentro do contexto de gestão de projetos ágeis.

Para cada uma das Áreas de Processo do escopo foi realizada uma análise detalhada entre suas práticas específicas e as práticas gerenciais do *Scrum* classificando cada prática de acordo com os critérios definidos na Tabela 1.

Tabela 1. Critérios para classificação das áreas de processo.

Classificação		Critério
NS	Não Satisfeito	Não há evidências da prática no Scrum.
PS	Parcialmente Satisfeito	Há evidências da prática no Scrum, embora a prática não esteja plenamente atendida.
S	Satisfeito	A prática está totalmente atendida no Scrum.

Após a fase de classificação, foi calculado o percentual de satisfação de cada área de processo entre critérios definidos, tomando como base o número total de práticas específicas da área de processo. Em seguida, os resultados foram agrupados e foi gerada uma visão consolidada da cobertura do *Scrum* nas áreas de processo PP, PMC, RSKM e IPM + IPPD.

Nas subseções seguintes são apresentados, para cada área de processo do escopo do trabalho, os resultados gerais da análise realizada enfatizando os pontos nos quais o *Scrum* atende ou não diretamente as práticas do CMMI. Uma análise detalhada por subprática pode ser encontrada em Marçal [17].

4.1 Análise do Planejamento do Projeto

No Scrum, a definição do escopo inicial do projeto ocorre durante a fase de Planejamento onde todos os *stakeholders* podem contribuir com a criação do *Product Backlog*. Juntos, o documento de Visão e *Product Backlog* formam a base para a elaboração de um plano de projeto em alto nível compatível com a volatilidade de projetos ágeis.

O comprometimento do plano é realizado continuamente no início de cada iteração, durante a *Sprint Planning Meeting*. O *Product Owner*, *ScrumMaster*, e time em comum acordo, definem as prioridades do *Product Backlog* para cada *Sprint* e determinam que funcionalidades o time irá trabalhar na próxima *Sprint*.

As estimativas são realizadas em 2 níveis: *Product Backlog* e *Sprint Backlog*. As estimativas do *Product Backlog* são pouco precisas, de alto nível, tendo como propósito oferecer uma visibilidade do esforço de cada requisito. Já as estimativas do *Sprint Backlog* são mais precisas. Entretanto, as estimativas de esforço realizadas no *Scrum* não necessariamente seguem um método formal, nem tampouco são derivadas de um tamanho ou complexidade como sugerido pelo modelo CMMI.

Apesar de fazer menção ao uso de dados estimados em *Sprints* passadas, o *Scrum* não reforça a utilização de uma base histórica organizacional. O custo não é mencionado explicitamente no processo, mas somente o esforço, apesar de o custo ser necessário para o cálculo do orçamento do projeto e financiamento do mesmo pelo *Product Owner*.

A partir do *Product Backlog*, é capturado o cronograma que é priorizado é subdividido em *Sprints* considerando a alocação do time de acordo com sua capacidade de produção. O cronograma é então automaticamente obtido após esta divisão, considerando o número total de *Sprints* do projeto, já que cada *Sprint* tem a duração de 30 dias. A reconciliação do trabalho é realizada durante da *Sprint Planning*, quando o time, define, em conjunto com o *Product Owner* e *ScrumMaster* o máximo de funcionalidades que poderá ser implementada na *Sprint*.

A alocação do time e preparação da infra-estrutura do projeto são realizadas no início do projeto durante a fase de *Stagging* [2]. No *Product Backlog* são adicionados os recursos necessários ao desenvolvimento, tais como: máquinas, ferramentas e demais investimentos necessários para a configuração do ambiente de desenvolvimento. Além disso, o *ScrumMaster* é o responsável por prover os recursos necessários ao longo do projeto de acordo com necessidades e impedimentos que são reportados nas reuniões diárias.

O time, que é um grupo multifuncional, auto-gerenciado, deve na medida do possível ser configurado considerando as melhores pessoas, isto é, aquelas com maiores conhecimentos e habilidades (técnicos e de negócio) que são necessários para implementar o *Sprint Backlog*. Mas, se isso não é possível, necessidades de capacitação são incluídas no *Product Backlog*.

