

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
MBA EM GESTÃO DA TECNOLOGIA DA INFORMAÇÃO E
COMUNICAÇÃO**

JOICYELLEN PEREIRA

**ESTUDO DA APLICAÇÃO DE METODOLOGIA ÁGIL PARA ACELERAR O
PROCESSO DA MATURIDADE EM PROCESSOS DE TI NA PEQUENA
EMPRESA.**

MONOGRAFIA

CURITIBA – PR

2019

JOICYELLEN PEREIRA

**ESTUDO DA APLICAÇÃO DE METODOLOGIA ÁGIL PARA
ACCELERAR O PROCESSO DA MATURIDADE EM PROCESSOS
DE TI NA PEQUENA EMPRESA.**

Monografia apresentada como requisito parcial para obtenção do título de Especialista em Gestão da Tecnologia da Informação e Comunicação da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Roberto Candido
Coorientador: Prof. Dr. Alexandre Miziara

CURITIBA – PR

2019



TERMO DE APROVAÇÃO

Estudo da aplicação de metodologia ágil para acelerar o processo de maturidade em processos de TI na pequena empresa.

Por

Joicyellen Pereira

Esta monografia foi apresentada às **16 h** do dia **13/05/2019** como requisito parcial para a obtenção do título de Especialista no CURSO DE ESPECIALIZAÇÃO EM GESTÃO DE TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO, da Universidade Tecnológica Federal do Paraná, **Câmpus Curitiba**. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho:

1		Aprovado
2		Aprovado condicionado às correções Pós-banca, postagem da tarefa e liberação do Orientador.
3		Reprovado

Prof. _____
UTFPR - Examinador

Prof. Roberto Candido
UTFPR – Orientador

Prof. Msc. Alexandre Jorge Miziara
UTFPR – Coordenador do Curso

AGRADECIMENTOS

A Deus, por ter me dado forças para trilhar esse caminho. Ao meu amor e companheiro, Daniel Ristoff, por ter me motivado e se disposto a me ajudar no que fosse possível, além de todo amor, compreensão e carinho. Ao longo de todo trabalho, contei com o auxílio do Prof. Roberto Candido, que na condição de meu orientador, me direcionou através de suas revisões e correções, proporcionando melhor clareza para a estruturação do estudo. Ao coordenador do curso, Alexandre Miziara, que pacientemente viabilizou e apoio para que apresentação fosse possível.

RESUMO

PEREIRA, Joicyellen. **Estudo da aplicação de metodologia ágil para acelerar o processo da maturidade em processos de TI na pequena empresa.** 2019. 47 f. Monografia (MBA em Gestão da Tecnologia da Informação e Comunicação - Universidade Tecnológica Federal do Paraná. Curitiba, 2019.

Pequenas empresas voltadas ao desenvolvimento de software tem buscado cada vez mais a adesão de metodologias ágeis de desenvolvimento para se tornarem competitivas frente ao mercado, porém, muitas vezes, as práticas ágeis acabam deixando os projetos mais distantes dos objetivos definidos no planejamento do projeto, diminuindo a previsibilidade de sua execução e conseqüentemente impactando na qualidade do produto desenvolvido. O CMMI-DEV - *Capability Model Integration for Development* – se mostra como alternativa aos problemas de maturidade para essas empresas, propondo um modelo reconhecido mundialmente, que destaca a priorização e formalização dos processos, enquanto o Scrum destaca o foco nos indivíduos e sua comunicação. Este trabalho apresenta alguns conceitos de desenvolvimento de software, gestão de projetos e metodologia ágil, complementado por um estudo de aderência entre estes dois modelos distintos, metodologia ágil Scrum junto ao CMMI, aplicados a uma empresa de desenvolvimento de software de pequeno porte para obtenção de maturidade nível 2 em seus processos na prática de Monitoração e Controle do Projeto (PMC), demonstrando o resultado dessa aplicação nas *Sprints* através do gráfico de *Burndown* do *Scrum*.

Palavras-chave: Metodologia Ágil. Modelo de trabalho para integração entre Scrum e CMMI. Maturidade de Software. Maturidade em Pequenas Empresas. Processo de Desenvolvimento.

ABSTRACT

PEREIRA, Joicyellen. **Study of the application of agile methodology to accelerate the process of maturity in IT processes in small business.** 2019. 47 p. Monograph (MBA in Information Technology and Communication Management - Federal University of Technology - Paraná. Curitiba, 2019.

Small software development companies have increasingly sought adherence to agile development methodologies to become competitive in the face of the market; however, agile practices often leave projects more distant from the objectives defined in project planning, the predictability of its execution and consequently impacting the quality of the product developed. The CMMI-DEV - Capability Model Integration for Development - is shown as an alternative to the maturity problems for these companies, proposing a globally recognized model that emphasizes the prioritization and formalization of processes, while Scrum emphasizes the focus on individuals and their communication. This work presents some concepts of software development, project management and agile methodology, complemented by a study of adherence between these two distinct models, agile Scrum methodology with the CMMI, applied to a small software development company to obtain maturity level 2 in their Project Monitoring and Control (PMC) practice, demonstrating the result of this application in Sprints through the Scrum Burndown chart.

Keywords: Agile Methodology. Working model for integration between Scrum and CMMI. Software Maturity. Maturity in Small Businesses. Development Process.

LISTA DE ILUSTRAÇÕES

Figura 1 - Ciclo de vida de um projeto subdividido Grupos de Processos.....	15
Figura 2 - As 10 Áreas de Conhecimento.....	16
Figura 3 - Custos de mudanças em diferentes ambientes de desenvolvimento.....	17
Figura 4 - Representação do Ciclo de Vida Scrum.	19
Figura 5 - Níveis de maturidade da representação por estágio CMMI.....	22
Figura 6 - As três dimensões críticas.....	23
Figura 7 - Áreas de Processo na representação Contínua.....	24
Figura 8 - Áreas de Processo na representação por Estágio	24
Figura 9 - Cronograma da Sprint 10 do projeto.....	38
Figura 10 - Gráfico de Burndown da Sprint antes das práticas do CMMI.....	39
Figura 11 - Gráfico de Burndown da Sprint após práticas do CMMI.	39

LISTA DE TABELAS

Tabela 1 - Áreas de processo CMMI e seus níveis de maturidade.....	25
--	----

LISTA DE ABREVIATURAS E SIGLAS

CMM – *Capability Maturity Model*

CMMI – *Capability Maturity Model for Integration*

CMMI-DEV - *Capability Model Integration for Development*

PMBOK - *Project Management Body of Knowledge*

PMC - *Project Monitoring and Control*

PMI - *Project Management Institute*

SEI - *Software Engineering Institute*

SW-CMM - *Capability Maturity Model for Software*

TI – *Tecnologia da Informação*

SUMÁRIO

1. INTRODUÇÃO	12
1.1. OBJETIVO GERAL.....	13
1.2. OBJETIVOS ESPECÍFICOS	13
1.3. JUSTIFICATIVA	13
2. REFERENCIAL TEÓRICO	14
2.1. GERENCIAMENTO DE PROJETO	14
2.2. GERENCIAMENTO ÁGIL	17
2.2.1. Scrum.....	18
2.2.2. Papéis do Scrum.....	18
2.2.3. Ciclo de vida Scrum	19
2.2.4. Artefatos do Scrum	20
2.3. CMMI-DEV	21
2.3.1. Tipo de Representação	23
2.3.2. Componentes das Áreas de Processo	25
2.3.3. Níveis de Maturidade	26
3. PROPOSTA DE MODELO.....	28
3.1. MONITORAR OS PARÂMETROS DE PLANEJAMENTO DO PROJETO	28
3.2. MONITORAR OS COMPROMISSOS	29
3.3. MONITORAR OS RISCOS DO PROJETO	30
3.4. MONITORAR O GERENCIAMENTO DE DADOS	30
3.5. MONITORAR O ENVOLVIMENTO DOS <i>STAKEHOLDERS</i>	31
3.6. CONDUZIR REVISÃO DE PROGRESSO.....	32
3.7. CONDUZIR REVISÃO DE MARCOS.....	32
3.8. ANALISAR PROBLEMAS	32
3.9. TOMAR AÇÕES CORRETIVAS	33
3.10. GERENCIAR AÇÕES CORRETIVAS	33
4. ANÁLISE E APLICAÇÃO.....	34
4.1. MONITORAR OS PARÂMETROS DE PLANEJAMENTO DO PROJETO	34
4.2. MONITORAR OS COMPROMISSOS	34
4.3. MONITORAR OS RISCOS DO PROJETO	35
4.4. MONITORAR O GERENCIAMENTO DE DADOS	35
4.5. MONITORAR O ENVOLVIMENTO DOS <i>STAKEHOLDERS</i>	36
4.6. CONDUZIR REVISÃO DE PROGRESSO.....	36
4.7. CONDUZIR REVISÃO DE MARCOS.....	36
4.8. ANALISAR PROBLEMAS	37
4.9. TOMAR AÇÕES CORRETIVAS	37

4.10. GERENCIAR AÇÕES CORRETIVAS	37
5. CONCLUSÃO E RESULTADOS.....	38
5.1. TRABALHOS FUTUROS.....	40
REFERÊNCIAS.....	41

1. INTRODUÇÃO

A evolução tecnológica em produtos, serviços ou processos de uma empresa que atua na área da computação é mensurada de acordo com os processos de desenvolvimento de software. Dentro do processo de desenvolvimento existem várias etapas pelas quais a concepção deverá passar até que esteja finalizada e seja entregue a fase de produção e o sistema esteja aderente a aplicação no negócio da empresa (TONINI, 2008).

Segundo o CMMI-DEV (2006) a Engenharia de Software baseia-se na organização de todo o processo, ou ciclo de vida, de um software desde sua concepção até o suporte e essa atividade pode tornar-se complexa e acarretar problemas dependendo do nível de maturidade da empresa que precisar implementá-lo.

Observando o mercado de TI, nota-se que comumente as novas empresas que estão ingressando neste ramo procuram investir seus esforços em personalização de produtos para que atendam as necessidades de seus clientes e deixam de lado os próprios processos internos; na maior parte dos casos, numa expansão futura esta situação acarreta diversos problemas relacionados a qualidade do produto, desempenho do processo (improdutividade) e insatisfação do cliente (SANTOS, 2007).

De acordo com McMahon (2005), visando obter maior desempenho em sua área, as empresas do ramo de desenvolvimento de software têm optado cada vez mais pela implementação de modelos de qualidade voltados a maturidade de processos, como por exemplo, o *Capability Maturity Model for Software* (SW-CMM) e o *Capability Model Integration for Development* (CMMI-DEV).

