



**Universidade Federal Rural de Pernambuco**  
**Departamento de Estatística e Informática**



**IMPLANTANDO CMMI EM EMPRESAS DE MANUTENÇÃO E  
DESENVOLVIMENTO DE PRODUTOS DE SOFTWARE :  
UM ESTUDO DE CASO**

**Matheus Benicio Rodrigues Evangelista**

**Recife  
2014**

**Matheus Benicio Rodrigues Evangelista**

**IMPLANTANDO CMMI EM EMPRESAS DE MANUTENÇÃO E  
DESENVOLVIMENTO DE PRODUTOS DE SOFTWARE :  
UM ESTUDO DE CASO**

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Profa. Dra. Ana Cristina Rouiller

**Recife  
2014**

*Dedico esse trabalho aos meus pais que sempre me apoiaram e acreditaram em mim e a toda a equipe da SWQuality .*

## AGRADECIMENTOS

*À Deus, por ser a razão da minha vida.*

*Aos meus pais, meus maiores modelos de força e coragem, as minhas irmãs, Ana Luiza, Ana Carolina e à toda a minha família. Obrigado pelo amor e incentivo que sempre me deram.*

*Ao minha orientadora, professora, mentora, amiga Ana Rouiller, pela orientação, pelo incentivo, ensinamentos, dedicação e apoio, o que fez com que eu conseguisse chegar onde estou.*

*A Aninha, meu amor, pelo carinho, amor, atenção, paciência e pelo seu inestimável apoio, estando sempre ao meu lado.*

*Aos meus amigos que estiveram comigo em toda essa jornada do curso! Em especial a aqueles que estiveram comigo não só na faculdade, mas em todas as saídas de fim de semana, madrugadas de dota e desenvolvimento de projetos: Bruno, Rodrigo e Victor.*

*Aos meus amigos e companheiros da SWQuality que foram essenciais na minha caminhada: Ana, Heron, Guilherme, Sérgio, Sandro, Mauricio, Carlos, Renata, Mery, Victor, Mariana, Eça e Luiz. Em especial a Mauricio pelo apoio imensurável, que fez com que eu conseguisse finalizar o meu trabalho.*

## RESUMO

Esta dissertação descreve um estudo de caso sobre a implantação do nível 2 de maturidade do modelo "*Capability Maturity Model – Integration*" (CMMI) – *Development* utilizando métodos ágeis, realizado no âmbito dos trabalhos da empresa SWQuality Consultoria e Sistemas. Inicialmente esta dissertação apresenta uma revisão conceitual sobre a área de melhoria de processo e qualidade de software, bem como uma introdução ao modelo CMMI. Apresenta uma visão geral sobre métodos ágeis e do framework *Scrum*. A seguir, relata como ocorreu a implantação do modelo e os resultados obtidos pela empresa decorrente desta implantação, apresentando o processo de desenvolvimento definido e os documentos e controles criados para a utilização deste processo. Além de serem mostradas as estratégias de implementação, o antes e após CMMI e todas as dificuldades inerentes à execução desta implementação, este trabalho tem como principal objetivo procurar apurar se todos os resultados obtidos após a implementação satisfazem as exigências do modelo de referência e mapear os benefícios obtidos pela organização ao final do programa de melhoria.

## **ABSTRACT**

This dissertation describes a case study on the deployment of Level 2 maturity model “Capability Maturity Model - Integration “(CMMI ) - . Development using Agile methods , performed in the work of the company SWQuality Consulting and Systems Initially, this dissertation presents a conceptual review to the area of process improvement and software quality , as well as an introduction to the CMMI model . presents an overview of agile methods and Scrum framework . Herein, we report how the implementation of the model and the results obtained by the company occurred arising from this deployment, showing the process of development and set the documents and controls designed for the use of this process . Besides being shown deployment strategies , before and after the CMMI and all the difficulties inherent in the implementation of this implementation , this work is main goal seek to ascertain whether all the results obtained after the implementation meet the requirements of the reference model and map the benefits obtained by the organization at the end of the improvement program .

## LISTA DE FIGURAS

Figura 1 - Visão do modelo IDEAL.....	13
Figura 2 - Processo e seus Componentes .....	15
Figura 3 - CMMI: Áreas de Processo em duas Representações: por Estágio e Contínua.....	17
Figura 4 - Scrum: Visão do Processo.....	27
Figura 5 - Gráfico Burndown .....	30
Figura 6 - Quadro Scrum.....	31
Figura 7 - Resultado do Gap Analysis.....	36
Figura 8 - Backlog .....	37
Figura 9 - Quadro de Atividades.....	38
Figura 10 - Ata de Planejamento de Sprint .....	40
Figura 11 - Checklist de Qualidade .....	41
Figura 12 - Registro de Não Conformidade.....	42
Figura 13 - Necessidades de medição e indicadores relacionados.....	42
Figura 14 - Indicadores (parte 1) .....	43
Figura 15 - Indicadores (parte 2) .....	43
Figura 16 - Ciclo de Vida do Projeto.....	44
Figura 17 - Plano de Projeto.....	45
Figura 18 - Acompanhamento do Projeto.....	46
Figura 19 - Documentação do Projeto no Redmine .....	47
Figura 20 - Gráfico do Histórico de Revisões do SVN.....	48
Figura 21 - Baseline de Produto.....	49
Figura 22 - Baseline de Projeto .....	50
Figura 23 - Documentos Organizacionais mantidos no Redmine.....	51
Figura 24 - Resultado do segundo Gap Analysis Final .....	52

## LISTA DE QUADROS

Quadro 1 - Grau de atendimento das evidências geradas às práticas específicas da PA Project Planning .....	53
Quadro 2 - Grau de atendimento das evidências geradas às práticas específicas da PA Project Monitoring and Control .....	54
Quadro 3 - Grau de atendimento das evidências geradas às práticas específicas da PA Requirement Management .....	55
Quadro 4 - Grau de atendimento das evidências geradas às práticas específicas da PA Measurement and Analysis .....	56
Quadro 5 - Grau de atendimento das evidências geradas às práticas específicas da PA Configuration Management .....	57
Quadro 6 - Grau de atendimento das evidências geradas às práticas específicas da PA Product and Process Quality Assurance .....	58



## SUMÁRIO

1 INTRODUÇÃO .....	9
1.1 Contextualização e Motivação .....	9
1.2 Objetivos e Justificativas .....	11
1.3 Metodologia .....	12
1.4 Organização do Trabalho.....	13
2 INTRODUÇÃO A PROCESSOS E AO MODELO CMMI.....	14
2.1 Processos .....	14
2.2 Modelo CMMI.....	15
2.2.1 Arquitetura do Modelo.....	16
2.2.2 Disciplinas do Modelo .....	20
2.2.3 Elementos do Modelo .....	23
3 FRAMEWORK SCRUM.....	25
4 RELATO DE EXPERIÊNCIA DA IMPLANTAÇÃO DO NÍVEL 2 DO CMMI-DEV EM UMA EMPRESA DE MANUTENÇÃO .....	32
4.1 Objetivos e abordagem de melhoria .....	32
4.2 Caracterização da Empresa.....	33
4.3 Descrição do caso.....	34
4.3.1 Diagnóstico e planejamento.....	34
4.3.2 Fase 1 .....	36
4.3.3 Fase 2.....	44
4.3.4 Fase 3.....	47
4.3.5 Fase 4.....	50
4.3.6 Fase 5.....	51
4.4 Mapeamento dos resultados obtidos com o CMMI-DEV.....	52
4.5 Benefícios observados para a empresa .....	59
5 CONCLUSÃO.....	61
REFERÊNCIAS.....	62

## 1 INTRODUÇÃO

Neste capítulo introdutório é apresentado o contexto, no qual o trabalho está inserido, a motivação, os objetivos, a metodologia utilizada e a estrutura em que o mesmo foi descrito

### 1.1 Contextualização e Motivação

Grande parte da população mundial depende de aplicações de *software* para realizar suas atividades diárias [Rocha 2011]. Se alguns sistemas de uso global deixarem de funcionar, aproximadamente 40% da população mundial sofrerá as consequências deste problema [Reed 2010]. Desta forma, a área de *software* está se tornando cada vez mais significativa na economia mundial. No entanto, o Brasil está em sétimo lugar nesse mercado, tendo movimentado cerca de 27 bilhões de dólares, dedicados ao desenvolvimento, produção e distribuição de *software* e prestação de serviços com foco em *software* [ABES 2012]. Isso evidencia uma oportunidade de crescimento, tanto do mercado interno quanto externo, vinculada à produção de *software*. Contudo, a inserção do Brasil no mercado externo depende de muitos fatores, dentre eles o reconhecimento por todos os players da qualidade dos produtos de *software* brasileiros. A qualidade de produtos de *software*, entretanto, está fortemente relacionada à qualidade do processo de *software* [Fuggetta 2000].

Durante esta década, ainda que tenham ajustado seus processos para a produção de *software* de qualidade dentro dos prazos e orçamentos confiáveis, as organizações são pressionadas por seus concorrentes a reduzir substancialmente os prazos para a entrega dos produtos. Organizações que são capazes de integrar, harmonizar e acelerar seus processos de desenvolvimento e manutenção de *software* tem primazia no mercado [Curtis 2000]. É nesse cenário de aquecimento que as metodologias de gestão mais avançada, visando à obtenção de qualidade e agilidade nos projetos, emergem como um importante diferencial competitivo entre os players, balizando cada vez mais as atividades do setor.

Atingir padrões internacionais de qualidade e produtividade no setor de *software*

no Brasil é condição essencial para a busca da competitividade mundial das indústrias [MCT/SEITEC 2007]. Com esta motivação, inclusive, foram desenvolvidos alguns programas como o Programa Brasileiro da Qualidade e Produtividade de Software – PBQP-Software. Este Programa procura estimular a adoção de normas, métodos, técnicas e ferramentas da qualidade e da engenharia de software, promovendo a melhoria da qualidade dos processos, produtos e serviços de software brasileiros, de modo a tornar as empresas mais capacitadas a competir no mercado globalizado.

Desde 1993, entidades governamentais e não governamentais vêm investindo coerentemente na capacitação das empresas para a melhoria da qualidade e produtividade em software. Em particular, obteve-se excelente resultado com o projeto denominado “Rumo à ISO 9000”, referenciado internacionalmente na revista inglesa *Quality Word* do IQA (*Institute of Quality Assurance*) em novembro de 1995, [Weber, 1995]. A mesma metodologia utilizada no projeto “Rumo à ISO 9000” (consórcio de empresas) vem sendo empregada na condução de outros projetos para qualificar empresas de software em CMM, CMMI, ISO/IEC 15504 e, mais recentemente, no projeto MPS.BR, com o modelo MR-MPS.BR.