Uma vez que no *Scrum* um risco é um possível impedimento para o projeto, a identificação dos riscos é realizada de forma iterativa, durante as reuniões diárias do time sendo documentados em *whiteboards*, *flipcharts* e na lista de impedimentos (*Impediment Backlog*). Apesar de a identificação dos riscos ocorrer iterativamente, esta não ocorre, no entanto, de forma parametrizada e sistemática, utilizando, por exemplo, categorias e fontes de riscos como indicado no CMMI.

Considerando-se o aspecto da comunicação, percebe-se que as práticas e regras definidas no *Scrum* contribuem para uma boa comunicação e colaboração entre o time e os stakeholders, bem como para a visibilidade da condução e progresso do projeto. Os dados gerados no projeto podem ser colocados em uma pasta pública acessível por todos [20].

Muitas das informações do projeto são comunicadas face-a-face nas reuniões e outras são comunicadas através de documentos, e outras através de relatórios de progresso gerados ao final de cada *Sprint*. Entretanto, não existe um plano de comunicações que formalize como serão realizadas as coletas, consolidação e divulgação das informações e dados do projeto. A privacidade dos dados também fica comprometida. A Tabela 2 mostra o resultado final da classificação de cada prática específica da área de processo PP.

Tabela 2. Classificação da Área de Processo PP.

Objetivo Específico	Práticas Específicas	Classificação
SG 1 Estabelecer Estimativas	SP 1.1 Estimar o Escopo do Projeto	S
	SP 1.2 Estabelecer Estimativas de Atributos de Produtos de Trabalho e Tarefas	NS
	SP 1.3 Definir o Ciclo de Vida do Projeto	S
	SP 1.4 Determinar Estimativas de Esforço e Custo	PS
SG 2 Desenvolver um Plano de Projeto	SP 2.1 Estabelecer o Orçamento e o Cronograma	PS
	SP 2.2 Identificar os Riscos do Projeto	PS
	SP 2.3 Planejar o Gerenciamento de Dados	NS
	SP 2.4 Planejar os Recursos do Projeto	S
	SP 2.5 Planejar os Conhecimentos e Habilidades Necessários	S
	SP 2.6 Planejar o Envolvimento dos Stakeholders	S
	SP 2.7 Estabelecer o Plano do Projeto	S
SG 3 Obter Compromissos com o Plano	SP 3.1 Revisar Planos que Afetam o Projeto	S
	SP 3.2 Equilibrar Níveis de Trabalho e Recursos	S
	SP 3.3 Obter Compromissos com o Plano	S

4.2 Análise do Monitoramento e Controle do Projeto

O monitoramento do projeto é feito efetivamente através dos gráficos de *Burndown* e das reuniões de acompanhamento mencionadas anteriormente. O *Product BurnDown* mostra a velocidade com que o time está entregando os itens do *Product Backlog*. É analisado a cada término de *Sprint*. Ajuda a monitorar o planejamento das entregas das funcionalidades dando visibilidade se conseguiremos entregar o release ou se deveremos negociar a retirada de requisitos para conseguir entregar o release na data planejada.

Já o *Sprint Burndown* mostra diariamente ao time a velocidade e progresso da evolução das suas tarefas em uma *Sprint*, dando visibilidade se o time irá concluir suas tarefas no final da *Sprint*. O *Sprint Burndown* provê uma ferramenta de apoio ao planejamento de ações corretivas que são tomadas no projeto.

As reuniões de acompanhamento, sobretudo as reuniões diárias, permitem acompanhar o dia-a-dia de trabalho do time e perceber as dificuldades existentes para a realização das tarefas planejadas. Estas dificuldades devem ser rapidamente sanadas pelo *ScrumMaster* para que o time não perca seu foco e comprometa o objetivo da *Sprint*. Apesar disso, como as estimativas de custo, tamanho e esforço não são realizadas de maneira sistemática, não há um acompanhamento como solicitado no modelo CMMI.

O envolvimento dos *stakeholders* é monitorado naturalmente durante as reuniões do projeto, através do *ScrumMaster*, que é o responsável pelo entendimento e respeito às regras e práticas definidas no processo *Scrum* devendo fazer com que todos os envolvidos no projeto entendam suas práticas e regras. Apesar de não haver registro de documentação do monitoramento realizado.