Ahern, Clouse e Turner (2008) ressaltam que, a maior parte das empresas que aderem ao CMMI como modelo de processo, os objetivos para tal adequação são: criação produtos e/ou serviços de qualidade; aumento do grau de satisfação do cliente; reduzir custos, além de estabelecer e monitorar métricas de eficiência como ferramenta de gestão. Além dos aspectos citados (que são internos á empresa), Hansson et al. (2006) acrescenta que, há situações específicas em que a escolha pelo CMMI pode ter sido uma imposição de clientes externos que, ao procurarem empresas fornecedoras de software, dão preferência as que possuam certificação CMMI devido ao reconhecimento mundial.

Para aliviar a carga de documentação e especificações necessárias para o cumprimento dos modelos de maturidade as organizações também estão aderindo a técnicas ágeis de desenvolvimento. Assim, as frustrações com as documentações são diminuídas através de ciclos

iterativos, entrega de valor ao cliente mais rápidas e simples, de acordo com a definição publicada no manifesto ágil em 2001 (BECK, 2001).

1.1. OBJETIVO GERAL

Estudar a aplicação da metodologia ágil em projetos de TI vinculados a aceleração dos processos de maturidade em uma pequena empresa para a área de Monitoramento e Controle.

1.2. OBJETIVOS ESPECÍFICOS

- Realizar pesquisa bibliográfica a respeito da metodologia ágil em gestão de projetos;
- Estudar o modelo de maturidade CMMI-DEV;
- Propor procedimentos ágeis do Scrum para processos de Monitoramento e Controle de Projetos de TI;
- Avaliar o impacto da proposta de agilidade Scrum na análise de maturidade corporativa;
- Criar framework para aplicação em casos reais de desenvolvimento.

1.3. JUSTIFICATIVA

O mercado consumidor cada vez mais busca qualidade, não é diferente nos processos de desenvolvimento de TI que seguem normatizações internacionais inclusive.

A fim de conquistar elevar o nível de maturidade em seus processos internos de desenvolvimento de software, diversas empresas tem aplicado os padrões de processos propostos pelo CMMI e como consequência positiva do aumento dessa qualidade no produto entregue, o custo de produção é diminuído e a satisfação do cliente é aumentada.

O padrão de maturidade descrito pelo CMMI-DEV não define “como” essa alteração no processo deve ser feita para que um nível mais alto de maturidade seja atingido, fazendo com que algumas empresas fiquem inertes frente ao número de documentos criados para atender o modelo proposto. Com o intuito de reduzir esta carga de documentação que engessa o processo interno, este trabalho se utilizará da metodologia ágil Scrum adaptada para estabelecer como atender a área de Monitoramento e Controle de Projetos do CMMI-DEV.

2. REFERENCIAL TEÓRICO

Neste capítulo apresenta-se o referencial teórico dos principais modelos de processo de projetos, modelo de maturidade CMMI e a abordagem de gerenciamento ágil Scrum.

2.1. GERENCIAMENTO DE PROJETO

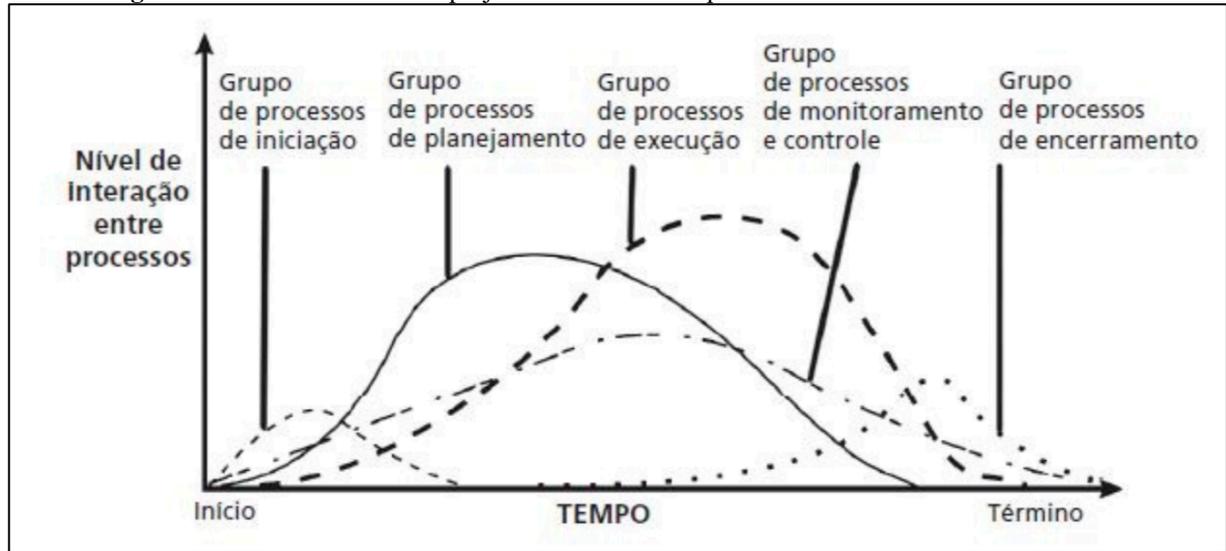
A multinacionalização de empresas dos mais variados segmentos econômicos tem feito com que empresas deixem de ser reconhecidas somente em sua região de localização geográfica e as tornas conhecidas e presentes no mercado global. Para tal feito estas empresas passaram a realizar planejamentos e estratégias focando o futuro, tendo a TI como sua base.

Projetos são os meios pelos quais empresas atingem suas metas e objetivos empresariais, constantemente dentro dos planos estratégicos e a equipe (ou departamento) de TI necessita implementá-los com eficácia e eficiência. De acordo com o PMI (PMI 2012, p. 3) um projeto “*é um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo*”. A execução de projetos de maneira satisfatória necessita de gerenciamento, que pode ser entendido como a aplicação do conhecimento, de habilidades, de ferramentas e técnicas as atividades relacionadas ao projeto para que seus requisitos sejam adequadamente atendidos (PMBOK, 2017).

Para alcançar um melhor controle das incertezas que surgem ao longo do projeto, sua condução é dividida em fases que são caracterizadas por entregas ou finalização de um determinado trabalho e está diretamente relacionado a natureza do projeto (VIEIRA, 2007.p.73); qualquer entrega deve ser tangível e seu valor deve ser facilmente identificado como, por exemplo, a entrega de um relatório (VARGAS, 2009).

De maneira genérica, o projeto é composto por fases e o agrupamento das fases de um projeto é nomeada de ciclo de vida, conforme o ilustrado na Figura 1 (VIEIRA, 2007.p.73).

Figura 1 - Ciclo de vida de um projeto subdividido Grupos de Processos.



Fonte: PMI®, PMBOK® Guide, 4rd ed., Project Management Institute, 2009.

As cinco fases ou grupos de processos (conforme denominado pelo PMBOK 6ª edição), podem ser conferidos a seguir:

- **Inicialização:** fase inicial do projeto, quando as necessidades são identificadas e estruturadas, documentos iniciais são confeccionados e as estratégias são analisadas e a melhor abordagem é selecionada. Nesta etapa estudos de viabilidade são executados alinhados com qualidade, prazos e custos para a concretização do projeto.
- **Planejamento:** é a fase responsável pelo detalhamento de tudo que será necessário para a execução do projeto como, por exemplo, cronogramas, relação entre as atividades, alocação de recursos, identificação das partes interessadas e envolvidas, análise aprofundada de custos com base no escopo, etc, para que, ao final dessa fase haja insumos para a execução da fase posterior com o mínimo de imprevistos. Nesta fase, todos os interessados devem ser envolvidos, inclusive o cliente.
- **Execução:** fase concretiza aquilo que foi planejado e documentado anteriormente e qualquer erro cometido nas fases anteriores fica evidente neste. Por ser a maior fase do projeto, grande parte dos recursos de esforço, tempo e custo é consumida nesta fase.

- **Monitoramento e controle:** fase vital que garante qualidade ao produto que será entregue e ocorre paralelamente às outras fases do projeto; objetiva acompanhar e controlar aquilo que está sendo realizado no projeto em tempo hábil para solucioná-los.
- **Encerramento:** após a execução dos trabalhos a saída de cada fase anterior é avaliada de acordo com as métricas estabelecidas no planejamento, os documentos de projetos são encerrados e as falhas ocorridas são analisadas e documentadas para que o know-how adquirido fique de Lições Aprendidas para projetos futuros.

Além dos grupos de processos descritos acima, o guia PMBOK apresenta o agrupamento das dez áreas de conhecimentos distintas, que representam um conjunto sólido de conceitos, termos e atividades que integram o campo do gerenciamento de projetos (PMBOK, 2017). Conforme a Figura 2, as dez áreas do conhecimento são:

Figura 2 - As 10 Áreas de Conhecimento.



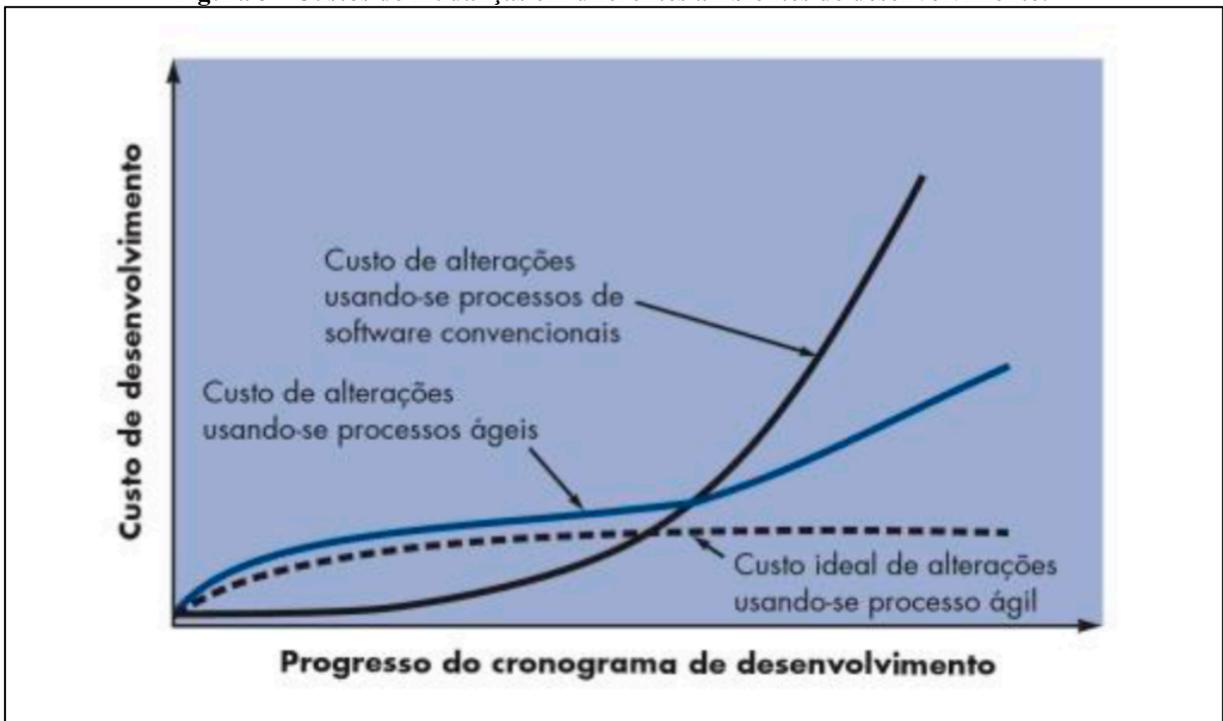
Disponível em: <https://escritoriodeprojetos.com.br/guia-de-gerenciamento-de-projetos/areas-de-conhecimento-em-gerenciamento-de-projetos>.