O Programa MPS.BR é um exemplo claro do direcionamento do governo nesta área. O projeto foi criado em dezembro de 2003 sob a coordenação da Associação para Promoção da Excelência do Software Brasileiro (SOFTEX), que recebeu incentivos consideráveis do Banco Interamericano de Desenvolvimento (BID) – US\$ 1,35 milhão- e do Ministério da Ciência e Tecnologia – através do Consórcio de Municípios/Ciência, Tecnologia e Inovação e possibilitou a melhoria efetiva de processos de desenvolvimento de software em diversas empresas brasileiras.

Apesar dos esforços realizados até o momento, dados publicados pelo SEI-*Software Engineering Institute* demonstram o quanto a indústria de software nacional precisa melhorar para atingir níveis elevados de maturidade. Segundo o SEI, em 2010, apenas 144 empresas no Brasil possuíam certificação CMMI (*Capability Maturity Model Integration*), sendo a grande maioria no nível 2 de maturidade. Traçando um comparativo, na Índia, por exemplo, só no ano de 2010, mais de 400 empresas foram avaliadas por este modelo.

Os números acima, ainda que demonstrem que a preocupação com qualidade e adoção de padrões internacionais de qualidade e produtividade apresente-se como uma realidade da indústria nacional de software, transparece o fato de que, esta é

uma realidade vivenciada por poucas empresas, uma vez que o total de empresas de software existentes no Brasil é aproximadamente 8.500 [ABES, 2009].

Alguns fatores que levam a esta realidade são:

- Altos custos associados à adoção de modelos padrões da qualidade (CMMI e MPS.BR) diante de recursos humanos e recursos financeiros limitados;
- Adoção de planos, especificações e documentações não adequados à realidade das empresas para simplificar o processo de certificação e a conformidade com os modelos;
- Carência de profissionais que dominem o conhecimento e possuam a experiência prática que permita acelerar o processo de qualificação das empresas, em prazos adequados e a preços condizentes com sua capacidade financeira, e especialmente produzindo melhoria nas empresas;
- Falta e/ou pouca cultura em processos, falta de definição das metas organizacionais e imaturidade metodológica.

Neste cenário, esse projeto tem como objetivo de avaliar um método de implantação do nível 2 do CMMI , com o intuito de validar a metodologia. Com essa validação, este trabalho pode apoiar em trabalhos futuros para consolidação do método que servirá como ferramenta de apoio para empresas de pequeno e médio porte que são as principais empresas impactadas pelos fatores listados acima.

## **1.2 Objetivos e Justificativas**

Este trabalho tem como objetivo de avaliar um método de implantação do nível 2 do CMMI , com o intuito de validar a metodologia utilizada pela SWQuality Consultoria e Sistemas para implantação do nível 2 do CMMI em empresas de pequeno e médio porte. A SWQuality é uma empresa brasileira que iniciou suas atividades no ano de 2003 oferecendo serviços de consultoria, avaliação e capacitação em Gestão de Projetos, Preparação para Certificação em processos de Qualidade em Serviços, Tecnologia da Informação e desenvolvimento de software. Com essa validação, este trabalho pode apoiar em trabalhos futuros para consolidação do método, que servirá como ferramenta de apoio para empresas de pequeno e médio porte.

### 1.3 Metodologia

Este trabalho foi desenvolvido utilizando o CMMI como modelo de qualidade de processo e o IDEAL [IDEAL 2014] como modelo de melhoria de processo. Os passos para criação do processo foram baseados nos 5 estágios(Figura 1) do IDEALSM:

- Iniciação (*Iniating*): foram feitas reuniões para aprovação do trabalho na empresa do estudo de caso e definição do modelo de processo a ser utilizado, em seguida foram definidos os recursos necessários;
- Diagnóstico (*Diagnosing*): foi feito um diagnóstico da empresa no modelo CMMI, identificando o nível de aderência;
- Estabelecimento (*Establishing*): foi elaborado um plano de projeto, onde foram identificadas as atividades necessárias para criação do processo, a sua duração e a estratégia de desenvolvimento;
- Ação (*Action*): foi executado e acompanhado o planejamento feito para o trabalho;
- Aprendizagem (*Learning*): a cada iteração eram analisados os pontos negativos e positivos dos trabalhos realizados;

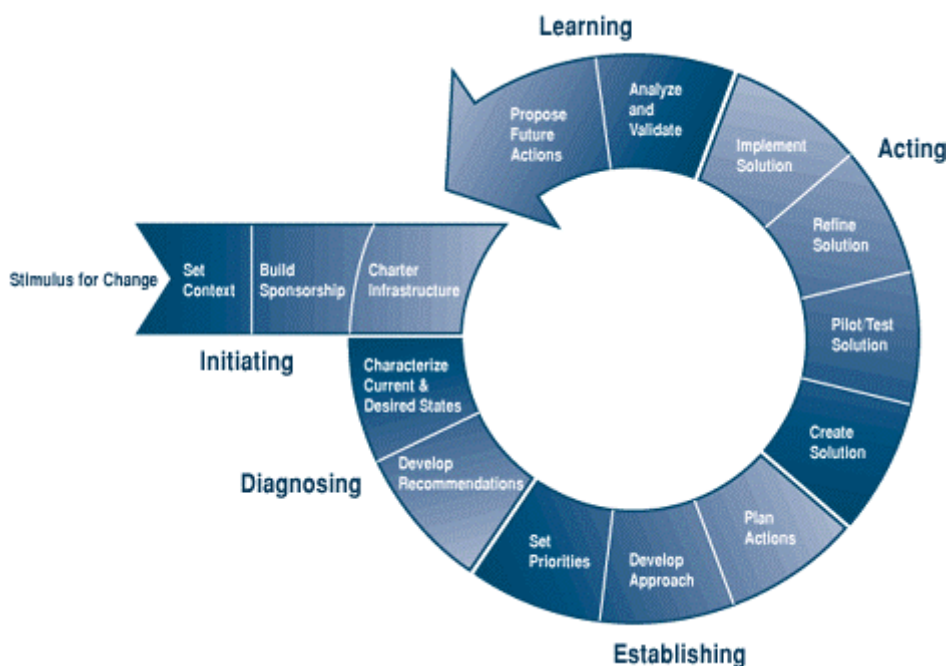


Figura 1 - Visão do modelo IDEAL

#### 1.4 Organização do Trabalho

Este trabalho encontra-se dividido em cinco capítulos: Além deste capítulo de Introdução, o Capítulo 2 refere-se ao Modelo CMMI, sendo abordada sua origem, objetivos e características. O Capítulo 3 diz respeito ao framework SCRUM, que foi utilizado como base para a implantação do programa de melhoria. No Capítulo 4 é apresentado o estudo de caso, o processo de definição e implementação do modelo de maturidade CMMI, as estratégias e ações e ações tomadas, bem como os estudos e observações feitas com o objetivo de verificar se o resultado dessas ações e estratégias atenderam as exigências do modelo de referência. No capítulo 5 são feitas diversas considerações finais.

## 2 INTRODUÇÃO A PROCESSOS E AO MODELO CMMI

Neste capítulo apresenta-se uma visão geral de Processos e do Modelo CMMI.

### 2.1 Processos

O conceito de processo é quase que intuitivo. As engenharias comumente descrevem processos como sendo diversas operações pelas quais passa um produto até ele ficar pronto[SOUZA 2004].

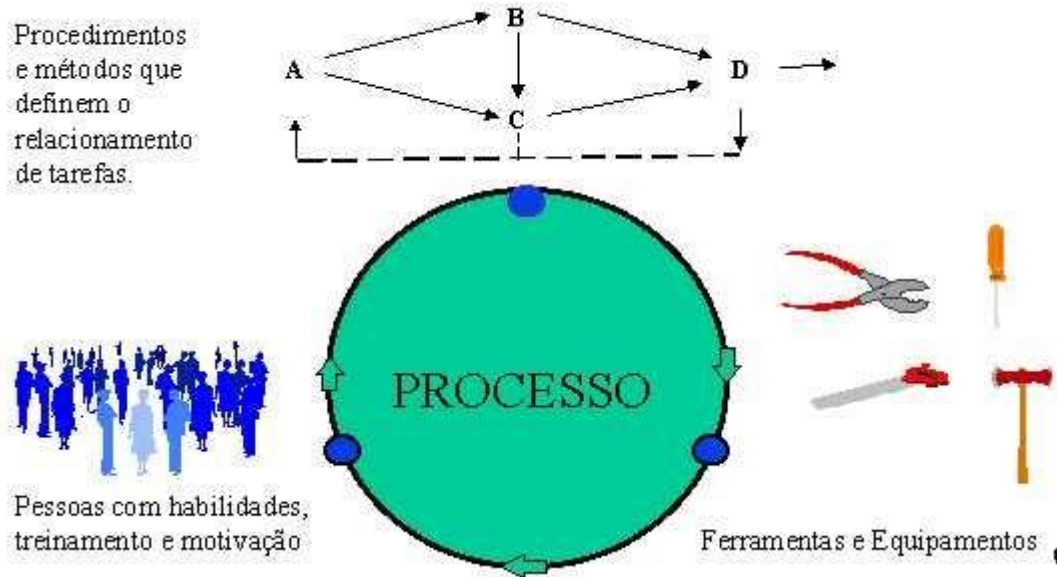
Algumas definições de processo seguem abaixo:

- A NBR define processo como um conjunto de atividades inter-relacionadas, que transforma entradas em saídas [ABNT 1994];
- O IEEE (*Institute of Electrical and Electronic Engineers*) define Processo como uma sequência de passos realizados para um determinado propósito [IEEE 1990].

Esta definição pode ser aplicada a qualquer atividade, seja ela da manufatura ou não.

Paulk [PAULK 1995] define Processo de Software como um conjunto de atividades, métodos, práticas e tecnologias que as pessoas utilizam para desenvolver e manter software e seus produtos relacionados. A Figura 2 ilustra esta definição.

## Definição de Processo de Software



**Figura 2 - Processo e seus Componentes**

Fonte: Autoria própria.

Ainda segundo Paulk, considerando que o software é resultado do processo de desenvolvimento, espera-se que a sua qualidade seja altamente influenciada pela qualidade do processo que o gera.

Focando-se somente no produto deixam-se de lado assuntos relacionados como a escalabilidade, ou seja, o aumento do tamanho da equipe possui o risco de se perder qualidade. Além disso, não há a preocupação de como realizar melhor as tarefas.

Focando-se no processo, prevê-se a repetição de resultados, tendências futuras para os projetos, mantendo as características do produto. Um processo bem controlado evita surpresas e minimiza os riscos.