Os compromissos de cada *Sprint* são estabelecidos durante a *Sprint Planning*, monitorados durante a execução da *Sprint* através da *Sprint Burndown* e das reuniões diárias e revistos na *Sprint Retrospective*. Durante uma *Sprint*, o time não pode receber trabalho adicional dos stakeholders e/ou *Product Owner*. Apenas o time pode alterar o *Sprint Backlog* o qual deve manter foco ininterrupto na realização das tarefas da *Sprint*.

Assim, as reuniões diárias buscam levantar impedimentos (problemas, dependências, riscos, etc.). Logo há uma identificação e monitoramento dos impedimentos pelo *ScrumMaster*. Entendemos que no *Scrum* as ações corretivas para os impedimentos levantados são tomadas. Entretanto não há registro de planos de ações corretivas, nem de documentação relativa a isso, nem tampouco análise de resultados para determinar a eficácia das ações corretivas tomadas.

Não há procedimento e planejamento formal no *Scrum* para a gestão dos dados do projeto o que prejudica o comprometimento do monitoramento dos dados como solicitado no CMMI. A Tabela 3 mostra o resultado final da classificação de cada prática específica da área de processo PMC.

Tabela 3. Classificação da Área de Processo PMC.

Objetivo Específico	Práticas Específicas	Classificação
SG 1 Monitorar o Projeto em relação ao Plano	SP 1.1 Monitorar os Parâmetros de Planejamento do Projeto	PS
	SP 1.2 Monitorar os Compromissos	S
	SP 1.3 Monitorar os Riscos do Projeto	PS
	SP 1.4 Monitorar o Gerenciamento de Dados	NS
	SP 1.5 Monitorar o Envolvimento dos Stakeholders	S
	SP 1.6 Conduzir Revisões de Progresso	S
	SP 1.7 Conduzir Revisões em Marcos	S
SG 2 Gerenciar Ações Corretivas até o Encerramento	SP 2.1 Analisar Problemas	S
	SP 2.2 Tomar ações corretivas	PS
	SP 2.3 Gerenciar ações corretivas	PS

4.3 Análise do Gerenciamento Integrado do Projeto + IPPD

Não é definido no *Scrum* um conjunto de processos padrões para a organização, apenas estabelece o conjunto de práticas e regras que devem ser usadas na execução do projeto, bem como atividades e caminhos que devem ser seguidos de acordo com o tipo do projeto.

Também não há nenhuma menção a estabelecimento e manutenção de um processo definido para o projeto, uso de base histórica de projetos e ativos do processo organizacional para realização do planejamento e estimativa do projeto, integração entre planos bem como contribuições com produtos de trabalho, medições e experiências para o processo organizacional.

Tabela 4. Classificação da Área de Processo IPM + IPPD.

Objetivo Específico	Práticas Específicas	Classificação
SG 1 Utilizar o Processo Definido do Projeto.	SP 1.1 Estabelecer o Processo Definido do Projeto.	NS
	SP 1.2 Utilizar os Ativos de Processos Organizacionais para o Planejamento das Atividades do Projeto.	NS
	SP 1.3 Estabelecer o Ambiente de Trabalho do Projeto.	NS
	SP 1.4 Integrar os Planos.	NS
	SP 1.5 Gerenciar o Projeto Utilizando os Planos Integrados.	NS
	SP 1.6 Contribuir para os Ativos de Processos Organizacionais.	NS
SG 2 Coordenar e Colaborar com os Stakeholders Relevantes.	SP 2.1 Gerenciar o Envolvimento dos Stakeholders.	S
	SP 2.2 Gerenciar Dependências.	PS
	SP 2.3 Resolver Questões de Coordenação.	PS
IPPD		
SG 3 Aplicar Princípios do Desenvolvimento Integrado do Produto e do Processo.	SP 3.1 Estabelecer a Visão Compartilhada do Projeto.	S
	SP 3.2 Estabelecer uma Estrutura Integrada para o Time.	S
	SP 3.3 Alocar Requisitos para times integrados.	S
	SP 3.4 Estabelecer times integrados	S
	SP 3.5 Garantir colaboração entre as interfaces dos times	S

Quando mais de um time trabalha simultaneamente no projeto, o projeto é chamado de projeto escalonado e os mecanismos usados para coordenar o trabalho desses times são chamados de mecanismos escalonados. Algumas práticas são críticas para o sucesso de projetos escalonados, dentre as quais [20]:

- Construa a infra-estrutura para o escalonamento antes de escalonar o projeto, que deve ser implementada por um único time inicial, em *Sprints* sucessivas até que a infra-estrutura esteja completa. Requisitos não funcionais para construir a infra-estrutura do escalonamento devem ser adicionados ao *Product Backlog* e devem ser priorizados juntamente com outras funcionalidades de negócio na fase de *Stagging* do *Scrum*, que antecede a primeira *Sprint*;
- Sempre entregue valor de negócio nas iterações que estão sendo realizadas para a construção da infra-estrutura;
- Aperfeiçoe as capacidades do time inicial e depois forme os times adicionais. Os times adicionais devem possuir, no mínimo, 1 membro do time inicial, atuando como especialista da infra-estrutura e arquitetura.

Estas práticas em conjunto, de certa forma estabelecem os requisitos necessários para o trabalho de times integrados do CMMI. A Tabela 4 mostra o resultado final da classificação de cada prática específica da Área de processo IPM + IPPD.

4.4 Análise do Gerenciamento de Riscos

A identificação de riscos no *Scrum* é realizada como impedimentos durante as reuniões diárias, porém não existem práticas explícitas para definir fontes, parâmetros e categorias de riscos que devem ser usados para analisar, categorizar bem como para controlar o esforço do gerenciamento dos riscos.

Também não há uma estratégia para o gerenciamento dos riscos, nem mesmo um plano de mitigação para os riscos mais importantes e uso de bases históricas. Assim sendo, a avaliação, categorização e priorização dos riscos ficam comprometidas. Para essa área de processo, todas as práticas específicas foram consideradas Não Satisfeitas, à exceção da prática SP 2.1 Identificar Riscos que foi considerada como Parcialmente Satisfeita.

4.5 Resultados da Avaliação

Na Figura 2, são apresentados os resultados gerais da avaliação realizada através de uma visão consolidada da cobertura do *Scrum* nas áreas de processo de PP, PMC, RSKM e IPM+IPPD da categoria de gerenciamento de projetos do CMMI.

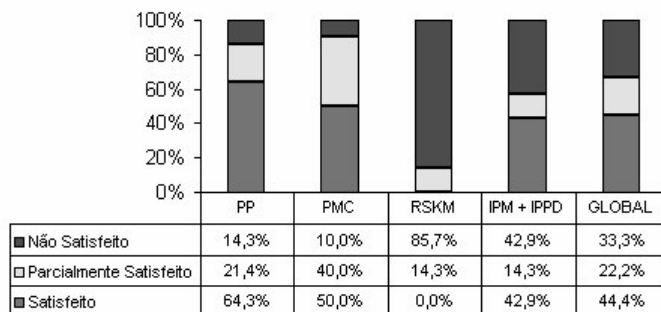


Figura 2. Cobertura do Scrum nas Áreas de Processo PP, PMC, RSKM e IPM+IPPD

Considerando-se estes resultados, tem-se que 44,4% das práticas específicas encontram-se na categoria Satisfeito, 22,2% na categoria Parcialmente Satisfeito e 33,3% na categoria Não Satisfeito. Percebe-se também que este resultado deve-se principalmente a:

- Ausência de técnicas ou práticas alternativas para a realização das estimativas do projeto. Isto afeta diretamente práticas de PP (SP 1.2 e 1.4) e PMC (SP 1.1).
- Ausência de um melhor gerenciamento dos riscos, impactando práticas de RSKM (todas), PP (SP 2.2) e PMC (SP 1.3).
- Lacunas no gerenciamento de ações corretivas de problemas e dependências, afetando as práticas relacionadas com o objetivo SG 2 de PMC e práticas de IPM (SP 2.2. e SP 2.3).
- Lacunas no planejamento e gerenciamento do orçamento do projeto, o que compromete práticas de PP (SP 2.1) e PMC (SP 1.1).

- Ausência de um planejamento e monitoramento dos dados do projeto, impactando a aderência às práticas de PP (SP 2.3) e PMC (SP 1.4).
- Lacunas no gerenciamento integrado do projeto devido à ausência de um processo integrado e definido que é adaptado a partir do conjunto de processos padrão da organização, conforme definido no objetivo SG1 de IPM + IPPD.