2.2. GERENCIAMENTO ÁGIL

O termo agilidade em engenharia de software significa a capacidade de responder e adequar as mudanças de forma rápida ao projeto (Jacobson, 2002).

A gestão de projetos utilizando metodologias ágeis tem apresentado diversos benefícios aos projetos como a redução de custos, por exemplo, pois, ao permitir que mudanças sejam implementadas em qualquer etapa do desenvolvimento poucos impactos serão refletidos no tempo de entrega ou no custo total do projeto (PRESSMAN, 2011). A Figura 3 apresenta a comparação entre os custos das mudanças em projetos que utilizam metodologias ágeis e os que utilizam métodos clássicos.

Figura 3 - Custos de mudanças em diferentes ambientes de desenvolvimento.



Fonte: Pressman (2010).

Como exemplos de metodologias ágeis temos *Kanban*, *Scrum*, *Extreme Programming (XP)*, *Lean Software Development* e Desenvolvimento Dirigido à Funcionalidade, onde destaca-se o método *Scrum* como o mais utilizado, que objetiva entregar ao cliente o mais alto valor possível.

2.2.1. Scrum

Scrum é um framework ágil que oferece suporte ao desenvolvimento e manutenção de produtos complexos por meio de práticas e regras simples que compreende controle (orientação) de requisitos de transparência, inspeção e adaptação, para que agregue ao produto o maior valor possível (SCHWABER, 2008).

2.2.2. Papéis do Scrum

O Scrum dispõe seis papéis que detêm tarefas e propósitos diferentes ao longo do processo de desenvolvimento do projeto. Segundo Franco (2007), os papéis são os seguintes:

- **Product Owner (P.O.) - Cliente:** representado por uma pessoa do lado do cliente, é o responsável por guiar o projeto e por tornar a lista de funcionalidades visível (gerenciando o backlog), maximizando o valor do produto. O P.O. é o responsável pela priorização e detalhamento das tarefas Sprint a Sprint, além de garantir que o projeto entregue o maior retorno sobre o investimento possível.
- **Scrum Master:** responsável por assegurar que o Scrum seja executado de forma apropriada, interagindo tanto com os clientes e gerente de projetos quanto com a Equipe Scrum ao longo de todo o projeto para que sigam a teoria, regras e práticas da metodologia.
- **Time Scrum - Equipe de Desenvolvimento:** é a equipe responsável pela execução e entrega do projeto Sprint a Sprint de acordo com o priorizado pelo P.O. No Scrum, os times de desenvolvimento são auto gerenciáveis, ou seja, eles gerenciam seu próprio trabalho.

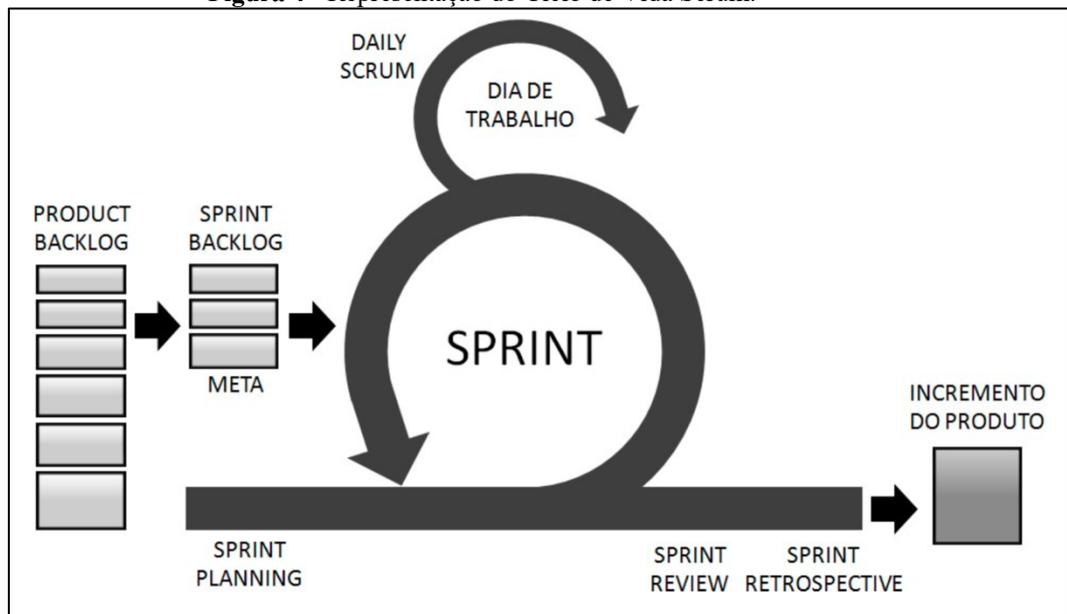
A equipe é envolvida nas seguintes etapas: Reunião de estimativa de esforço, criação e revisão da lista de funcionalidades do produto e possíveis impedimentos.

2.2.3. Ciclo de vida Scrum

Para reduzir a necessidade de reuniões não planejadas o Scrum possui alguns eventos definidos em sua metodologia e cada um deles possui o papel de inspecionar e adaptar o processo e/ou o produto, pois, são planejados para proporcionar transparência no andamento do projeto (KNIBERG, 2007).

Na Figura 4 é ilustrado o ciclo de vida do Scrum através dos seguintes artefatos:

Figura 4 - Representação do Ciclo de Vida Scrum.



Fonte: Schwaber & Beedle; Schwaber (2002; 2004, *apud* KNOWLEDGE21).

- **Sprint:** são períodos de tempo em que as partes do projeto, priorizadas do Backlog do Produto para o Backlog da Sprint, são desenvolvidas. As Sprints são compostas por Reunião de Planejamento, Revisão da Sprint e Retrospectiva da Sprint.
- **Sprint Planning Meeting:** no início de cada Sprint é necessário realizar o evento de *planejamento* para definir o escopo da Sprint. Nesta reunião participam o P.O., o Scrum Master e, esporadicamente, a equipe de desenvolvimento que juntos definem e esclarecem o trabalho a ser realizado durante a Sprint, com a duração máxima de 8 horas. As atividades priorizadas nesta reunião serão implementadas pelo Time de Desenvolvimento. A reunião de Planejamento da Sprint também pode ser dividida em duas partes sendo a primeira conhecida como *Sprint Planning Meeting* onde o PO, o Scrum Master

e o Time de desenvolvimento definem “o que” é para ser feito e a *Sprint Planning Meeting 2* onde o Scrum Master e o time de Desenvolvimento definem “como” as atividades deverão quebradas e desenvolvidas.

- **Daily Scrum Meeting:** durante a Sprint, todos os membros da equipe se reúnem diariamente, em pé, para uma reunião com *timebox* de 15 minutos que objetiva alinhar o conhecimento entre o time e identificar para remover possíveis impedimentos. Para isso, cada integrante da equipe deve responder três perguntas padrão: “O que eu fiz ontem?”, “O que farei hoje?” e “Há algum impedimento?”; essa reunião também auxilia no acompanhamento do desenvolvimento da Sprint. Qualquer pessoa da empresa pode participar da reunião, porém, apenas o *Time de Desenvolvimento*, *Scrum Master* e o *Product Owner* devem falar.
- **Sprint Review Meeting:** ao final de cada Sprint a reunião de Revisão da Sprint é realizada para que o *Product Owner* os demais interessados avaliem, levantem críticas e sugestões para o desenvolvimento do produto com base nas funcionalidades implementadas; as sugestões, se aceitas pelo P.O., serão adicionadas ao *Backlog* do Produto. Para esta apresentação é necessário que o Time de Desenvolvimento disponibilize um entregável no ambiente de homologação.
- **Sprint Retrospective Meeting:** o evento ocorre após a Revisão da Sprint com o objetivo de avaliar a Sprint entregue, observando os pontos positivos e negativos, para que sejam documentados e utilizados de lições aprendidas para as próximas *Sprints* do projeto.

2.2.4. Artefatos do Scrum

Do mesmo modo que os eventos (reuniões) do Scrum, os artefatos foram projetados para entregar maior transparência na troca de informações relacionadas a entrega do incremento do produto (SCHWABER, 2004). De acordo com o apresentado por Filho (2008) os principais elementos de apoio do Scrum são os *cards* que agrupam com funcionalidade e gráficos com informações dos *Backlogs* atualizados com frequência para refletirem o real estado do andamento do projeto, conforme o descrito a seguir:

- **Product Backlog:** é a lista de requisitos (funcionalidades) que o produto deve entregar e que ainda não foram desenvolvidas, também chamados de itens de *Backlog*. O responsável por gerencia-lo é o *Product Owner*.
- **Sprint Backlog:** grupo de itens que foram priorizados do *Product Backlog* pelo cliente para que seja executado na Sprint. Os itens selecionados não poderão ser alterados no decorrer da Sprint.
- **Product Burndown:** este artefato gráfico é utilizado para acompanhar o progresso do projeto em relação ao produto final. Ele apresenta o trabalho executado versus o trabalho restante para que seja possível acompanhar o cronograma de entrega do projeto em tempo real.
- **Release Burndown:** com a mesma lógica empregada no *Product Burndown*, este é voltado ao acompanhamento do progresso de uma Release, acompanhando seu cronograma de entrega.
- **Sprint Burndown:** com a mesma logica empregada no *Product Burndown*, porém, objetiva acompanhar as horas estimadas para a Sprint e as horas que ainda restam.
- **Taskboard:** é apresentado em um quadro com colunas que representam os status das tarefas, permitindo seu acompanhamento de forma clara por todos os envolvidos. No *taskboard* também pode ser implementado o gráfico de *Burndown*.