## 2.2 Modelo CMMI

Grande parte do conteúdo teórico desta seção foi retirada de [AGUIAR, 2004].

A sigla CMMI representa as iniciais de *Capability Maturity Model Integration* e nomeia tanto um projeto, quanto os modelos resultantes deste projeto. O Projeto

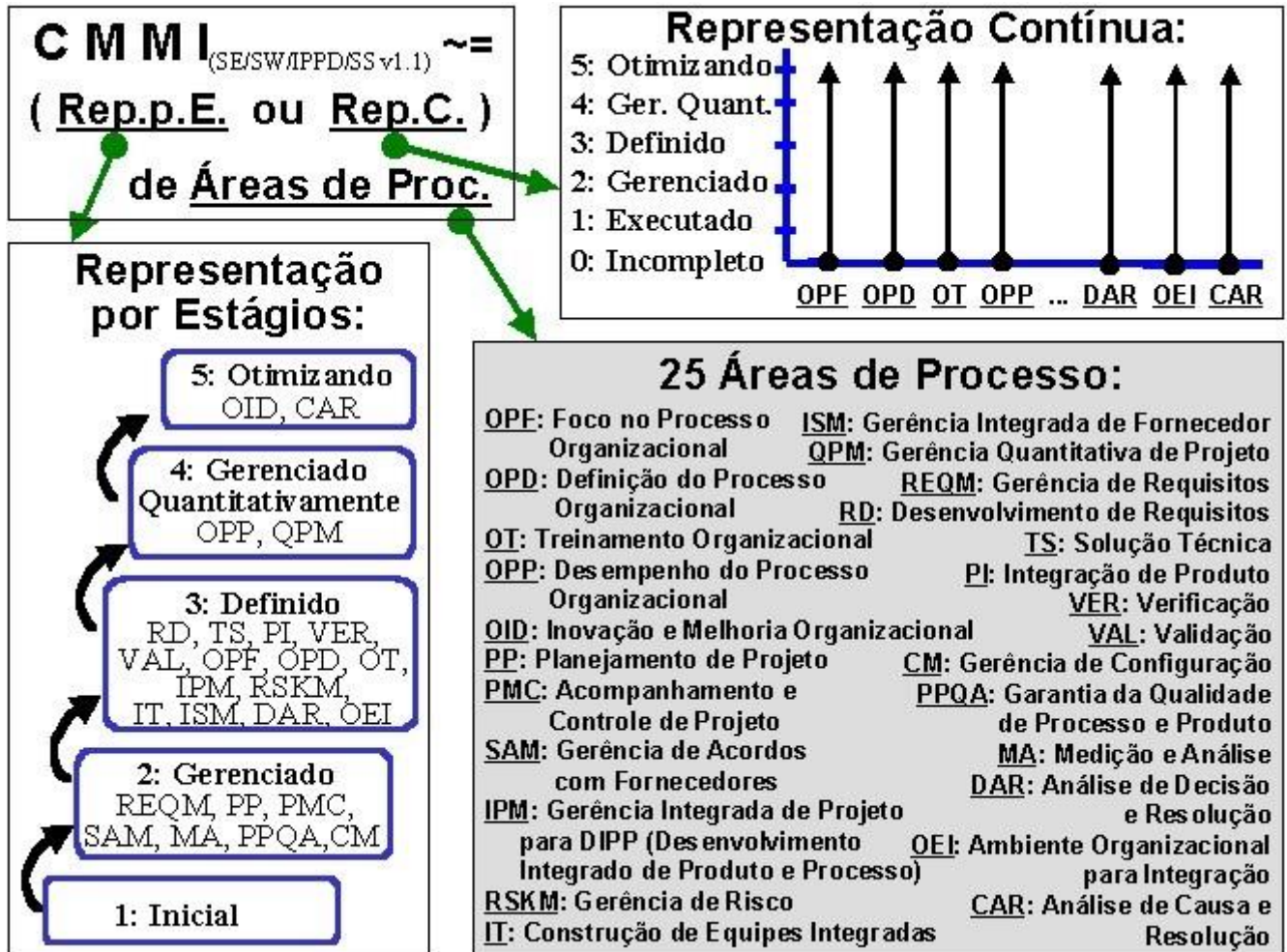


CMMI, que pode ser traduzido como Projeto de Integração dos Modelos de Maturidade da Capacidade, está sendo executado pelo *CMMI Institute* em cooperação com a indústria e governo, para consolidar um framework para modelos, evoluir e integrar modelos desenvolvidos pelo SEI (inicialmente os modelos SW-CMM, SE-CMM e IPD-CMM), e gerar seus produtos associados, incluindo material de treinamento e método de avaliação. Estes três modelos que foram evoluídos e integrados inicialmente foram a versão 2.0 do SW-CMM (*Capability Maturity Model for Software*), o SE-CMM: EIA 731 (*System Engineering Capability Maturity Model*) e o IPD-CMM (*Integrated Product Development Capability Maturity Model*).

Esta integração e evolução tiveram como objetivo principal a redução do custo da implementação de melhoria de processo multidisciplinar baseada em modelos. Multidisciplinar porque além da engenharia de software, o CMMI cobre também a engenharia de sistemas, aquisição, e a cadeia de desenvolvimento de produto. Esta redução de custo é obtida por meio da eliminação de inconsistências; redução de duplicações, melhoria da clareza e entendimento, utilização de terminologia comum, utilização de estilo consistente, estabelecimento de regras de construção uniforme, manutenção de componentes comuns, garantia da consistência com a Norma ISO/IEC 15504.

### **2.2.1 Arquitetura do Modelo**

A arquitetura do CMMI é composta basicamente pela definição de um conjunto de áreas de processo, organizadas em duas representações diferentes: uma como um modelo por estágio, semelhante ao SW-CMM, e outra como um modelo contínuo (semelhante à ISO/IEC 15504). A versão atual é composta por 25 áreas de processo, conforme ilustrado na Figura 3.



**Figura 3 - CMMI: Áreas de Processo em duas Representações: por Estágio e Contínua**  
 Fonte: Autoria própria.

Cada área de processo é definida no modelo por meio da descrição de seu propósito, notas introdutórias, relacionamentos com outras áreas, metas específicas, metas genéricas, práticas e subpráticas específicas, práticas e subpráticas genéricas, produtos de trabalho típicos e referências para outros elementos do modelo relacionados.

Na representação por estágio, as 25 áreas de processo estão agrupadas em 4 níveis de maturidade: níveis 2, 3, 4 e 5. O nível 1 não contém nenhuma área de processo, e a única exigência para que a empresa de software seja qualificada neste nível é a sua própria existência. Em relação a esta representação, o processo, de desenvolvimento e manutenção, de software ou sistema de uma unidade organizacional, pode estar classificado em um dos seguintes cinco níveis de maturidade:

- nível 1: Inicial (*Initial*)

- nível 2: Gerenciado (*Managed*)
- nível 3: Definido (*Defined*)
- nível 4: Gerenciado Quantitativamente (*Quantitatively Managed*)
- nível 5: Otimizando (*Optimizing*)

Cada nível de maturidade é definido basicamente pelo conjunto de áreas de processo do nível.

Na representação contínua, as mesmas 25 áreas de processo estão agrupadas em quatro grupos (gerência de processos, gerência de projetos, engenharia e suporte) e são definidos seis níveis de capacidade de processo. Nesta representação, o conjunto de atividades correspondente a cada uma destas áreas de processo, pode ter sua capacidade de execução classificada em um dos seguintes seis níveis de capacidade de processo:

- nível 0: Incompleto (*Incomplete*)
- nível 1: Executado (*Performed*)
- nível 2: Gerenciado (*Managed*)
- nível 3: Definido (*Defined*)
- nível 4: Gerenciado Quantitativamente (*Quantitatively Managed*)
- nível 5: Otimizando (*Optimizing*)

Cada nível de capacidade é definido por um conjunto de características que o processo deve satisfazer para estar naquele nível.

De forma semelhante ao que ocorre em relação ao modelo SW-CMM e à futura Norma ISO/IEC 15504, existem várias interseções, relacionamentos e contribuições do CMMI com a disciplina de gerência de projetos. Entre elas podemos destacar como talvez a mais relevante, o grupo de áreas de processo de gerência de projeto. Este grupo está descrito na seção seguinte.

Na representação contínua da versão atual do CMMI, oito áreas de processo contêm atividades relacionadas ao planejamento, acompanhamento e controle de projetos. Por isto estas oito áreas de processo compõem o grupo de gerência de projeto:

- Planejamento de Projeto (Project Planning – PP)
- Acompanhamento e Controle de Projeto (*Project Monitoring and Control – PMC*)
- Gerenciamento de Acordo com Fornecedor (*Supplier Agreement Management – SAM*)

- Gerenciamento Integrado de Projeto para IPPD (Desenvolvimento Integrado de Produto e Processo) (*Integrated Project Management for IPPD (Integrated Development of Product and Process – IPM for IPPD)*)
  - Gerenciamento de Riscos (*Risk Management – RSKM*)
  - Construção Integrada de Equipes (*Integrated Teaming – IT*)
  - Gerenciamento Integrado de Fornecedor (*Integrated Supplier Management – ISM*)
  - Gerenciamento Quantitativo de Projeto (*Quantitative Project Management – QPM*)

As áreas de processo PP, PMC e SAM representam, na visão do CMMI, uma gerência básica de projetos, compatível com organizações no nível 2 de maturidade, segundo a representação por estágio. Estas três áreas de processo definem referências para o estabelecimento e manutenção de planos de projeto e comprometerimentos, acompanhamento do desempenho real do projeto, gerenciamento e execução das ações corretivas, e gerenciamento dos acordos com os fornecedores.

O planejamento é iniciado com os requisitos definidos para o produto a ser desenvolvido e do projeto para o desenvolvimento. Este plano cobre várias atividades da engenharia e do gerenciamento do projeto, incluindo o nível apropriado do acompanhamento do projeto, a frequência das revisões do progresso e as medidas a serem utilizadas. O progresso é determinado basicamente pela comparação do realizado em relação ao planejado. Quando o realizado desvia significativamente do planejado, ações corretivas devem ser tomadas e gerenciadas para resolver o problema. Estas ações podem envolver o replanejamento.

O gerenciamento dos acordos com os fornecedores é realizado quando o projeto decide adquirir componentes do trabalho de fornecedores. Quando este componente é identificado, um fornecedor que produza este componente é selecionado e um acordo é estabelecido e mantido para gerenciar este fornecimento. O progresso e desempenho do fornecedor são acompanhados. Testes e revisões de aceitação são conduzidos nos produtos e componentes fornecidos.