Além disso, as descobertas da avaliação revelam que parte das lacunas encontradas está relacionada com a falta de documentação (evidências escritas) na execução das atividades. Isto se deve a um dos valores do manifesto ágil que enfatiza “Software funcionando sobre documentação compreensiva”, significando que o nível de documentação deve ser definido como o mais baixo possível.

Entretanto, isso pode ser revisto, de forma que alguma documentação simples possa ser introduzida no *Scrum*, deixando-o assim mais em conformidade com o modelo CMMI. Práticas alternativas também podem ser analisadas e implementadas, o que aumentaria o grau de adequação do *Scrum* ao CMMI. Estas práticas, no entanto, precisam ser criticamente analisadas quanto ao seu atendimento ao modelo.

5 Estendendo o *Scrum*

Neste trabalho, a proposta de extensão do *Scrum* não abrange todas as lacunas do *Scrum* em relação às práticas de gerenciamento de projetos do CMMI. Focamos inicialmente na resolução das fraquezas relacionadas com as práticas de estimativas, gerenciamento de riscos e gerenciamento de problemas e suas ações corretivas, pois acreditamos que este subconjunto contribui para uma grande variação no resultado de cobertura do *Scrum* no CMMI. Posteriormente, esta proposta deverá complementada, visando compatibilidade plena do *Scrum* com as áreas de processo PP, PMC, RSKM e IPM + IPPD do CMMI.

5.1 Práticas para Estimativas de Complexidade por *Story Points*

A estimativa de complexidade por *Story Points* deve ser introduzida ao processo de estimativa e priorização dos itens do *Product Backlog* na *Sprint Planning* 1. A idéia por trás do método de estimativas por *Story Points* [9] é que o cliente e o time sejam parte fundamental das estimativas, na qual o primeiro considera o valor que cada item do *Product Backlog* agrega para o seu negócio, e o segundo dimensiona a complexidade de implementação de cada item baseado no contexto atual do projeto.

Uma vez que todo o *Product Backlog* sido valorado de acordo com o valor de negócio de cada item, a equipe do projeto estima a complexidade de cada um destes itens em *Story Points*. Para isto, podem utilizar a técnica *Planning Poker*, que segundo *Cohn* [9] é um método de atribuição de estimativas colaborativo, no qual cada integrante do time possui um conjunto de cartas identificadas com rótulos diferentes (por exemplo, com a seqüência de *Fibonacci*: 2, 3, 5, 8, 13, 21 e 34).

Inicialmente, o time identifica o item de *backlog* mais simples (para este item é atribuído o valor 2) que passa a ser o item de referência na estimativa dos demais. Em seguida, para os próximos itens do *Product Backlog*, cada integrante do time apresenta uma carta com o rótulo que define a complexidade daquele item em relação à referência pré-definida.

A estimativa do esforço necessária para desenvolver o projeto é derivada a partir do tamanho estimado em *Story Points* considerando-se a capacidade ou velocidade de produção do time (definida a partir de bases históricas e do resultado de *Sprints* passadas). Similarmente, a duração em dias do projeto e quantidade de *Sprints* é obtida dividindo-se o esforço total do projeto pela quantidade de participantes do time.

Um fluxo de atividades para realização de estimativas por *Story Points* no nível organizacional é proposto por *Freitas* [11] podendo o mesmo ser facilmente introduzido no *Scrum*. Observa-se ainda que a inclusão destas atividades no fluxo do *Scrum* contribui para que o mesmo passe a estar satisfeito com as práticas de PP (SP1.2 e 1.4), auxiliando também no monitoramento dos parâmetros do projeto conforme definido na prática de PMC (SP 1.1).

5.2 Práticas para o Gerenciamento de Riscos

O gerenciamento ágil de processos tem práticas que visam atender a dois desafios [18]: o primeiro é que elas devem integrar as atividades de gerenciamento de riscos dentro das atividades de planejamento da iteração; o segundo é que elas devem adaptar as práticas de gerenciamento de riscos de forma que todo o time possa realizá-las rapidamente.