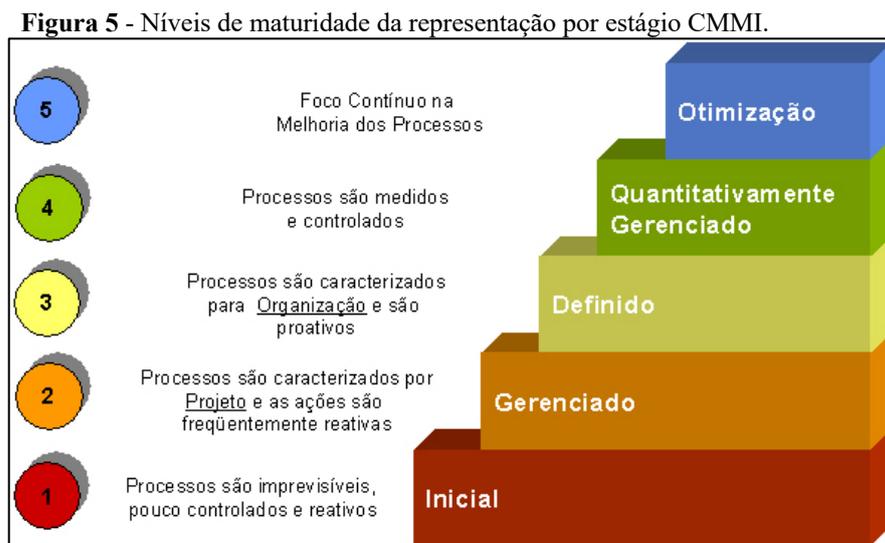
2.3. CMMI-DEV

Voltado a processos organizações existem vários modelos de maturidade e metodologias que visam auxiliar na melhoria contínua de processos. Entretanto, grande parte das abordagens presentes no mercado está focada em uma parte específica do negócio e não no negócio de forma panorâmica, criando barreiras para serem seguidos pelas organizações. Para evitar essas barreiras o SEI (*Software Engineering Institute*) desenvolveu um *framework* para guiar e aprimorar os processos organizacionais e competências das empresas para gerenciar o desenvolvimento, aquisição e o manutenção de processos e serviços.

Segundo Tonini, Carvalho e Spinola (2008), maturidade só é atingida de forma gradativa e com persistência, o que implica que a empresa só estará apta a gerenciar seus projetos se os fizer de forma sistemática.

O CMMI propõe às empresas um *framework* estruturado com práticas já efetivadas pelo mercado, onde estabelece prioridade para melhoria e guias para execução do processo de melhoria (CMU/SEI, 2006), sendo assim o CMMI-DEV (CMMI para desenvolvimento) reúne as melhores práticas relacionados ao desenvolvimento e suporte de software (*CMMI® for Development*, 2006).

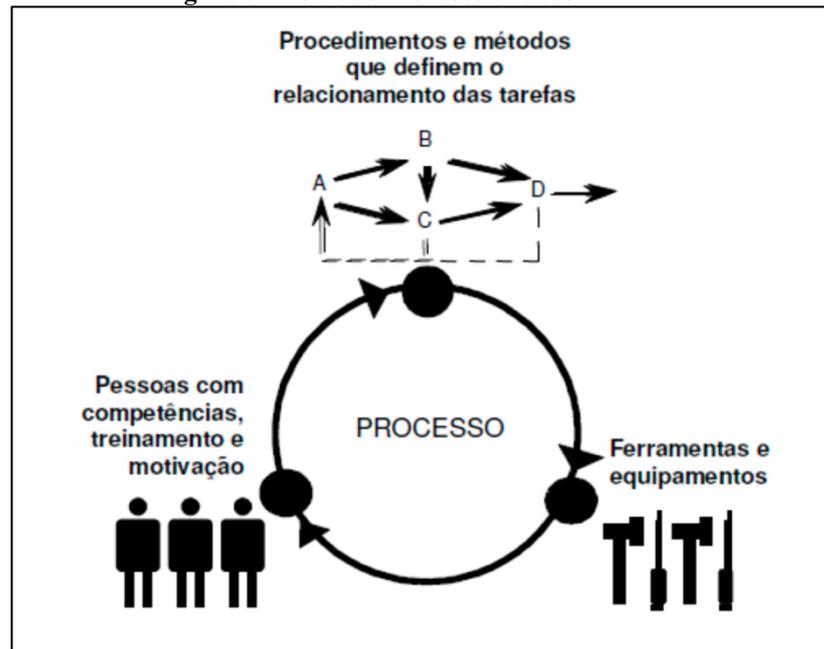
Para a implementação do CMMI é necessário a utilização do método de avaliação estabelecido pelo SEI (SCAMPI – SEI Members of the Assessment Method Integrated Team, 2001), acompanhado por um avaliador credenciado, para identificar o estágio ou nível do CMMI que a empresa encontra-se; após a identificação do nível de maturidade atual, a empresa buscará identificar e adquirir as competências necessárias para atingir o próximo nível, acarretando em uma melhoria natural na qualidade dos processos de desenvolvimento de produtos e serviços entregues pela empresa. Os níveis da representação por estágio estão dispostos na Figura 5.



Fonte: ISDBrasil (2012).

Em uma empresa há três áreas básicas e críticas nas quais as empresas devem buscar melhorias para se desenvolver em produtividade e tornarem-se mais competitivas: Pessoas, Tecnologias e Processos, conforme o apresentado na Figura 6.

Figura 6 - As três dimensões críticas



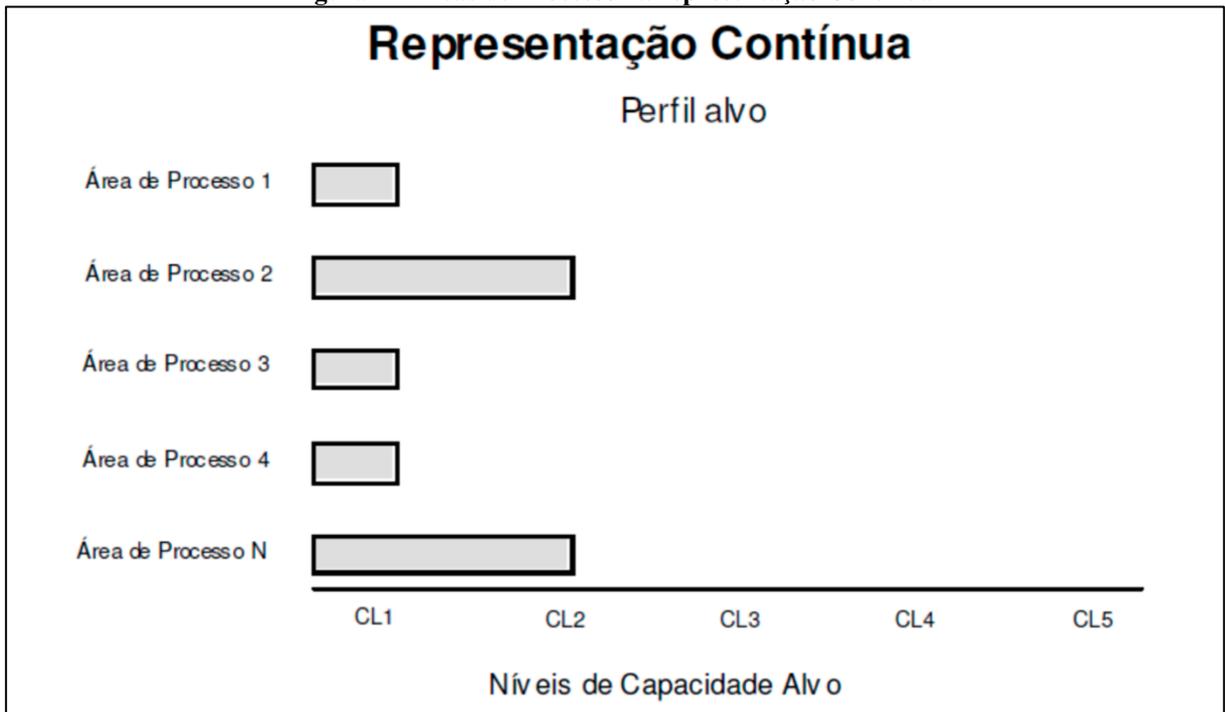
Fonte: CMMI® for Development (2006)

Um ponto relevante sobre o CMMI é ser um modelo de processo e não um processo, sendo assim, ele descreve as práticas necessárias para obter-se um processo mais eficiente e maduro e conseqüentemente elevar a qualidade de produtos e serviços.

2.3.1. Tipo de Representação

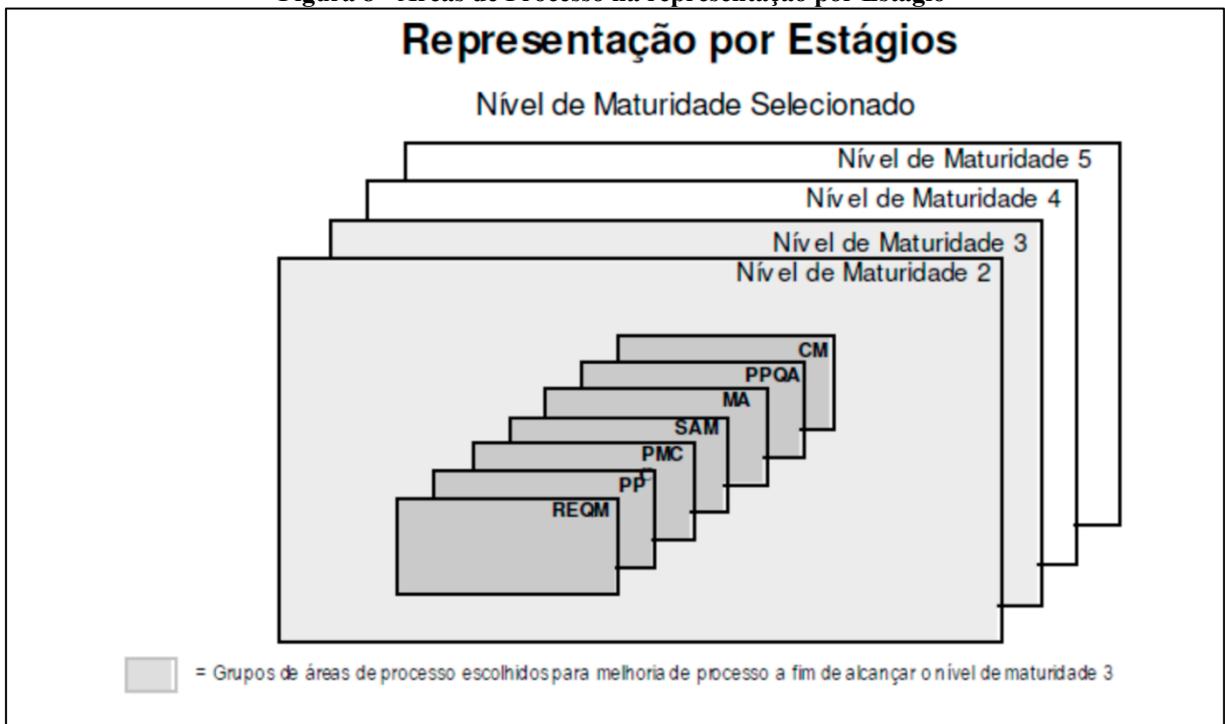
O *framework* CMMI-DEV utiliza dois tipos diferentes de representação para melhoria e avaliação dos processos: por estágio e contínua. No tipo Representação Contínua (Figura 7) a empresa tem como alvo a melhoria por áreas de processo de forma individual, utilizando seus níveis de capacidade, conforme sua estratégia. Já na representação por estágio (Figura 8) são usados conjuntos pré-definidos de processos para as áreas em que a empresa busca melhoria, de acordo com o nível de maturidade que ela deseja obter, ajudando na melhoria do nível da empresa como um todo (CMMI® for Development, 2006).