As áreas de processo IPM, RSKM, IT, ISM e QPM representam, na visão do CMMI, uma gerência de projetos avançada, compatível com organizações nos níveis 3 e superiores de maturidade, segundo a representação por estágio. Estas cinco áreas de processo definem referências para o estabelecimento de um processo

definido que é adaptado do conjunto de processos padrões; a coordenação e colaborações com os interessados, incluindo os fornecedores; o gerenciamento dos riscos; a construção e manutenção de equipes integradas para a condução dos projetos; e o gerenciamento quantitativo do processo definido do projeto.

Em relação à disciplina de gerência de projeto, o modelo CMMI apresenta um conjunto compreensivo de áreas de processo para orientar a implantação e evolução das atividades desta gerência. O CMMI evoluiu e ampliou as orientações para gerência de projeto do SW-CMM e dos outros modelos precursores. A representação como um modelo contínuo, consistente com a futura Norma ISO/IEC 15504, permite uma maior flexibilidade na utilização das áreas de processo, facilitando a obtenção de melhorias adicionais.

Como o objetivo do CMMI é representar no modelo metas e recomendações genéricas para orientar a melhoria dos processos em geral, não são descritas soluções prontas para serem executadas nas organizações. Cabe a cada organização entender e interpretar estas orientações no contexto, objetivo e estratégia de negócio da organização para obter melhorias relevantes.

### **2.2.2 Disciplinas do Modelo**

O objetivo do CMMI é ser um framework extensível, para poder acrescentar novas disciplinas. Cada modelo do CMMI inclui uma ou mais disciplinas, de forma integrada. Atualmente os modelos do CMMI cobrem 4 disciplinas:

- Engenharia de Sistemas (*System Engineering SE*)
- Engenharia de Software (*Software Engineering SW*)
- Desenvolvimento de produtos e processos integrados (*Integrated Product and Process Development IPPD*)
- Aquisição (*Supplier Sourcing SS*)

Conforme resumido na Figura 2.2, apresentada anteriormente, a base dos modelos CMMI são as áreas de processo, que são organizadas em níveis de maturidade (na representação por estágio) e em categorias de processo (na representação contínua).

Na representação por estágio, as áreas de processo são organizadas por quatro níveis de maturidade. A seguir estão relacionadas as áreas de processo em cada nível de maturidade. Cada área de processo é identificada por uma sigla (baseada no nome em inglês), uma tradução do nome para o português e o nome original em inglês:

Nível 2:

REQM: Gestão de Requisitos (*Requirements Management*)

PP: Planejamento de Projeto (*Project Planning*)

PMC: Acompanhamento e Controle de Projeto (*Project Monitoring and Control*)

SAM: Gestão de Acordos com Fornecedores (*Supplier Agreement Management*)

CM: Gestão de Configuração (*Configuration Management*)

PPQA: Garantia da Qualidade de Processo e Produto (*Process and Product Quality Assurance*)

MA: Medição e Análise (*Measurement and Analysis*)

Nível 3:

RD: Desenvolvimento de Requisitos (*Requirements Development*)

TS: Solução Técnica (*Technical Solution*)

PI: Integração de Produto (*Product Integration*)

VER: Verificação (*Verification*)

VAL: Validação (*Validation*)

OPF: Foco no Processo Organizacional (*Organizational Process Focus*)

OPD: Definição do Processo Organizacional (*Organizational Process Definition*)

OT: Treinamento Organizacional (*Organizational Training*)

IPM: Gestão Integrada de Projeto (*Integrated Project Management*)

RSKM: Gestão de Riscos (*Risk Management*)

DAR: Análise de Decisão e Resolução (*Decision Analysis and Resolution*)

Nível 4:

QPM: Gestão Quantitativa de Projeto (*Quantitative Project Management*)

OPP: Desempenho do Processo Organizacional (*Organizational Process Performance*)

Nível 5:

CAR: Análise de Causa e Resolução (*Causal Analysis and Resolution*)

OID: Inovação e Melhoria Organizacional (*Organizational Innovation and Deployment*)

Na representação contínua, as áreas de processo são organizadas em quatro categorias de processo. A seguir estão relacionadas as áreas de processo em cada categoria de processo. Cada área de processo é identificada por uma sigla (baseada no nome em inglês), uma tradução do nome para o português e o nome original em inglês:

Categoria Gestão de Projeto:

PP: Planejamento de Projeto (*Project Planning*)

PMC: Acompanhamento e Controle de Projeto (*Project Monitoring and Control*)

SAM: Gestão de Acordos com Fornecedores (*Supplier Agreement Management*)

IPM: Gestão Integrada de Projeto (*Integrated Project Management*)

RSKM: Gestão de Riscos (*Risk Management*)

QPM: Gestão Quantitativa de Projeto (*Quantitative Project Management*)

Categoria Gestão de Processo:

OPF: Foco no Processo Organizacional (*Organizational Process Focus*)

OPD: Definição do Processo Organizacional (*Organizational Process Definition*)

OT: Treinamento Organizacional (*Organizational Training*)

OPP: Desempenho do Processo Organizacional (*Organizational Process Performance*)

OID: Inovação e Melhoria Organizacional (*Organizational Innovation and Deployment*)

Categoria Engenharia:

REQM: Gestão de Requisitos (*Requirements Management*)

RD: Desenvolvimento de Requisitos (*Requirements Development*)

TS: Solução Técnica (*Technical Solution*)

PI: Integração de Produto (*Product Integration*)

VER: Verificação (*Verification*)

VAL: Validação (*Validation*)

Categoria Apoio:

CM: Gestão de Configuração (*Configuration Management*)

PPQA: Garantia da Qualidade de Processo e Produto (*Process and Product Quality Assurance*)

MA: Medição e Análise (*Measurement and Analysis*)

CAR: Análise de Causa e Resolução (*Causal Analysis and Resolution*)

DAR: Análise de Decisão e Resolução (*Decision Analysis and Resolution*)

Em relação ao SW-CMM, o CMMI por estágio é uma revisão deste modelo, com ajustes. No nível 2 de maturidade, por exemplo, foi incluída a área de processo de medição e análise como uma ampliação deste assunto, que já era coberto em parte em cada área do SW-CMM. No nível 3, por exemplo, a área de engenharia de produto do SW-CMM foi melhor descrita por meio de cinco áreas de processo: Desenvolvimento de Requisitos, Solução Técnica, Integração de Produto, Verificação, e Validação. Outras mudanças ocorrem para refletir melhor a orientação para atendimento dos níveis de maturidade.

Em relação à utilização para melhoria de processo, o CMMI por estágio sugere abordagens semelhantes às aquelas utilizadas com sucesso com o SW-CMM. Em relação ao CMMI contínuo, a tendência é toda a experiência de utilização da ISO/IEC 15504, ser aproveitada para ajustar as abordagens já utilizadas com sucesso pelo SW-CMM.

### **2.2.3 Elementos do Modelo**

Os elementos que compõe o modelo CMMI são agrupados em três categorias que refletem como eles devem ser interpretados:

**Elementos Requeridos:** Metas específicas e as genéricas são elementos requeridos do modelo. Estes elementos devem ser estabelecidos pelos processos definidos, e implementados pela organização. São essenciais para avaliar a satisfação de uma área de processo. O estabelecimento (ou satisfação) das metas é usado em avaliações como base para satisfação da área de processo e maturidade organizacional. Apenas a declaração das metas é um elemento requerido do modelo, o título e qualquer nota associada são considerados elementos informativos do modelo.

**Elementos Esperados:** Práticas específicas e as genéricas são elementos esperados no modelo. Estes elementos descrevem o que uma organização tipicamente irá implementar para satisfazer um componente requerido. Servem como guia para aqueles que implementam melhorias e ou executam avaliações. As práticas descritas, ou as alternativas aceitáveis para elas, espera-se que estejam presentes nos processos planejados e implementados da organização antes que as



metas sejam consideradas satisfeitas. Apenas a declaração da prática é um elemento esperado do modelo, o título e qualquer nota associada são considerados elementos informativos do modelo.

Componentes Informativos: Subpráticas, produtos de trabalho típicos, particularidades da disciplina, elaborações de práticas genéricas, títulos, notas e referências são elementos informativos do modelo. Estes elementos ajudam o usuário do modelo a entender as metas e práticas e como elas podem ser estabelecidas, fornecendo detalhes para ajudar a começar a pensar em como estabelecer as práticas e metas.

Características comuns são elementos não classificados do modelo, apenas constituem um grupo que fornece uma maneira de apresentar as práticas genéricas.

Quando se usa um modelo CMMI como guia, processos são planejados e implementados em conformidade com os elementos esperados e requeridos das áreas de processo. Conformidade com uma área de processo significa que nos processos planejados e implementados existe um processo associado (ou processos) que endereça as práticas específicas e genéricas da área de processo, ou alternativas, que geram um resultado de acordo com a meta associada com aquela prática específica ou genérica.

### 3 FRAMEWORK SCRUM

Dentre as várias metodologias ágeis existentes, podemos destacar o XP, do inglês, *Extreme Programming* (BECK, 2000) e o *Lean Development* (POPPENDIECK, 2003) para o desenvolvimento de software, e o *Scrum*, para o gerenciamento de projetos, como as mais citadas em artigos e livros sobre o tema.

O *Scrum* foi criado na empresa de fabricação de automóveis e produtos Toyota por Takeuchi e Nonaka e a primeira utilização do termo ocorreu no livro "The New Product Development Game" (TAKEUCHI & NONAKA, 1986). Eles notaram que projetos usando equipes pequenas e multidisciplinares (*cross-functional*) produziam os melhores resultados, e associaram estas equipes altamente eficazes à formação do *Scrum*, uma jogada do *Rugby*, esporte coletivo semelhante ao futebol americano.

Jeff Sutherland documentou, concebeu e implantou o *Scrum* na empresa Easel Corporation, em 1993, incorporando estilos de gerenciamento observados por Takeuchi e Nonaka. Em 1996, Ken Schwaber formalizou a definição de *Scrum* e ajudou a implantá-lo no gerenciamento de projetos de software ao redor do mundo. Desde então, o *Scrum* se tornou uma das principais alternativas às metodologias tradicionais. O *Scrum* tem sido adotado por gestores que buscam garantir a obtenção do melhor produto possível e por engenheiros que buscam garantir que farão seu trabalho da melhor forma possível, sendo usado em milhares de projetos por todo o mundo (SCHWABER & BEEDLE, 2002).

O *Scrum* é uma evolução na forma de se pensar e desenvolver projetos de software, mantendo sempre o foco na relação entre os principais envolvidos no processo de desenvolvimento: o desenvolvedor, o gestor e o cliente. O uso de *Scrum* implica na colaboração e cumplicidade entre todas as áreas, dando a TI a visibilidade de apoiadora, viabilizadora e aliada.