Para que as práticas do RSKM sejam compatíveis com o *Scrum*, bem como as práticas de PP e PMC relacionadas com riscos, é necessário que sejam incluídas em seu fluxo atividades para identificação, análise, priorização e mitigação dos riscos com a criação de planos de ação para tratar os riscos de maior prioridade. Essas atividades devem ser inseridas no contexto das reuniões do *Scrum*, conforme descritas a seguir:

1. *Identificar Riscos*: a identificação dos riscos deve acontecer: a) durante a *Sprint Planning 1*, com foco nos itens do *Product Backlog* com maior valor de negócio. A identificação dos riscos pode ser realizada através do método de *Brainstorming* [18]. Um *checklist* contendo fontes e categorias de riscos pode ser introduzido visando facilitar esta tarefa; b) durante as *Daily Scrum*, como possíveis impedimentos para o projeto. Riscos identificados devem ser adicionados ao *Impediment Backlog*.
2. *Analisar Riscos*: a análise de riscos deve ser realizada durante a *Sprint Planning 1* e compreende etapas para determinar: (i) impacto (alto, médio e baixo); (ii) probabilidade (alta, média e baixa); e (iii) o fator de exposição do risco, obtido a partir do produto entre o seu impacto e a sua probabilidade. Alguns ajustes no fator de exposição do risco podem ser aplicados para tratar aspectos específicos do projeto e de qualidade como definidos por Chin [8]. A análise dos riscos nos ajuda a estabelecer uma importância relativa entre os mesmos e é útil na priorização dos riscos.
3. *Priorizar os Riscos*: seguindo um princípio ágil, Preston [18] sugere que os top 3-5 riscos sejam priorizados. Esses riscos são considerados os mais urgentes e por isso mesmo devem ser gerenciados e mitigados na próxima *Sprint*. Os demais riscos ficam “guardados” devendo ser reavaliados na próxima reunião de *Sprint Planning*. A priorização dos riscos deve ocorrer na reunião de *Sprint Planning 1*, auxiliando na priorização e seleção dos itens de *backlog* que serão desenvolvidos na próxima *Sprint*.
4. *Criar planos de ações*: os planos de ação correspondem a todas as tarefas (ações) que devem ser executadas para mitigar os riscos, isto é, tarefas que devem ser executadas visando reduzir o fator de exposição do risco (probabilidade e/ou impacto). Este plano deve ser criado durante a *Sprint Planning 2*, em conjunto com identificação das tarefas que serão executadas para implementar os itens do *Selected Product Backlog*. Assim sendo, as ações entram no *Sprint Backlog* e devem ser realizadas e monitoradas continuamente pelo time durante as reuniões diárias do *Scrum*. Durante a execução da *Sprint*, riscos podem se transformar em problemas, e nestes casos, passam a ser tratados como impedimentos no *Scrum*. Planos de ações de contingência, seguindo uma abordagem ágil, são elaborados (na *Sprint Planning 2*) apenas para os riscos priorizados e com fator de exposição alto e devem ser reavaliados à medida que os riscos transformam-se em problemas, sendo tratados pelo *ScrumMaster* que é o responsável pela resolução dos impedimentos.
5. *Monitorar continuamente os riscos*: os riscos são monitorados diariamente, nas *Daily Scrum Meetings*, e também durante as *Sprint Plannings*, quando devem ser reavaliados em conjunto com os demais riscos identificados. É importante observar que para garantirmos a agilidade, apenas um subconjunto dos riscos está sendo monitorado a cada *Sprint*. Este subconjunto é representado pelos riscos que foram priorizados e que estão diretamente relacionados com os itens do *Selected Product Backlog*.

5.3 Práticas para o Gerenciamento de Ações Corretivas

De acordo com Chin [8], identificar e monitorar sistematicamente os problemas do projeto, bem como suas ações corretivas facilita a resolução eficiente de problemas críticos e é necessário para um bom gerenciamento do projeto. Práticas para o gerenciamento de ações corretivas de problemas conforme sugeridas em PMC devem ser acrescidas às práticas já existentes no *Scrum* para a identificação e resolução dos impedimentos, conforme descritas a seguir:

1. *Analisar Problemas*: os problemas identificados durante as reuniões diárias devem ser devidamente registrados pelo *ScrumMaster* no *Impediment Backlog*, juntamente com uma análise dos mesmos a qual inclui informações sobre: (i) prioridade de resolução (alta, média e baixa); (ii) responsável pela resolução; (iii) data de identificação; (iv) data alvo para a resolução; (v) observações e (vi) estado do problema (aberto, fechado, cancelado). A análise dos problemas determina ajuda na comparação entre os problemas e na identificação da necessidade de tomada de ações corretivas para a sua resolução.
2. *Tomar Ações Corretivas*: seguindo o princípio ágil, ações corretivas devem ser identificadas pelo *ScrumMaster*, com a colaboração do time para os problemas/dependências mais prioritários. Uma lista das ações corretivas relacionadas com os problemas identificados deve ser documentada, incluindo informações que permitam o seu monitoramento, tais como: (i) descrição do item de ação; (ii) o responsável item de ação; (iii) a data de abertura; (iv) a data alvo para a resolução; (v) observações; (vi) estado da ação (aberta, fechada, cancelada).

3. *Gerenciar Ações Corretivas*: o gerenciamento das ações corretivas deve ser introduzido nos seguintes momentos: (i) durante as *Daily Scrum Meetings*, quando os responsáveis pela resolução de problemas devem relatar o que fizeram e o estado atual do problema. O registro do monitoramento deve ser documentado; (ii) durante as reuniões de retrospectiva (*Sprint Retrospective Meeting*), com o objetivo avaliar a eficácia das ações corretivas tomadas para os problemas identificados, melhorando assim, o gerenciamento de problemas para a próxima *Sprint*.

6 Uma Aplicação da Extensão do Scrum

A proposta de extensão do *Scrum* está sendo inicialmente aplicada em dois projetos reais de desenvolvimento de software em um instituto de inovação e pesquisa no Brasil considerando apenas as práticas de estimativas de complexidade por *Story Points*.

Os dois projetos em estudo (que tiveram como requisito do cliente o uso método ágil *Scrum* para o seu gerenciamento), são similares em termos de plataforma tecnológica (.NET), no tempo estimado para a sua execução (seis meses) bem como no tamanho da sua equipe (dez pessoas). Diferem, entretanto, nos seus objetivos. O Projeto A visa o desenvolvimento de uma aplicação de *Customer Relationship Management* (CRM). Já o Projeto B consiste na manutenção e desenvolvimento de novos requisitos para uma ferramenta case de apoio ao planejamento, projeto e execução de testes de software. Outro aspecto relevante sobre os projetos diz respeito à falta de experiência das equipes com a metodologia ágil *Scrum* e com a tecnologia adotada.

Vários resultados práticos foram encontrados pelos times de desenvolvimento dos projetos com o uso desta extensão do *Scrum* com o uso de estimativas por *Story Points*, entre os quais se pode destacar:

- Não é necessária tanta informação para começar a estimar. Precisamos apenas de uma referência – o item de *Backlog* com menor complexidade, e a partir daí podemos realizar a estimativa dos demais itens através de comparações com este item.
- Maior comprometimento do time na hora da execução e cumprimento das tarefas estimadas já que as estimativas foram realizadas colaborativamente pelo próprio time.
- Não é preciso se preocupar com horas para a estimativa dos itens de *Backlog*. Isso facilita o processo de estimativa, uma vez que estamos apenas trabalhando com uma complexidade, ou melhor, *Story Points*.
- O esforço do projeto pode ser facilmente encontrado, por derivação, considerando-se a velocidade/capacidade de implementação de uma *Story Point* em um dia de trabalho pelo time. Desta forma, automaticamente derivamos a quantidade de *Sprints* que serão realizadas no projeto.
- A estimativa por *Story Points* ajuda a priorizar os itens do *Product Backlog*, e a selecionar os itens que devem compor o *Selected Product Backlog*.

A comparação das estimativas entre projetos distintos foi uma das dificuldades encontradas na aplicação realizada. Isso se deve ao fato de que a estimativa por *Story Points* é baseada em uma referência para o projeto (o item de *backlog* mais simples de ser desenvolvido), e que provavelmente não será a mesma referência para um outro projeto.

A precisão também foi outro fator de dificuldade percebido pelas equipes durante a realização das estimativas. Estimativas das primeiras *Sprints* foram pouco precisas ocasionando em horas adicionais de trabalho. Mas, à medida que o time foi se conhecendo melhor, se integrando e se sentindo mais confortável com o domínio da aplicação e com a tecnologia as estimativas tornaram-se mais precisas.