Figura 7 - Áreas de Processo na representação Contínua



Fonte: CMMI® for Development (2006)

Figura 8 - Áreas de Processo na representação por Estágio



Fonte: CMMI® for Development (2006)

2.3.2. Componentes das Áreas de Processo

O CMMI-DEV compreende 22 áreas de processos e seus níveis de maturidade conforme o apresentado a seguir na Tabela 1.

Nível de Maturidade	Área do Processo
2	Monitoração e Controle do Projeto (PMC)
2	Planejamento do Projeto (PP)
2	Gerência de Requisitos (REQM)
2	Análise e Medição (MA)
2	Garantia da Qualidade do Processo e do Produto (PPQA)
2	Gerência de Configuração (CM)
3	Gerência de Fornecedor Integrada (SAM)
3	Gerência de Projeto Integrada (IPM)
3	Gerência de Riscos (RSKM)
3	Definição do Processo Organizacional (OPD)
3	Foco no Processo Organizacional (OPF)
3	Treinamento Organizacional (OT)
3	Desenvolvimento de Requisitos (RD)
3	Integração do Produto (PI)
3	Solução Técnica (TS)
3	Validação (VAL)
3	Verificação (VER)
3	Análise de Decisão e Resolução (DAR)
4	Gerência Quantitativa do Projeto (QPM)
4	Desempenho do Processo Organizacional (OPP)
5	Inovação Organizacional e Posicionamento Estratégico (OID)
5	Resolução e Análise causal (CAR)

Tabela 1 - Áreas de processo CMMI e seus níveis de maturidade.
Fonte: A autora.

Cada área de processo apresentada reúne algumas práticas que se bem aplicadas auxiliarão a empresa a melhorar seu processo. Uma área de processo é composta por componentes que são fragmentadas em três categorias, sendo elas:

- **Requeridos:** são componentes requeridos que retratam o que a organização deverá implementar para determinada área de processo e por isso serão usados

como critério de avaliação na hora de decidir se a melhoria está ou não implementada.

- **Esperados:** são componentes que a empresa espera uma possibilidade de implantação para que um objetivo específico (requerido) seja satisfeito (INSTITUTE, 2010).
- **Informativos:** conjunto de práticas genéricas que oferece suporte as empresas satisfazerem a implementação dos componentes requeridos e esperados.

2.3.3. Níveis de Maturidade

Maturidade em gestão de projetos para o desenvolvimento de sistemas e processos pode ser entendida como sendo um conjunto de atividades, de natureza repetitiva e que asseguram alta qualidade e probabilidade de sucesso (KERZNER, 2006), ou seja, atingir um nível em que os padrões de gestão e controle de projetos abranjam a empresa como um todo, deixando de depender exclusivamente da ação de alguns profissionais (CRISSIS; KONRAD; SCHRUM, 2003).

O modelo proposto pelo CMMI possibilita a quantificação do grau de maturidade em que uma empresa se encontra por meio da avaliação de seus processos, utilizando a possibilidade de serem gerenciados como um parâmetro de avaliação (CMMI® for Development, 2006). Esses “graus de maturidade” são representados em 5 níveis chamados de: Iniciantes, Gerenciado, Definido, Gerenciado Quantitativamente e Otimizado.

- **Iniciante – Nível 1:** os processos não são passíveis de gerenciado e mesmo sendo possível que atinjam seus objetivos há um déficit em seu controle. Neste nível não é possível ter uma previsibilidade do que ocorrerá ao final do processo e a normalidade dá-se de forma reativa, conhecido como *Fire Fighting*; a eficiência e efetividade é baixa.
- **Gerenciado – Nível 2:** a evolução do 1 é caracterizada principalmente pelos processos serem gerenciáveis e mais bem definidos e. O perfil dos recursos envolvidos no projeto passa a ser cada vez mais qualificado e específico na função que cada um atua. Como resultado positivo desse melhor gerenciamento os projetos passam a ter uma taxa maior de sucesso nos quesitos: custo, qualidade e prazos; como ponto negativo é que a empresa fica com seu processo depende desses recursos mais capacitados.

- **Definido – Nível 3:** esse nível identifica como “definidos” os processos que podem ser definidos, compreendidos e aplicados, de forma padronizada e sincronizada, na empresa como um todo. Ao contrário do nível inicial (nível 1), neste, a empresa aplica os processos de forma proativa evitando que tenha que “apagar incêndio” a cada problema.
- **Gerenciado quantitativamente – Nível 4:** processos passam a ser quantificados, ou seja, metas de qualidade e desempenho são estabelecidas para que seja possível mensurá-los. Após a avaliação de qualidade e desempenho a empresa poderá ter base para ser mais assertiva nas estimativas de prazo e cronograma, tendo condições de antever estes acontecimentos com base em dados estatísticos.
- **Em Otimização – Nível 5:** a principal diferença do nível 5 para seu predecessor é que neste, a empresa está focada na melhoria contínua de seus processos utilizando as métricas quantitativas como referência.

Conforme Hsueh (2008) o CMMI pode ser usado para aprimoramento do processo através de projetos, de uma área ou da organização de forma completa. O tratamento para essas empresas é mapear os processos seguindo os seguintes modelos: processo, verificação de processo, transformação, validação, educação e execução de projeto piloto. O CMMI pode ser implementado continuamente ou subdividido em fases.

3. PROPOSTA DE MODELO

Neste capítulo será apresentada o resultado da aderência da metodologia ágil Scrum ao processo da empresa na busca pela certificação CMMI-DEV.

De acordo com Sutherland, um dos criadores do Scrum, o Scrum aliado ao CMMI resulta em uma combinação mais robusta nos quesitos de adaptabilidade e de previsibilidade que se comparado a cada um de forma isolada (SUTHERLAND et al., 2007). O modelo CMMI pode auxiliar empresas ágeis a institucionalizar seus métodos ágeis de maneira mais consistente e também a entender que processos sobrescrever, para que os processos de desenvolvimento se tornem mais dinâmicos e flexíveis. A dinâmica defendida pelo Scrum permite aquilo que Schwaber (2004) denomina de controle empírico do processo.

Antes da empresa analisada incluir a prática ágil a seus processos fez-se necessário a realização de uma análise detalhada da aderência entre as práticas gerenciais da metodologia Scrum e as práticas específicas da área de Monitoramento e Controle de Projetos para entender até que ponto a metodologia ágil comportaria os processos da área. Assim sendo, os tópicos a seguir explanam os resultados dessa análise, evidenciados os pontos em que a metodologia Scrum atende ou não as práticas do CMMI para a área de PMC.

Práticas de monitoramento analisadas:

3.1.MONITORAR OS PARÂMETROS DE PLANEJAMENTO DO PROJETO

- **CMMI-DEV:** o monitoramento e controle de parâmetros tem o intuito de monitorar e controlar os parâmetros de projeto como escopo, tarefas, estimativas, orçamento e cronograma definidos no plano de gerenciamento de projeto; para tal prática é necessário indicações que assegurem que o projeto foi bem monitoramento ao longo de seu ciclo de vida, visando comparar os parâmetros planejados com os valores reais e atuais do projeto (WANGENHEIM 2008).
- **Scrum:** no Scrum o monitoramento é praticado com base nos gráficos de *Product Burndown*, *Release Burndown*, *Sprint Burndown* e das *Daily Scrum Meetings*. O gráfico *Product Burndown* permite o acompanhamento da evolução da equipe com relação ao escopo do projeto (mapeado através do *Product Backlog*), ajudando no rastreamento das entregas e funcionalidades conforme o planejado para apoiar a tomada de decisão dos requisitos que

poderão ser entregues na *release* e, se necessário, fazer a negociação dos requisitos que não puderam ser entregues. O gráfico *Sprint Burndown* permite o acompanhamento diário da evolução das tarefas com relação ao escopo de uma Sprint, proporcionando uma visão mais a curto prazo do que o time tem condições de entregar ou não.

- **Plano de Ação:** utilizar as reuniões de acompanhamento, em destaque a diária, para monitorar o dia-dia do trabalho executado pelo time de desenvolvimento; nela problemas e impedimentos são levantados e reportados ao Scrum Master que poderá tratá-los de maneira preventiva para que não reflitam negativamente no objetivo da Sprint.

No que diz respeito a projetos que buscam elevar seu nível de maturidade e utilizam dos métodos ágeis, tendo em vista a ausência de planos mais específicos e detalhados, os critérios de conformidade como planejamento são incompletos, já que o monitoramento oferecido pelos gráficos de *Burndown* partem de elementos como escopo e tempo, trazendo alguns limitadores importantes para o controle e por isso precisam de adaptações para que atendam a prática por completo Highsmith (2009).

3.2.MONITORAR OS COMPROMISSOS

- **CMMI-DEV:** monitorar os compromissos do time referente a execução do projeto conforme o identificado no plano do projeto é a objetivo desta prática. As evidências geradas ao fim desse processo permitirão o controle dos compromissos assumidos entre as partes interessadas e possível revisão deles, conforme necessário.
- **Scrum:** os compromissos são assumidos de forma gradual, ou seja, a cada Sprint por meio da *Sprint Planning* onde o time cria e aceita, junto ao *Product Owner*, monitorados através do *Sprint Burndown* e das *Daily Scrum Meetings* e, por fim, revistos na *Sprint Retrospective*.
- **Plano de ação:** com base nos artefatos gerados em cada uma das fases citadas foi entendido que o Scrum pode ser utilizado de maneira satisfatória para a esta prática, sem necessidade de ajustes.

3.3.MONITORAR OS RISCOS DO PROJETO

- **CMMI-DEV:** a prática de monitorar os riscos do projeto tem o objetivo de monitorar os riscos identificados no documento de planejamento do projeto. As evidências criadas para essa prática permitem o acompanhamento dos riscos ao longo do ciclo de vida do projeto.
- **Scrum:** do ponto de vista ágil, reuniões diárias são realizadas para encontrar problemas e/ou riscos para que o monitoramento seja feito através do *taskboard*, mas esse registro dos no *taskboard*, por si só, não garante o monitoramento da resolução ou mitigação dos itens identificados. Em suma, não há uma abordagem específica no *Scrum* que comporte explicitamente a gestão de riscos, requerendo uma formalidade maior das iterações que resultam em identificação, análise, priorização ou mesmo mitigação desse risco.
- **Plano de ação:** para essa prática, adaptar o levantamento de riscos com a equipe nas *Daily Scrum Meetings*, *Retrospective Meeting*, ou *Planning Meeting*.