Não há a definição de um método específico para o desenvolvimento do software no *Scrum*, mas são definidas regras e práticas de gestão que devem ser adotadas para garantir o sucesso de um projeto. As análises de estudos de caso como os apresentados por KNIBERG (2007), QUMER (2008) e (CHENG, 2009) mostram que a utilização de um método bem definido para implantação das práticas do *Scrum* pode aumentar a agilidade no desenvolvimento e trazer um maior retorno do investimento realizado no projeto em menos tempo.

O *Scrum* é fundamentado na teoria de controle de processos empíricos e emprega uma abordagem iterativa e incremental para aperfeiçoar a previsibilidade e controlar riscos. Seguindo esta abordagem, tem-se a produção frequente de incrementos que podem ser testados, ajustados, documentados e expandidos e para tal, as iterações devem ser curtas para promover visibilidade para o desenvolvimento. Três pilares sustentam qualquer implementação de controle de processos empíricos, são eles: transparência, inspeção e adaptação.

O primeiro pilar é a transparência que garante que aspectos do processo que afetam o resultado devem ser visíveis para aqueles que gerenciam os resultados. Esses aspectos não apenas devem ser transparentes, mas também o que está sendo visto deve ser conhecido. Isto é, quando alguém que inspeciona um processo acredita que algo está pronto, isso deve ser equivalente à definição de pronto utilizada.

O segundo pilar é a inspeção, pois diversos aspectos do processo devem ser inspecionados com uma frequência suficiente para que variações inaceitáveis no processo possam ser detectadas. A frequência da inspeção deve levar em consideração que qualquer processo é modificado pelo próprio ato da inspeção. O problema acontece quando a frequência de inspeção necessária excede a tolerância do processo à inspeção. Os outros fatores relacionados são a habilidade e a aplicação das pessoas em inspecionar os resultados do trabalho.

A adaptação é o terceiro pilar, pois muitas vezes a partir da inspeção, é observado que um ou mais aspectos do processo estão fora dos limites aceitáveis e que conseqüentemente o produto resultante será inaceitável. Neste caso, ajustes deverão ser feitos no processo ou no software sendo desenvolvido. Esses ajustes devem ser efetuados o mais rápido possível para minimizar desvios posteriores.

Existem três pontos para inspeção e adaptação em Scrum. A Reunião Diária é utilizada para inspecionar o progresso em direção à Meta da Sprint (tempo de 2 a 4 semanas onde o software é desenvolvido) e para realizar adaptações que otimizem o valor do próximo dia de trabalho. Além disso, as reuniões de Revisão de Sprint e de Planejamento de Sprint são utilizadas para inspecionar o progresso em direção à Meta da Versão para Entrega e para fazer as adaptações que otimizem o valor da próximo Sprint.

Finalmente, a Retrospectiva da Sprint é utilizada para revisar o Sprint passado e definir que adaptações tornarão a próxima Sprint mais produtiva, recompensadora e

gratificante. A Figura 4 demonstra a visão geral do processo.

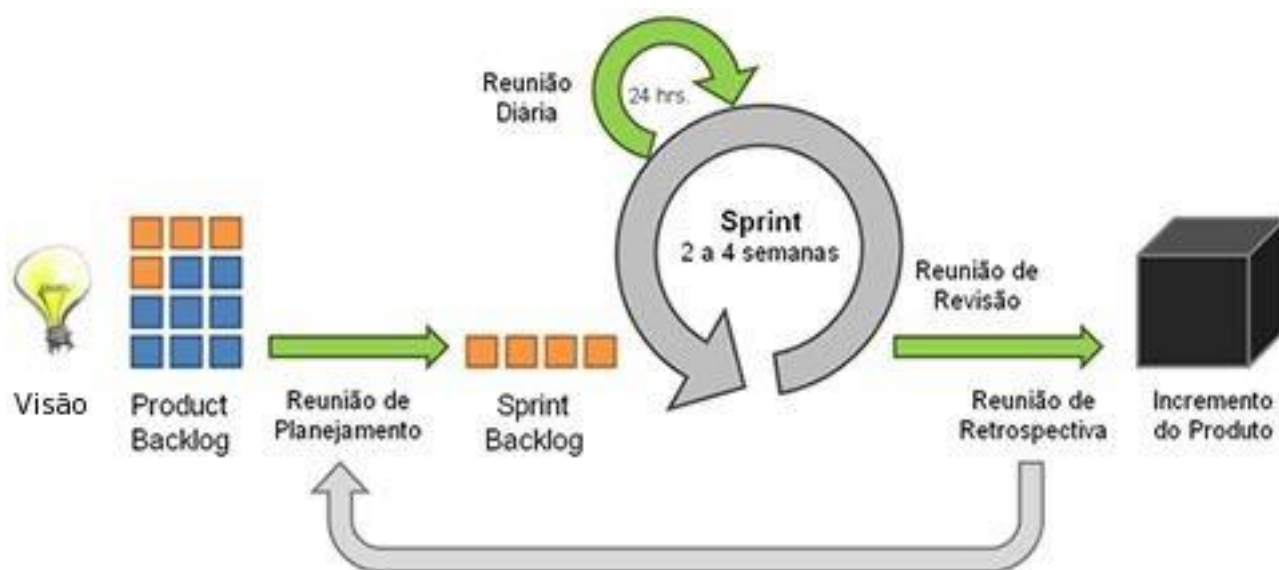


Figura 4 - *Scrum*: Visão do Processo

Equipes *Scrum* são projetadas para otimização, flexibilidade e produtividade. Para isso, eles são auto-organizáveis, multifuncionais e trabalham em interações. A auto-organização é um aspecto essencial no *Scrum*, pois permite que a equipe se auto gerencie, aperfeiçoando a colaboração, aumentando a moral da equipe, além de aumentar a motivação e responsabilidade no cumprimento dos prazos. Segundo (SCHWABER & BEEDLE, 2002), cada equipe *Scrum* possui três papéis:

O *Scrum Master* é responsável por garantir que o *Scrum* seja compreendido e seguido na empresa, remover impedimentos (elementos que estejam impedindo o time de completar uma tarefa) e auxiliar o time a ser auto-organizável. O *Scrum Master* não faz parte do time e não o gerencia;

O *Product Owner* (Dono do Produto) é responsável por maximizar o valor do trabalho do time *Scrum* através da priorização dos itens do *Backlog* do Produto (lista de funcionalidades que serão desenvolvidas no projeto) e garantir a sua visibilidade. É importante que este papel seja realizado por apenas uma pessoa por projeto de forma que não haja divergências e conflitos de opiniões. O *Product Owner* pode ser aconselhado por outras pessoas, mas é ele quem tem a palavra final;

O Time realiza o trabalho, transformando os itens do *Backlog* do Produto em incrementos de produto potencialmente entregáveis a cada Sprint, devendo ser interdisciplinar, sem títulos e auto-organizável. O tamanho do time deve ser de 5 a 9 pessoas, de forma a otimizar a comunicação e produtividade.

*Scrum* emprega os eventos com duração fixa (*time-boxes*) para criar regularidade e facilitar o controle, dentre esses eventos temos: a reunião de planejamento de release, a revisão da Sprint, a retrospectiva da Sprint e a reunião diária, detalhadas a seguir.

A Reunião de Planejamento do *Release* tem como objetivo estabelecer um plano de metas para a release (grupo de *Sprints* necessários para atingir uma meta), maiores prioridades do *Backlog* do Produto, os principais riscos e funcionalidades que deverão estar contidas no *release* e uma data de entrega e custo prováveis se houverem poucas alterações. Esta reunião é a única opcional, mas é importante para reduzir os impedimentos durante os *Sprints*;

A Sprint é o coração do *Scrum*. Consiste em uma iteração de duas a quatro semanas de duração, onde ocorre o esforço de desenvolvimento. Todas as Sprints utilizam o mesmo modelo de *Scrum* e todas as *Sprints* têm como resultado um incremento do produto final que é potencialmente entregável. Cada Sprint começa imediatamente após a anterior;

A Reunião de Planejamento da Sprint é o momento no qual a iteração é planejada. Tem um *time-box* de 5% do tempo do Sprint e é dividida em duas partes. Na primeira, com o auxílio do *Product Owner*(PO), o time define o que vai ser feito no Sprint. Na segunda parte, o time entende como desenvolverá cada item do *Backlog* do Produto. Nesta reunião, o PO também define a meta da Sprint;

A Revisão da Sprint também possui um *time-box* de até 5% do tempo da Sprint. É a reunião onde o time mostra ao PO o que foi realizado e responde a questionamentos. Neste momento, o PO aceita ou rejeita o resultado de cada estória do Sprint e gera discussões que servem como entradas para o planejamento dos *Sprints* seguintes;

A Retrospectiva da Sprint ocorre após a Revisão da Sprint e é onde o time revisa o modelo de trabalho e práticas do processo de forma a adaptá-lo para um melhor aproveitamento. O resultado desta reunião é uma lista de itens que foram realizados com sucesso e outra dos que devem ser melhorados para o próximo Sprint;

A Reunião Diária é onde o time se encontra diariamente para uma reunião de 15 minutos como uma forma de melhorar a comunicação, eliminar outras reuniões, identificar e remover impedimentos e melhorar o conhecimento de todos sobre o andamento do projeto. Nela, cada membro do time responde a três perguntas: 1) O que ele realizou desde a última reunião diária; 2) o que ele vai fazer antes da

próxima reunião diária; e 3) quais os obstáculos estão em seu caminho. Ela não é uma reunião de Status para o *Scrum Master*. A reunião diária serve como uma inspeção do time sobre o cumprimento da meta estabelecida para o Sprint. É parte importante do processo de inspeção e adaptação do *Scrum*.

*Scrum* utiliza três artefatos principais, que são o *Backlog* do Produto, *Backlog* da Sprint e o Sprint *Burndown*.

O *Backlog* do Produto é uma lista priorizada de tudo que pode ser necessário no produto. À medida que o projeto e o produto a ser desenvolvido evoluem mais itens podem ser acrescentados ou removidos. Um item do *Backlog* do Produto é chamado de estória. Ele deve ser constantemente atualizado e priorizado pelo *Product Owner*.

O *Backlog* da Sprint é uma lista de tarefas para transformar os itens do *Backlog* do Produto que tenham maior prioridade, no período de um Sprint, em um incremento de produto a ser entregue. O time é responsável por criar e estimar as tarefas do *Backlog* da Sprint na reunião de planejamento do Sprint, e, se necessário criar ou remover tarefas durante a execução do Sprint. O *Backlog* da Sprint sempre deve conter tarefas que derivem de estórias que auxiliem a atingir a meta do Sprint.