7 Conclusões

O *Scrum* não cobre diretamente todas as práticas específicas de gerenciamento de projeto do CMMI. No entanto, pode ser um ótimo ponto de partida para empresas que não possuem processos definidos e têm times pequenos, pois consegue criar grande parte dos fundamentos necessários para que estas empresas institucionalizem as Áreas de Processo de Gestão de Projetos do nível 2 do CMMI, sem comprometer a agilidade de que necessitam. Por outro lado, para empresas que buscam níveis de maturidade maiores, o *Scrum* pode ser uma excelente “largada”, mas este necessita da adição de práticas complementares, como as sugeridas neste trabalho.

Com poucas adaptações, relacionadas ao gerenciamento ágil dos riscos, gerenciamento de ações corretivas e adição de um método de estimativas, o *Scrum* passa a ficar muito mais compatível ao CMMI,

podendo o mesmo ainda ser estendido para atender as demais áreas de processo que não foram tratadas neste trabalho.

No entanto, práticas alternativas podem ser identificadas, considerado as recomendações do *Scrum*, de acordo com a realidade de cada organização, o que minimiza o número de práticas não diretamente atendidas.

Referências

- [1] Advanced Development Methods – ADM, Controlled chaos: living on the edge, <http://www.controlchaos.com/old-site/ap.htm>, (Dezembro 2006).
- [2] Advanced Development Methods – ADM, Scrum Methodology – Incremental, Iterative Software Development from Agile Processes”, Revision 0.9, 2003.
- [3] Anderson, D. J. Agile Management for Software Engineering - Applying the Theory of Constraints for Business Results, Prentice Hall, 2003.
- [4] Anderson, D. J., Stretching Agile to fit CMMI Level 3, presented at Agile 2005 Conference, Dever, http://www.agilemanagement.net/Articles/Papers/Agile_2005_Paper_DJA_v1_5.pdf, 2005.
- [5] Boehm, B. and Turner, R., Balancing Agility and Discipline: A Guide for the Perplexed, AddisonWesley, 2003.
- [6] Boehm, B., A View of 20th and 21st Century Software Engineering, ICSE 2006.
- [7] Beck, K. et al., Manifesto for Agile Software Development, <http://www.agilemanifesto.org/>, (Dezembro 2006).
- [8] Chin, G., Agile Project Management: How to Succeed in the Face of Changing Project Requirements. Amacon, 2004.
- [9] Cohn, M., Agile Estimating and Planning, Prentice Hall, 330 p, 2006.
- [10] Davis, C., Glover, M., Manzo, J. and Opperthausen D., An Agile Approach to Achieving CMMI, <http://www.agiletek.com/thoughtleadership/whitepapers/>, (Março 2007).
- [11] Freitas, B., Marçal, A., Soares, F., Tenório, L. e Maciel, T., Adaptando Story Points para Estimativas de Software no Nível Organizacional, 2007.
- [12] Gloger, B., The Zen of Scrum, <http://www.glogerconsulting.de>, (Março 2007).
- [13] Goldenson, D. and Gimbson, D., Demonstrating the Impact and Benefits of CMMI: An Update and Preliminary Results, CMU/SEI-2003-SR-009. SEI, 2003.
- [14] Highsmith, J., Agile Project Management – Creating Innovative Products, AddisonWesley, 2004.
- [15] McMahan, P. E., Extending Agile Methods: A Distributed Project and Organizational Improvement Perspective, CrossTalk, The Journal of Defense Software Engineering, vol. 18, issue 5, pp. 1619, 2005.
- [16] Larman C., Agile & Iterative Development, A Manager’s Guide, Addison-Wesley, 2004.
- [17] 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, MD, USA, (6-8 March 2007).
- [18] Preston, S. and Pichler, R., Agile Risks, Agile Rewards, <http://www.ddj.com/dept/architect/184415308>, (Janeiro 2007).
- [19] Software Engineering Institute - SEI, CMMI-DEV: CMMI for Development, V1.2 model, CMU/SEI-2006-TR-008, <http://www.sei.cmu.edu/cmmi/general/>, (Dezembro 2006).
- [20] Schwaber, K., Agile Project Management With Scrum, Microsoft, 2004.
- [21] Schwaber, K. and Beedle, M., Agile Software Development With Scrum. NJ: Prentence Hall, 2002.
- [22] 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.