3.4.MONITORAR O GERENCIAMENTO DE DADOS

- **CMMI-DEV:** o PMI (2013) apresenta uma abordagem sobre processos que determina às áreas específicas aspectos como: cronograma, custos, escopo, qualidade, riscos e *stakeholds*; por outro lado a abordagem apresentada pelo CMMI-DEV considera peculiaridades da área de desenvolvimento de sistemas. Desta forma, o CMMI (*Software Engineering Institute [SEI]*, 2011) estabelece a atividade de monitoramento de dados que consiste basicamente em: análise de problemas, tomada de ações corretivas e o gerenciamento de ações corretivas. Como finalidade, essa prática objetiva monitorar o gerenciamento de dados planejados no planejamento do projeto. Para esta é esperado evidências que permitam certificar que o plano de dados e planejado esteja sendo seguido.
- **Scrum:** o Scrum não apresenta nenhuma prática específica ou planejamento de dados do projeto diretamente, sendo assim, não há evidência para o monitoramento dos dados como o CMMI-DEV especifica. Em vista disso, é necessário adequação no Scrum para que se torne aderente a prática.

- **Plano de ação:** para essa prática, a medida adotada é assegurar que todos os recursos envolvidos no projeto tenham acesso as informações essenciais, por meio de *kanban*, para que as atividades sejam realizadas de forma auto gerenciável.

3.5.MONITORAR O ENVOLVIMENTO DOS *STAKEHOLDERS*

- **CMMI-DEV:** o monitoramento do envolvimento dos *stakeholders* do projeto exerce um papel importante na gestão, pois, garante a participação dos envolvidos e interessados no projeto conforme o planejado.
- **Scrum:** no *framework* Scrum não há uma abordagem específica para a criação da evidência exigida nessa prática.
- **Plano de ação:** como estratégia de adaptação, o *Scrum Master* é o responsável por monitorar constantemente o engajamento e envolvimento das partes interessadas no projeto utilizando as cerimônias que compõem a metodologia. Outras ações importantes para garantir a melhor fluidez nesta prática é o cuidado no recrutamento e capacitação de recursos para alguns papeis chave como *Product Owner*, por exemplo, assegurando que a pessoa responsável possua o profundo conhecimento do negocio do cliente e a disponibilidade necessária para o compartilhamento dessas informações e priorização das tarefas de acordo com o valor que elas geram.

3.6.CONDUZIR REVISÃO DE PROGRESSO

- **CMMI-DEV:** a finalidade dessa prática é manter a verificação constante do progresso, desempenho (evolução) e questões do projeto, tal prática necessita evidências que garantam essa constante verificação.
- **Scrum:** para essa prática não é necessário adaptação na metodologia, pois, ela suporta os requisitos do CMMI-DEV.
- **Plano de ação:** a definição criada para atender a essa prática consiste em utilizar as *Daily Scrum Meetings* e *Sprint Review* para a coleta e documentação de dados e métricas que embasem a medição do progresso do time por ciclo de entrega (*Sprint*), realizando os ajustes necessários para obter maior desempenho no projeto.

3.7.CONDUZIR REVISÃO DE MARCOS

- **CMMI-DEV:** o intuito da prática é analisar o resultado do projeto através de marcos. As evidências necessárias devem assegurar a realização de revisão nos marcos do projeto, conforme o planejamento.
- **Scrum:** no framework há a cerimônia *Sprint Review Meeting*, que acontece a cada final de *Sprint*, e é utilizada para verificação do que foi feito e do que deveria melhorar de acordo com os critérios de aceito definidos pelo *Product Owner*, podendo ser adequada para atender o CMMI-DEV.
- **Plano de ação:** revisar os marcos vinculados a cada *release* em cada *Sprint Review Meeting* ou a cada mudança ocorrida.

3.8.ANALISAR PROBLEMAS

- **CMMI-DEV:** prática tem o objetivo de coletar, analisar e determinar medidas para correção desses problemas. Para essa prática, é esperado evidências que permitam o controle e rastreabilidade dos problemas ocorridos durante a execução do projeto e a ações corretivas tomadas pelos interessados.
- **Scrum:** os problemas – também chamados de impedimentos – são alinhados entre o time e o *Scrum Master* para serem resolvidos.

- **Plano de Ação:** o Scrum Master é o responsável por remover impedimentos e solucionar os problemas junto aos interessados e a equipe.

3.9. TOMAR AÇÕES CORRETIVAS

- **CMMI-DEV:** prática visa a utilização de ações corretivas ou preventiva para os problemas identificados no decorrer do projeto. A evidência gerada para essa prática assegura que ações corretivas foram tomadas pelos interessados.
- **Scrum:** os problemas são passados ao Scrum Master para que a tomada de ações seja feita, mas, não há documentos que registrem e evidenciem essas medidas tomadas.
- **Plano de Ação:** adaptar o modelo de trabalho utilizado pelo time sempre que for necessário. Ajustar os parâmetros negociados com o cliente, como escopo e prazo, quando os problemas sinalizarem o possível não cumprimento das tarefas assumidas pela equipe. A documentação das ações corretivas tomadas pelo *Scrum Master*.

3.10. GERENCIAR AÇÕES CORRETIVAS

- **CMMI-DEV:** prática visa a utilização de ações corretivas para os problemas identificados no decorrer do projeto. A evidência gerada para essa prática assegura quais ações foram executadas pelos envolvidos.
- **Scrum:** os problemas são passados ao *Scrum Master* para que a tomada de ações seja feita, mas, não há documentos que registrem e evidenciem essas medidas tomadas.
- **Plano de Ação:** as ações corretivas serão incluídas e documentadas nas *Daily Scrum Meetings*, assim que o problema for reportado pelos responsáveis por sua correção e a avaliação sobre a eficácia das medidas tomadas é revisada na *Sprint Retrospective Meeting*.

4. ANÁLISE E APLICAÇÃO

Nesta seção será apresentado os resultados da execução do plano de ação definido na seção anterior na área de Monitoramento e Controle de Projeto conforme o CMMI-DEV para cada prática específica utilizando a metodologia ágil Scrum, que foram implantados na **Empresa A** (nome ocultado por confidencialidade).

4.1.MONITORAR OS PARÂMETROS DE PLANEJAMENTO DO PROJETO

Conforme o estabelecido no plano de ação para esta prática, para que o Scrum a comporte foi utilizado alguns artefatos da própria metodologia associada a uma planilha para controle de custos.

Para o controle e acompanhamento de parâmetros como desempenho do projeto em relação ao prazo do cronograma, atributos de trabalho e tarefas foi usado o *Product Backlog*, onde estão todos os requisitos definidos no escopo do projeto aliado ao *Product Burndown*, que demonstra graficamente a evolução do desenvolvimento com relação a quantidade de *Sprints* planejadas (refletindo o estipulado no cronograma). No *Product Burndown* também é possível localizar os itens do *Backlog* que estão faltantes e os responsáveis pela sua conclusão.

Com relação ao monitoramento dos custos de projetos, pelo Scrum não possuir nenhum artefato de apoio a essa prática, foi utilizado uma planilha simples que dispunha a alocação de recursos, seus custos de cada recurso no projeto e prazos definidos para as entregas. Dessa forma, é possível verificar constantemente se há desvios no planejamento relacionado ao escopo, prazo, custos, cronograma ou super alocação dos recursos envolvidos.

O objetivo desta prática é garantir o rastreamento e identificação previa de possíveis desvios nos parâmetros que permeiam o projeto de forma que propicie a tomada de ações para que o prazo seja combinado seja cumprido. Com a complementação feita com a planilha de custos aos artefatos do Scrum a prática torna-se suportada.

4.2.MONITORAR OS COMPROMISSOS

O comprometimento da equipe com as obrigações do projeto é assumido na *Sprint Planning Meeting* e monitorados no decorrer da execução do projeto; para que satisfizesse a essa prática com a metodologia Scrum foi utilizado a cerimonia *Sprint Planning Meeting* somada a utilização do e-mail corporativo da equipe.

A cada *Sprint Planning Meeting* a equipe, juntamente com o *PO*, se comprometem a entregar os itens para a próxima *Sprint* e a evidência do comprometimento (ata da reunião) é enviada para os recursos envolvidos na *Sprint* e o *PO*, para que respondam com o aceite dos termos. No decorrer de uma *Sprint*, a equipe de desenvolvimento não poderá receber tarefas que não foram planejadas.

O acompanhamento efetivo dos compromissos da *Sprint* é feito por meio do *Sprint Burndown*, *Dailys* e *Sprint Retrospective Meeting*.

4.3.MONITORAR OS RISCOS DO PROJETO

No Scrum sem modificações a *Daily Scrum Meeting* tem o objetivo de identificar os impedimentos e riscos que estão ocorrendo no projeto por parte dos recursos envolvidos, porém, para que atendesse ao CMMI-DEV foi necessário a realização de adequação para que o documento de plano de riscos (que foi criado na reunião de planejamento de projeto), colocando ela na pauta a cada *Daily* planejamento para a tomada de ações proativas.

Outro artefato Scrum que foi adaptado foi a *Sprint Retrospective Meeting*; nela os compromissos internos ou externos ao projeto que foram documentados no decorrer da *Sprint* são revistos para que os não foram cumpridos façam parte das lições aprendidas que será revisto no início da *Sprint* subsequente para que condutas corretivas às futuras *Sprints* sejam tomadas preventivamente.

4.4.MONITORAR O GERENCIAMENTO DE DADOS

Como no Scrum não há procedimentos específicos que comportem a prática de acordo com o estipulado no CMMI-DEV, foi necessário realizar alguns ajustes.

A primeira ação tomada foi o levantamento e classificação (em confidências e não) dos dados indispensáveis ao projeto no documento de planejamento para que seja documentado e distribuído aos envolvidos e a fim de que seja revisto em nas cerimônias do Scrum. Outros pontos importantes a serem destacados é a necessidade de garantir que o desfecho desta revisão seja documento e distribuídos aos interessados.

4.5.MONITORAR O ENVOLVIMENTO DOS *STAKEHOLDERS*

O *Scrum* destaca o papel de *Scrum Master* como responsável pelo monitoramento dos *Stakeholders* promovendo as reuniões determinadas pelo *Scrum* e utilizando elas como meio de assegurar que os interessados compreendam as regras e práticas da metodologia. Também foi necessário a utilização do *Product Backlog* e *Sprint Backlog* como artefatos de registro do envolvimento dos *stakeholders* além de ferramenta de apoio as evidências de que esta prática foi devidamente atendida.

No decorrer das *Sprints*, o aconselhamento e a supervisão do envolvimento do *Product Owner* para que se mantivesse focado no objetivo do projeto e na eleição dos requisitos que agregassem mais valor a entrega, assim como foi apresentado na reunião de *KickOff* de projeto, também se fez necessário.

4.6.CONDUZIR REVISÃO DE PROGRESSO

Por meio das reuniões estabelecidas pelo *Scrum*, como as *Dailys* e *Sprint Retrospective*, é possível cumprir os objetivos esperados para esta prática, sem necessidade de adaptações, pois, as evidências esperadas para esta deverão comprovar a verificação do progresso do projeto por parte dos interessados.