O gráfico Sprint *Burndown* mede os itens do *Backlog* da Sprint restantes ao longo do tempo de um Sprint. O Sprint *Burndown*, mostrado na Figura 5, é um gráfico que possui, no eixo x o número de dias da Sprint e no eixo y a estimativa de horas do time para completar todas as estórias selecionadas para o Sprint. Este número de horas vai caindo à medida que algumas estórias são concluídas (linha irregular do gráfico). Uma linha é traçada do maior valor de X até o maior valor de Y que serve como guia da produtividade do time (SCHWABER & SUTHERLAND, 2009).

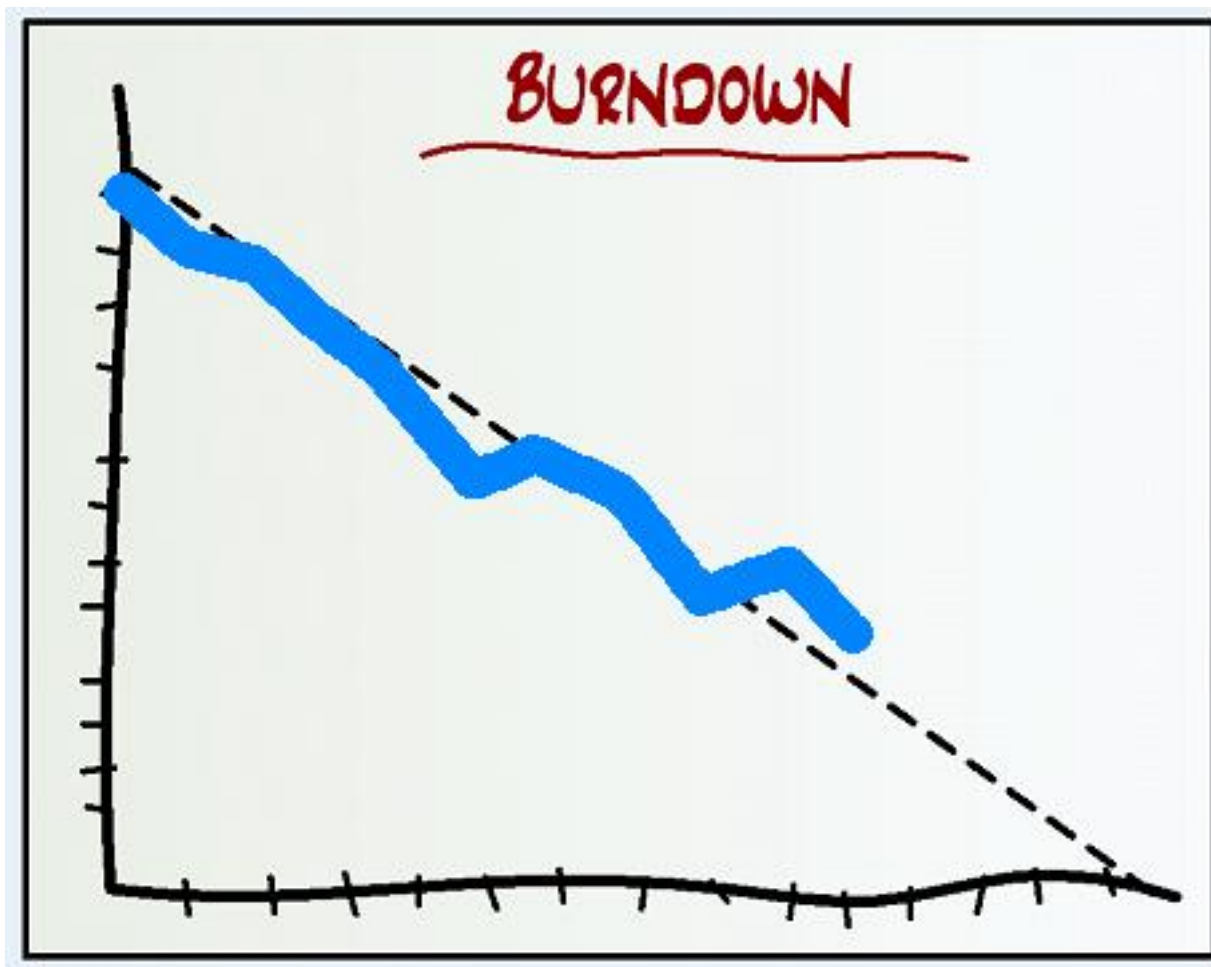


Figura 5 - Gráfico *Burndown*

Outro artefato importante para o time é o quadro *Scrum*. Ele mostra todo o trabalho que está sendo realizado dentro do Sprint, sendo atualizado constantemente no decorrer deste. Tarefas podem ser adicionadas ou removidas e estimativas podem ser modificadas. Na maior parte dos casos, junto com as tarefas que estão sendo realizadas, também ficam no quadro *Scrum* a meta do Sprint, definida pelo *Product Owner*, os Impedimentos – problemas que o time não consegue resolver internamente, precisando de auxílio externo – e o gráfico de Sprint *Burndown*.

O quadro *Scrum*, tradicionalmente, é composto por três colunas: *To Do*, *Doing* e *Done*; pelo objetivo do Sprint: uma frase que guia o desenvolvimento do time e o aceite das estórias; um gráfico de *burndown*, que auxilia na medição da velocidade do time durante o Sprint e gerenciamento de riscos; itens não planejados, como correções de bugs; e itens extras para o caso de o time finalizar as tarefas antes do

planejado. A Figura 6 demonstra um quadro *Scrum* típico.

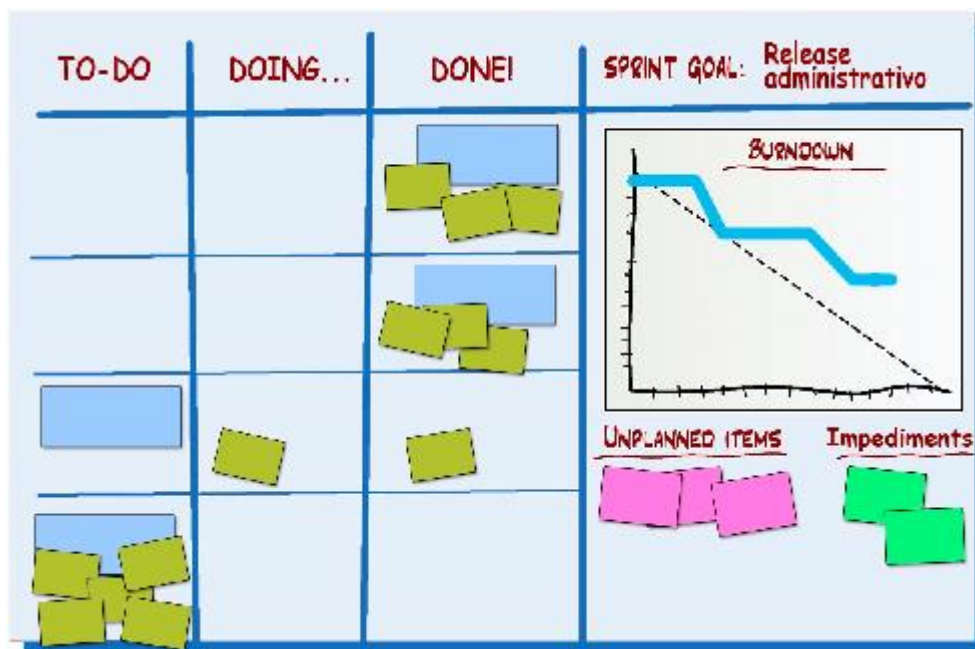


Figura 6 - Quadro *Scrum*



## 4 RELATO DE EXPERIÊNCIA DA IMPLANTAÇÃO DO NÍVEL 2 DO CMMI-DEV EM UMA EMPRESA DE MANUTENÇÃO

Neste capítulo é descrito um relato de implementação do Nível de Maturidade 2 do CMMI-DEV utilizando conceito da metodologia *Scrum*, em um empresa de software de pequeno porte. A abordagem utilizada foi desenvolvida pela empresa SWQuality, e por questões de propriedade intelectual, as diretrizes e detalhes da implementação desta abordagem não são descritas.

Este capítulo apresenta, inicialmente, os objetivos e a estratégia do projeto de melhoria de processo realizado; descreve o cenário da empresa em que o projeto de melhoria foi realizado; relata a execução do projeto de melhoria, com exemplos de evidências geradas e por fim faz um mapeamento entre as evidências desenvolvidas e as Práticas específicas do Nível 2 do CMMI-DEV.

A fim de zelar pela confidencialidade das informações, não serão disponibilizados dados que identifiquem pessoas, produtos ou organizações.

### 4.1 Objetivos e abordagem de melhoria

O projeto de melhoria de processo descrito neste capítulo teve como principal objetivo a implementação do Nível 2 de Maturidade do CMMI-DEV em uma empresa de desenvolvimento e manutenção de software do Estado da Paraíba. Dado que o objetivo do projeto era o Nível 2 do CMMI, o projeto adotou como metas a definição e institucionalização das Áreas de Processo (PA):

- Planejamento do Projeto - *Project Planning* (PP);
- Monitoramento e Controle do projeto - *Project Monitoring and Control* (PMC);
- Gestão de Requisitos - *Requirement Management* (REQM);
- Medição e Análise - *Measurement and Analysis* (MA);
- Gerência de Configuração - *Configuration Management* (CM);
- Garantia da Qualidade - *Process and Product Quality Assurance* (PPQA);

Para isto, a equipe de consultores responsável empregou uma metodologia de implantação de melhoria de processo baseada no modelo IDEAL, realizando ciclos bimestrais de melhoria (conforme descrito na subseção 4.3).

Desde o início do projeto, foi levado em conta que esta implementação do modelo de maturidade de processos de desenvolvimento de software iria ser uma mudança muito grande no dia-a-dia e no modo de trabalho dos colaboradores da empresa e, também, que iria ser encontrada muita resistência por parte dos mesmos..

De modo a tentar combater estes problemas foi determinado que a definição dos processos e criação de documentos fossem o mais de acordo possível com as pretensões e necessidades de trabalho diárias de todos os colaboradores, ou seja, seria definido um processo de mudança mas com o mínimo de impacto na sua forma de trabalho diária na organização. Todas as definições foram feitas com o *feedback* de todos os membros da organização.

A estratégia definida foi a de tentar definir e criar apenas a documentação necessária e mínima que permitisse dar resposta aos processos do nível 2 do CMMI.

## 4.2 Caracterização da Empresa

A empresa alvo deste estudo é uma empresa de pequeno porte da Paraíba, que atua no desenvolvimento e manutenção de software, fornecendo soluções para gestão de vendas, estoque, emissão de notas através de um portfólio de produtos mantidos. Tem uma extensa base de clientes ( tem domínio de 7% dos clientes em seu segmento de mercado). As principais tecnologias adotadas são Delphi, PHP e *Javascript*.