O *report* do andamento das atividades e marcos do projeto para que os *stakeholders* e partes envolvidas acompanhassem foi feito através do *Backlog* e o do gráfico de *Burndown*. O progresso após a conclusão de *Sprints* foi analisado e comparado com o previsto no documento de revisão do projeto e resultado da análise é documentado de forma incremental.

4.7.CONDUZIR REVISÃO DE MARCOS

A revisão de marcos é uma prática amparada pelo *Scrum* sem que seja necessário adaptações. De acordo com o plano de ação previamente definido a ação a ser adotada nesta prática é a revisão dos marcos definidos no plano de projeto a cada *Review Meeting* (após a entrega da *Sprint*). Os problemas identificados nesta análise são registrados e ações corretivas podem ser tomadas para impedir que possíveis problemas previstos sejam evitados; não conformidades nas entregas também foram documentadas.

4.8. ANALISAR PROBLEMAS

Nesta prática os problemas devem ser identificados para a especificação de ações necessárias para corrigi-los. Para satisfazer esta prática foi utilizado das *Dailys* para que os problemas (impedimentos) fossem reportados e a *Task Board* para acompanhamento dos impedimentos criados, incluindo o responsável pela resolução, data com as alterações dos status e a descrição; a resolução dos problemas identificados fica a cargo do Scrum Master.

O acompanhamento e a análise dos pontos críticos resultarão na listagem e classificação por nível de criticidade para determinar quais serão abordados prioritariamente.

4.9. TOMAR AÇÕES CORRETIVAS

De acordo com o Scrum, as ações corretivas estão diretamente relacionadas aos problemas reportados ao *Scrum Master* por integrantes do time ou interessados nos projetos nas *Dailys*, gerando a lista de problemas a serem tratados.

Para que o CMMI-DEV seja contemplado é necessário que a lista de problemas e suas respectivas tratativas sejam priorizados de acordo com seu nível. Ao realizar o planejamento das ações tomadas é necessário que a análise do perfil a seja realizada junto ao *Product Owner* e/ou os *stakeholders*.

4.10. GERENCIAR AÇÕES CORRETIVAS

A gestão de ações corretivas segundo o CMMI-DEV requer evidências que tenham registros do monitoramento dos problemas outrora relatados e das ações tomadas para resolução dos problemas.

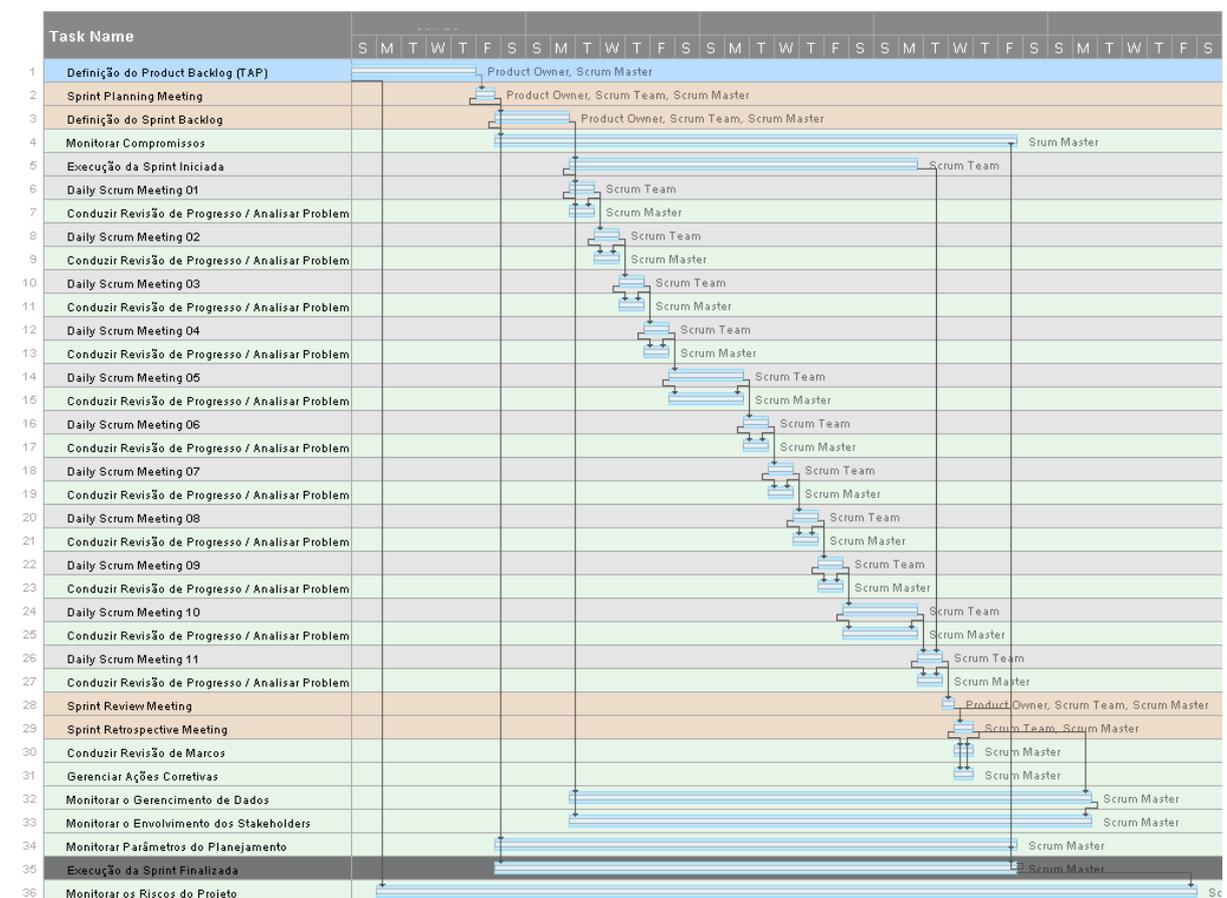
Na metodologia ágil foi possível ajustar as reuniões *Dailys* para que assuntos relacionados a problemas fossem tratados e os responsáveis pelas resoluções pudessem relatar e documentar as medidas tomadas; outro ajuste foi realizado na *Sprint Retrospective Meeting* para que as ações corretivas que foram implementadas fossem avaliadas, mensurando a eficácia delas, resultando em uma melhoria frequente do processo *Sprint a Sprint*.

Caso haja necessidade é recomendado que o plano de projeto seja alterado para que reflita a situação real do projeto com relação a riscos, problemas e ações corretivas para que sirva de base para que quando novos desvios surgirem já haja *know how* para que novas ações sejam tomadas.

5. CONCLUSÃO E RESULTADOS

O Scrum não comporta diretamente as práticas específicas de gerenciamento de projetos propostos pelo nível 2 do CMMI-DEV, mas, pode ser ajustado e adaptado de acordo com cada processo interno da empresa, através da complementação de práticas às cerimônias do Scrum, conforme o demonstrado na Figura 9, que contém a representação do cronograma de projeto para uma Sprint.

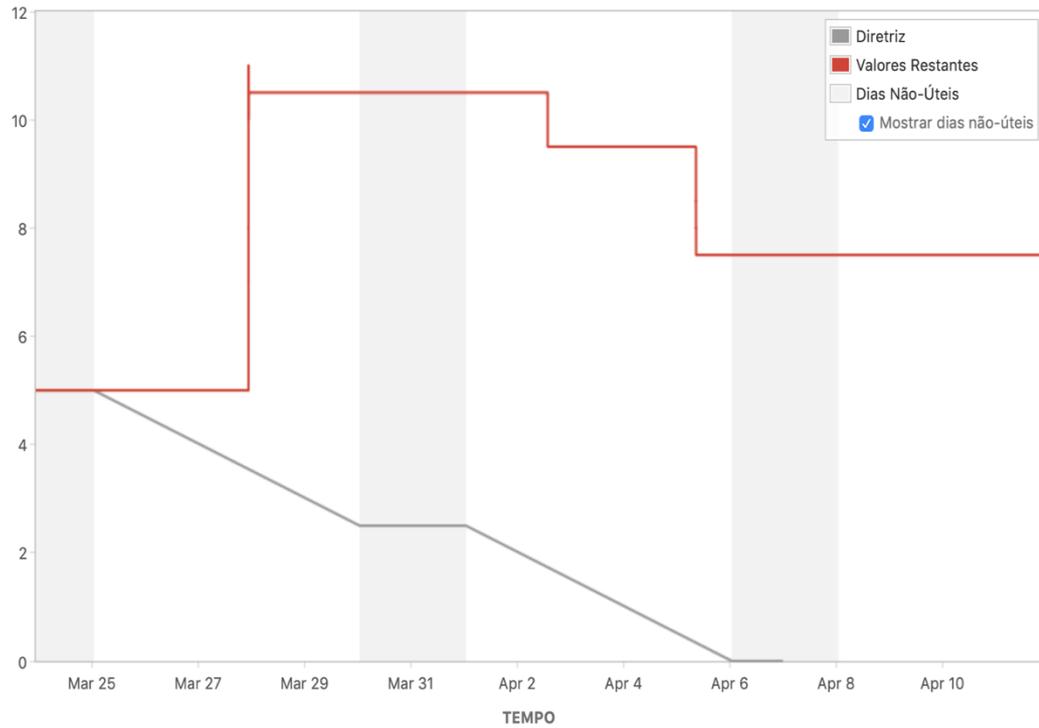
Figura 9 - Cronograma da Sprint 10 do projeto



Fonte: A autora.