A empresa é estruturada nos seguintes setores:

- Atendimento (2 pessoas): Registra chamados de cliente e abre ocorrências;
- Suporte (6 pessoas): Atende ocorrências e identifica bugs dos produtos;
- Teste (1 pessoa): Analisa se as ocorrências reportadas pelo suporte são bugs e os encaminha ao desenvolvimento quando necessário;
- Desenvolvimento (4 pessoas): Responsável pela manutenção e evolução dos produtos

Em relação ao desenvolvimento, foco do trabalho de melhoria de processo neste contexto, os principais problemas apontados eram:

- Necessidade de organização do trabalho;

- Falta de visibilidade;
- Insatisfação dos clientes;
- Adaptação as mudanças;
- Atraso na liberação de versões;
- Interferências externas.

### **4.3 Descrição do caso**

A execução do projeto de melhoria de processo foi organizado em 6 etapas, sendo uma etapa inicial de diagnóstico e planejamento, e cinco ciclos de implantação da melhoria (Fase 1 a Fase 5) adotando os princípios do modelo IDEAL, com periodicidade bimestral. Cada etapa é descrita, nas subseções a seguir, a respeito de seus objetivos, resultados, dificuldades e lições aprendidas.

#### **4.3.1 Diagnóstico e planejamento**

Para elaborar o planejamento das atividades e saber o estado da empresa, com relação aos elementos requeridos no nível 2 do modelo CMMI, foi feito um diagnóstico, levantando todos os pontos de melhoria, considerando os processos atuais existentes na organização.

Para realização do diagnóstico foi feito um *Gap Analysis*. O *Gap Analysis* é um estudo formal do que o negócio faz e onde se quer posicionar no futuro. É conduzido na perspectiva da organização, da direção ou processos do negócio e/ou das tecnologias da Informação.

O objetivo de um *Gap Analysis* é encontrar as diferenças entre as práticas em uso numa dada organização e o guia dado por um modelo de referência, neste caso o CMMI, ou seja, é uma ferramenta que permite a organização determinar, documentar e aprovar a variância entre a sua performance atual e os requisitos do negócio. Tal análise pode ser feita a um nível operacional ou estratégico.

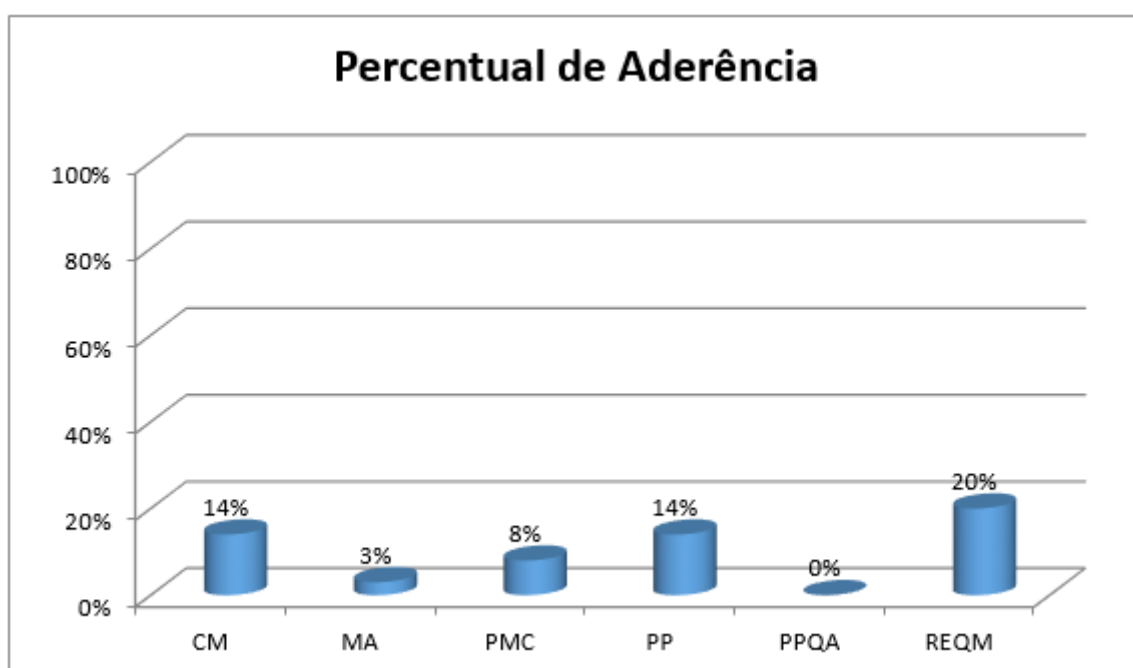
Foi realizado um Gap, considerando as áreas de processo do nível 2 do modelo

CMMI, cujo seu propósito foi: (i) identificar das práticas da engenharia e gestão usadas no desenvolvimento do software e atividades de manutenção; (ii) identificar pontos fortes e pontos de melhoria nos processos da organização; (iii) identificar o grau de satisfação para as áreas de processo examinadas e (iv) estabelecer recomendações e ações a serem tomadas para a melhoria dos processos.

O Gap contemplou as atividades da equipe de desenvolvimento (manutenção e evolução dos produtos da organização), assim participaram do diagnóstico como entrevistados:

- 1 Diretor de TI;
- 1 Gerente de desenvolvimento;
- 4 Desenvolvedores;
- 1 Testador

Neste diagnóstico, constatou-se que a organização encontrava-se no Nível 1 (Inicial) de maturidade do modelo CMMI. O diagnóstico constatou que as práticas realizadas pela empresa eram insatisfatórias em relação ao esperado para o Nível 2 do CMMI, sendo as práticas relacionadas a PPQA as mais críticas, seguida pelas práticas de MA, PMC, PP, CM e REQM. O resultado da aderência entre as práticas observadas no diagnóstico e as recomendações do CMMI pode ser observado na Figura 7.



### Figura 7 - Resultado do *Gap Analysis*

Foi observado que a PA SAM (*Software Acquisition Management* - Gestão de Aquisição de Software) não se aplicava aos negócios da empresa, assim não foi avaliada durante o *Gap Analysis* realizado. Uma vez que é uma prática não obrigatória em uma avaliação oficial do nível 2 do CMMI, a exclusão desta PA não traz impactos ao projeto de melhoria de processo realizado.

Com base no resultado identificado, um plano de melhoria foi estabelecido definindo uma estratégia para a implantação do CMMI na empresa.

#### 4.3.2 Fase 1

As metas definidas para a primeira fase foram a implantação do *Scrum*, a implantação de uma ferramenta gerencial para registro e acompanhamento de atividades, a institucionalização de auditorias de qualidade sobre as atividades do *Scrum* e a definição de um conjunto de indicadores.

Nesta fase, os processos do setor de desenvolvimento foram reformulados conforme o programa de melhoria da qualidade utilizando como referência metodologias ágeis. Foi ministrado um curso introdutório da metodologia *Scrum* e depois um acompanhamento da introdução e adaptação da metodologia no ambiente da empresa.

O primeiro passo foi definir os papéis e responsabilidades necessários para execução do *Scrum*, o *Product Owner*, *Scrum Master* e Time. O Diretor de TI assumiu o papel de *Product Owner*. O Coordenador de Desenvolvimento assumiu o papel de *Scrum Master*. Os quatro desenvolvedores juntamente com o testador formaram o Time *Scrum*.

A duração definida para as *Sprints* foi de uma semana. Dois fatores foram considerados para escolher esse *time-box*: (i) ciclos menores agilizam a institucionalização das cerimônias e (ii) garantem maior estabilidade do escopo planejado (*sprints* maiores estariam muito suscetíveis a mudanças, dado o histórico da organização).

A implantação do *Scrum* permitiu a gestão do desenvolvimento em ciclos curtos,

com cerimônias em marcos (planejamento e revisão). A partir deste ponto, foi possível organizar o escopo em forma de um *backlog* de histórias e bugs para as demandas da equipe de desenvolvimento. O *backlog* foi mantido em um quadro para facilitar a visibilidade do seu tamanho e, através de uma organização de cores, facilitar a análise do mesmo (Figura 8).



Figura 8 – *Backlog*

As cerimônias estabelecidas foram: (i) planejamento de *sprint*, (ii) revisão de *sprint*, (iii) retrospectiva. A primeira cerimônia (i) tem o propósito de estabelecer

quais os itens de *backlog* que serão trabalhados durante a *sprint*. Ao longo da *sprint* as atividades são acompanhadas através de um quadro de atividades (Figura 9), atualizado diariamente durante reuniões diárias, permitindo o acompanhamento das demandas e o registro e comunicação de impedimentos.