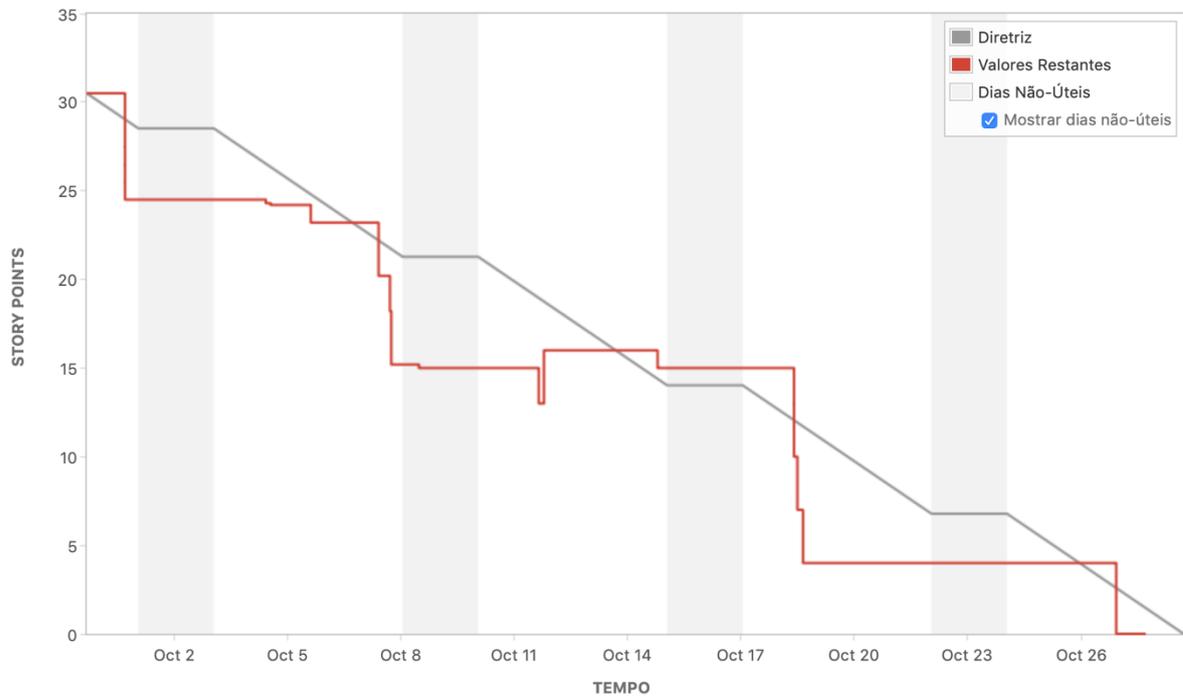
O estudo de caso apresentado demonstrou grandes taxas de sucesso para a área de Monitoramento e Controle e suas práticas específicas, com poucas adaptações na metodologia ágil Scrum, o que resulta em um bom início para as empresas que buscam maturidade, sem perder a agilidade que a metodologia proporciona. Sendo assim, mesmo empresas com times pequenos, e, muitas vezes, com poucos recursos financeiros podem elevar seu grau de maturidade no processo de desenvolvimento de softwares, conforme o demonstrado nas figuras 10 e 11.

Figura 10 - Gráfico de Burndown da Sprint antes das práticas do CMMI.



Fonte: Sprint screen da aplicação de gerenciamento da Sprint.

Figura 11 - Gráfico de Burndown da Sprint após práticas do CMMI.



Fonte: Sprint screen da aplicação de gerenciamento da Sprint.

Nas figuras acima é possível comparar o progresso de trabalho entre uma Sprint executada antes da implantação dos processos do CMMI-DEV e uma Sprint logo após essa implantação. Na figura 9, o término da Sprint foi comprometido e ela foi entregue após a data definida com o *Product Owner*, pois, a equipe de trabalho várias vezes precisou “apagar incêndios” que surgiam no decorrer no projeto e que não estavam mapeados ou não eram documentados, enquanto na figura 10, após a implementação das práticas de Monitoramento e Controle do CMMI-DEV, o trabalho foi realizado de forma contínua, respeitando aos prazos impostos.

5.1. TRABALHOS FUTUROS

Tendo este trabalho como base a sugestão é analisar continuamente os resultados da metodologia ágil Scrum na área de Monitoração e Controle, documentar a evolução e realizar um estudo aprofundado em outras áreas de processo do CMMI-DEV para abranger todas as áreas e elevar o nível de maturidade da empresa como um todo.

REFERÊNCIAS

- ABNT. ISO/IEC 12207 – Tecnologia Da Informação – **Processos De Ciclo De Vida De Software**. Rio De Janeiro: ABNT, 1996.
- AGILE ALLIANCE. **Manifesto for agile software development**. Disponível em <<http://www.agilemanifesto.org/>>. Acessado em 15/01/2019.
- AHERN, Dennis M.; CLOUSE, Aaron; TURNER, Richard. **CMMI Distilled: A Practical Introduction to Integrated Process Improvement**. – 3rd edition. Addison Wesley, 2008.
- AMARAL, Luis Manuel Gonzalez, and João Pascoal Faria. "A **gap analysis methodology for the team software process**." Quality of Information and Communications Technology (QUATIC), 2010 Seventh International Conference on the. IEEE, 2010.
- ANACLETO, Alessandra, and C. Wangenheim. **Método e modelo de avaliação para melhoria de processos de software em micro e pequenas empresas**. Diss. Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós- Graduação em Ciência de Computação., 2004.
- ASATO, Regina Yoneko; Spinola, Mauro Mesquita; Silva, Water Henrique de Farias. **Iniciando a implementação do modelo CMMI em uma fábrica de Software: Um processo para a elaboração do diagnóstico e plano de ação, XIII SIMPEP - Bauru, SP, Brasil, 6 a 8 de novembro de 2006**.
- BAETJER, Jr. H., “**Software as Capital**”, IEEE Computer Society Press, 1998, p. 85.
- BARTIÉ, Alexandre. **Garantia da Qualidade de Software: adquirindo maturidade organizacional**. Rio de Janeiro, Elsevier, 2002.
- BOEHM, Barry W. A spiral model of software development and enhancement. **Computer**, n. 5, p. 61-72, 1988.
- BOEHM, B.; TURNER, R. Management challenges to implementing agile process in traditional development organizations. **IEEE Software**, 2005.
- Building CMMI Process Performance Models Performance Models**, disponível em <http://www.dtic.mil/dtic/tr/fulltext/u2/a558352.pdf>. Capability Maturity Model® Integration (CMMI®) Overview, fornecido pelo Software Engineering Institute. Acesso em: 15 fev. 2019.
- CMMI® for Development, Version 1.2 CMMI-DEV**, Agosto 2006, fornecido pelo Software Engineering Process Management Program, disponível em <http://www.sei.cmu.edu/>. Acesso em: 14 fev. 2019.
- CRISSIS, M. B.; KONRAD, M.; SCHRUM, S. **CMMI: guidelines for process integration and product improvement**. [S.l.]: Addison-Wesley Professional, 2003.
- FIORINI, SOELI T., ET AL. **Engenharia De Software Com Cmm**, Rio De Janeiro, ED. BRASPORT, 1998.

FOWLER, M, “**The New Methodology**”, Junho 2002, disponível em <http://www.martinfowler.com/articles/newMethodology.html#N&B>. acesso em: 13 fev. 2019.

HANNA, M., “**Farewell to Waterfalls**”, Software Magazine, Maio 1995, pp 38-46.

HANSSON, C.; DITTRICH, Y; GUSTAFSSON B.; ZARNAK, S. How agile are industrial software development practices? **Journal of Systems and Software**, vol. 79, n. 9, p. 1295-1311, 2006.

HSUEH, N.-L. et al. Applying UML and software simulation for process definition, verification, and validation. **Information and Software Technology**, v. 50, n. 9-10, p. 897-911, 2008.

JACOBSON, I. “**A Resounding ‘Yes’ to Agile Process - But Also More**”, Cutter IT Journal, volume 15, numero 1, Janeiro 2002, pp 18-24.

KERZNER, H. **Gestão de Projetos: as melhores práticas**. Porto Alegre: Bookman, 2006.

LAZZARI, M. L. **Implantação de Melhoria de Processos de Desenvolvimento de Software Utilizando o CMMI: Um Estudo de Caso em uma Empresa de Varejo**. 2006. 70p. Monografia (Sistemas de Informação) – Ciências Exatas e Tecnológicas, Universidade do Vale do Rio dos Sinos, São Leopoldo, 2006

LESSA, Rafael. e Lessa Edson., “**Modelos de Processos de Engenharia de Software**”, agosto 2012.

MARCAL, A. S. C. **SCRUMMI: Um processo de gestão ágil baseado no SCRUM e aderente ao CMMI**. Dissertação (Mestrado) — Universidade de Fortaleza, 2009.

MCMAHON, 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.

McDermid, J. e Rook P., “**Software Development Process Models**”, Software Engineer’s Reference Book, CRC Press, 1993, pp 15/26 - 15/28.

Monteiro, Paula A. F. “**Tailoring CMMI-DEV and RUP Frameworks for ML2/3-Compliance Analysis**”, Abril 2014.

PIKKARAINEN, M., e Mäntyniemi, A., “An Approach for Using CMMI in Agile Software Development Assessments: Experiences from Three Case Studies”, Maio 2006.

PMI, Resolução. SEDETUR n. 001, de 31 de maio de 2012.

PRESSMAN, ROGER S. "**Processos De Software**." Engenharia De Software. 7aPorto Alegre (RS): AMGH, 2010.

PRESSMAN, ROGER S., **Engenharia De Software**, São Paulo, ED. MAKRON BOOKS, 1995.

PRESSMAN, ROGER S., **Engenharia De Software**, Rio De Janeiro, ED. MCGRAW-HILL, 2002.

PRIKLADNICKI, Rafael, Carlos Alberto Becker, and Marcelo Hideki Yamaguti. "**Uma Abordagem para a Realização de Diagnóstico Inicial em Empresas que Implementam o MPS. BR.**" I Workshop de Implementadores (W2-MPS. BR), Brasília. 2005.

PRIKLADNICKI, Rafael; WILLI, Renato; MILANI, Fabiano. **Métodos ágeis para desenvolvimento de software**. 1. ed. Porto Alegre: Bookman, 2014.

PROJECT MANAGEMENT BODY OF KNOWLEDGE – PMBOK. **Um guia do conjunto de conhecimentos em gerenciamento de projetos**. 4 ed. Pensilvânia: Project Management Institute, 2008.

RABECHINI JUNIOR, R. **Competências e maturidade em gestão de projetos: uma perspectiva estruturada**. 2003. 274 p. Tese (Doutorado em Engenharia de Produção) - Universidade de São Paulo – USP, São Paulo.

SCHWABER, K, e Beedle, M., “**Agile Software Development with SCRUM**”, Prentice Hall, 2001.

SBROCCO, José Henrique Teixeira de Carvalho; MACEDO, Paulo Cesar de. **Metodologias ágeis: engenharia de software sob medida**. 1. ed. São Paulo: Saraiva, 2012.

The Agile Alliance, Página principal disponível em <http://www.agilealliance.org/home>. Acesso em: 21 dez. 2018.

TONINI, Antonio Carlos; DE CARVALHO, Marly Monteiro; DE MESQUITA SPINOLA, Mauro. **Contribuição dos modelos de qualidade e maturidade na melhoria dos processos de software**. Production, v. 18, n. 2, p. 275-286, 2008.

WANGENHEIM, C. G. V, C.; DA SILVA, D. A.; BUGLIONE, L.; SCHEIDT, R.; PRIKLADNICKI, R. **Best Practice Fusion of CMMI- DEV v1.2 (PP, PMC, SAM) and PMBOK 2008**. Information and Software Technology, Elsevier, 2010.

SANTOS, Rodrigo Pinheiro dos et al. **Construção e validação de um instrumento para a análise da qualidade de serviço de empresas avaliadas no CMM®/CMMI®**. 2007.

SUTHERLAND, J.; JAKOBSEN, C. R.; JOHNSON, K. **Scrum and CMMI Level 5: The Magic Potion for Code Warriors**. In: AGILE 2007 (AGILE 2007), Washington, DC, 2007, p. 272-278. Disponível em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4293608&isnumber=4293563>.

DRAW.IO DIAGRAMS. Version: 10.5.0 <https://draw.io>. Acesso em: 12 mar. 2019.