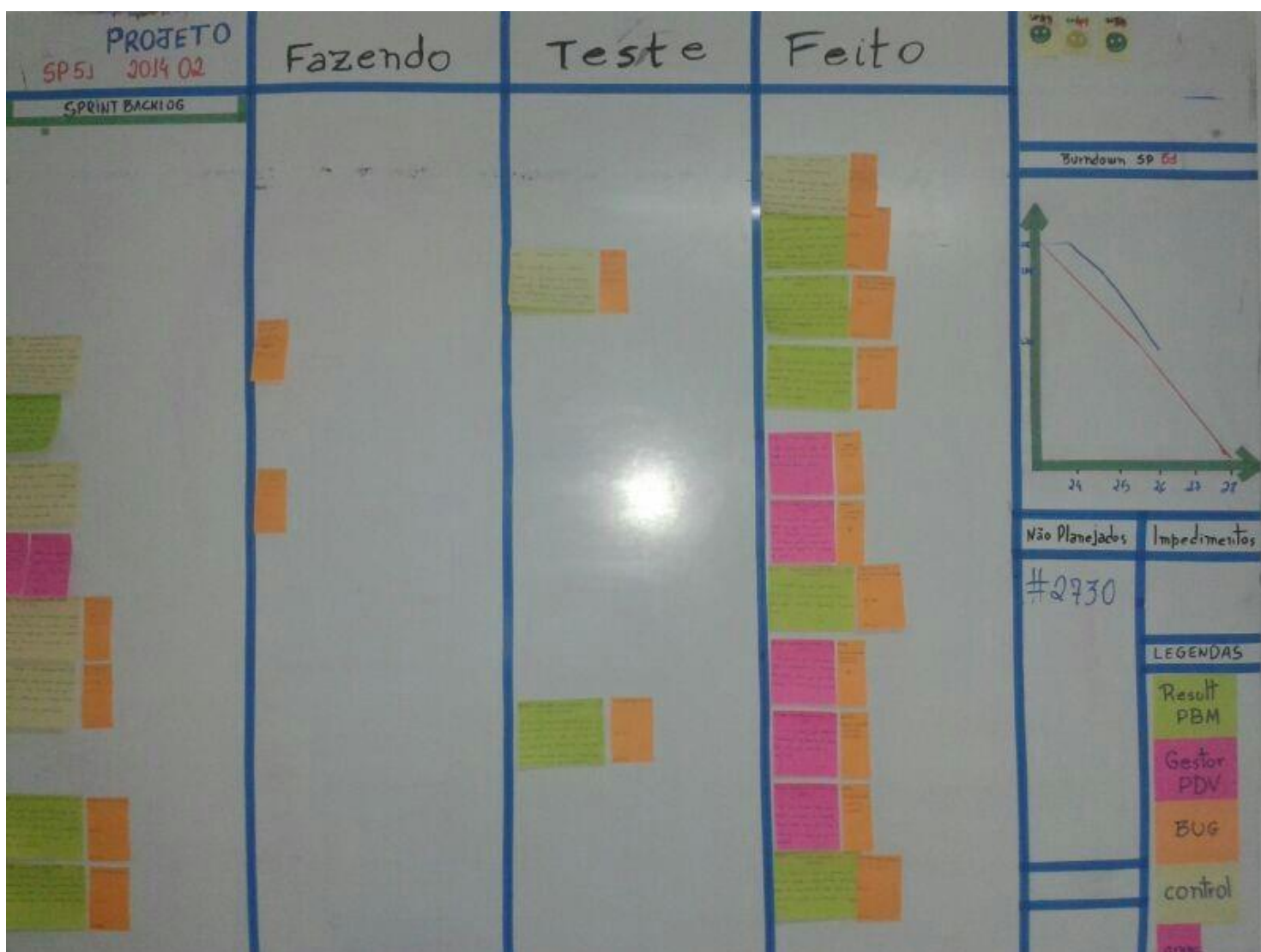


Figura 9 - Quadro de Atividades

Ao término da duração da *sprint*, as atividades desempenhadas durante a *sprint* são validadas junto ao *Product Owner* (ii) e é realizado um levantamento dos pontos fortes, fracos e oportunidades de melhoria da *sprint* (iii).

Ao longo da institucionalização do *Scrum*, a ferramenta *Redmine* foi selecionada e configurada para fornecer suporte a metodologia definida. O *Redmine* é uma

ferramenta de software livre para a gestão de projetos, fornecida sobre licença GNU *General Public License* [REDMINE, 2014]. A Figura 10 ilustra a adaptação do *Redmine* para o acompanhamento das *Sprints* e a Figura 11 apresenta uma ata de cerimônia registrada na ferramenta.



**Figura 10- Representação de uma Sprint no *Redmine***



Nome	Papel
[REDACTED]	P.O
[REDACTED]	Scrum Master
[REDACTED]	Time
[REDACTED]	Time
[REDACTED]	Time
[REDACTED]	Time
[REDACTED]	Time

**Objetivo e pauta**

O objetivo desta reunião é apresentar o plano do projeto e realizar a SP1 que é o Product Owner apresentando as histórias priorizadas de acordo com o backlog, o time de desenvolvimento aceitando as histórias para serem realizadas no Sprint e a SP2 onde o time irá subdividir as histórias em tarefas menores para poderem ser realizadas.

**Itens discutidos**

**Apresentação do Projeto**

Foi apresentado o Planejamento do Projeto Wolverine para a equipe, onde foi feita uma revisão técnica e gerencial pelos participantes, tendo sido aprovado por todos sem nenhuma alteração.

**Sprint Planning 1**

O time se comprometeu a entregar 71 pontos, conforme arquivo PDF em anexo.

- O P.O apresentou todas as histórias definindo junto com o time os requisitos, teste e critérios de validação. O time aceitou as histórias de acordo com os critérios de aceitação que são clareza, pontuação da história, coesão, dificuldade.

**Sprint Planning 2**

Nessa reunião o time dividiu cada histórias em tarefas e discussões feitas para repassar o conhecimento.

Escopo Planejado.pdf (340,96 KB) [REDACTED] 03/12/2013 17:09

**Figura Error! Bookmark not defined. - Ata de Planejamento de Sprint**

Após a execução de seis *Sprints*, foi concluída a institucionalização do *Scrum*. Para garantir que o processo fosse mantido, foi criado um *Checklist* de Auditoria, que tem como objetivo garantir que todos as cerimônias, práticas, métodos e técnicas consideradas importantes para organização são seguidas.

A periodicidade definida para as auditorias foram semanais. Ao final de cada Sprint, o *checklist* era aplicado para verificar a aderência do processo executado em relação ao processo definido. O *checklist* pode ser visualizado da Figura 11.

Garantia da Qualidade - História #45157: Realizar a auditoria das equipes

### Auditoria de Encerramento da Sprint 14 e Planejamento da Sprint 15

Adicionado por [Redacted]

Situação: Validado  
 Prioridade: Normal  
 Atribuído para: [Redacted]  
 Versão: Sprint 15  
 Percentual de Aderência: 81.81

% Terminado: 100%  
 Tempo gasto: 2.50 horas

Descrição [Responder](#)

Checklist do Processo

Encerramento da SPRINT 14				
Id	Área	Item	Situação	Observação
1	PMC, REQM	Todos os tickets da sprint foram finalizados?	NC	#44500, #44504, #44505, #44506
2	PMC	Os tickets da sprint com a situação "validado" possuem horas apontadas?	C	
3	PMC, MA	O apontamento de horas foi realizado pelo time dentro da meta de 70%?	C	
4	REQM	Os tickets rejeitados e cancelados foram duplicados e estão relacionados em outra sprint? (Tem que estar relacionado com o termo "copiada")	C	
5	REQM	Os tickets rejeitados e cancelados possuem notas de justificativa para o resultado final?	C	
6	REQM	Os tickets não planejados possuem uma análise de impacto da mudança no comentário do ticket?	NA	
7	PMC	As cerimônias de review e retrospectiva da sprint foram registradas em ata, conforme template do ticket scrum?	C	
8	PMC	Na ata de encerramento, seção de review, o time registrou um resumo dos pontos planejados, validados e rejeitados?	C	

Figura 11 - Checklist de Qualidade

Devido a necessidade de um indivíduo externo para aplicar o *checklist* de auditoria de forma imparcial, um membro do suporte foi selecionado e treinado para assumir o papel de analista de qualidade (SQA). O analista de qualidade é o responsável por auditar os processos e produtos de trabalho da organização, identificar pontos de não conformidade aos padrões definidos, atribuir e acompanhar ações para corrigir os desvios identificados e escalonar ações não realizadas dentro do prazo acordado.

A partir da realização das auditorias, "Não Conformidades" (pontos de desvio do processo) passaram a ser identificados. As Não Conformidades eram comunicadas aos responsáveis através do *Redmine*, conforme Figura 12.

Não Conformidade #45082 [Atualizar](#) [Tempo de trabalho](#) [Observar](#) [Copiar](#) [Excluir](#)

Garantia da Qualidade - História #45157: Realizar a auditoria das equipes

### Auditoria #45079: Auditoria de Encerramento da Sprint 14 e Planejamento da Sprint 15

[Item 12] O esforço gasto nas cerimônias de review e retrospectiva foi registrado indevidamente

Adicionado por [Redacted]

Situação: Validado  
 Prioridade: Normal  
 Atribuído para: [Redacted]  
 Categoria: -  
 Versão: Sprint 15  
 PA: PMC

% Terminado: 100%  
 Tempo gasto: -

Descrição [Responder](#)

Não está registrado o esforço gasto por Flávia nas cerimônias de review e retrospectiva da sprint 14.

Subtarefas [Adicionar](#)

Tarefas relacionadas [Adicionar](#)

**Figura 12 - Registro de Não Conformidade**

Com a realização das auditorias foi possível garantir a integridade dos dados armazenados no *Redmine* e iniciar o processo de medição e análise, para extrair informações que pudessem ajudar a entender a situação do setor de desenvolvimento para apoiar o alcance de objetivos organizacionais.

Foi realizada uma reunião com a alta gestão para identificar os objetivos de medição da organização. Os objetivos definidos foram: (i) Aumentar a produtividade e realizar alocação de pontos da Sprint de forma mais efetiva, (ii) Cumprir os prazos estabelecidos, (iii) Evolução do Time, (iv) Evitar desvio de escopo durante o andamento das Sprints, (v) Equipe cumprir com suas horas de trabalho e garantir a veracidade dos indicadores, (vi) Equipe trabalhar mais em novas funcionalidades e menos em Bugs ou Defeitos, (vii) Melhorar a aderência do projeto ao processo. Para cada objetivo de medição foram identificadas as necessidades de medição, conforme Figura 13

Após identificar as necessidades de medição, foi possível identificar quais indicadores seriam necessários para dar a visibilidade para alta gestão e para monitorar se os objetivos de medição estavam sendo atendidos. Conforme Figura 13, os indicadores definidos foram: (i) Custo do Ponto, (ii) Efetividade do Time, (iii) Velocidade do Time, (iv) Instabilidade do Escopo, (v) Apontamentos de Horas, (vi) Esforço por categoria, (vii) Aderência ao Processo.

OBJETIVOS ESTRATÉGICOS	NECESSIDADE DE MEDIÇÃO	INDICADOR
Aumentar a produtividade e realizar alocação de pontos da Sprint de forma mais efetiva.	Mensurar em quanto tempo a equipe desenvolve um ponto.	Custo do Ponto
Cumprir os prazos estabelecidos.	Identificar o andamento do Projeto frente aos marcos estabelecidos.	Efetividade do Time
Evolução do Time	Verificar se o time está evoluindo em relação à quantidade de pontos entregues.	Velocidade do Time
Evitar desvio de escopo durante o andamento das Sprints.	Mensurar as variações de escopo de acordo com tickets Não Planejados, Excluídos e Modificados, do tipo História, Bug ou Defeito.	Instabilidade do Escopo
Equipe cumprir com suas horas de trabalho e garantir a veracidade dos indicadores	Mensurar o esforço gasto pela equipe em relação ao planejado.	Apontamentos de Horas
Equipe trabalhar mais em novas funcionalidades e menos em Bugs ou Defeitos.	Mensurar o esforço da equipe por tipo de ticket.	Esforço por categoria
Melhorar a aderência do projeto ao processo	Mensurar o nível de aderência do projeto em relação ao processo adotado, visando aumentar a maturidade das equipes e garantir o uso do processo.	Aderência ao Processo

**Figura 13 - Necessidades de medição e indicadores relacionados**

Os indicadores passaram a ser coletados e analisados com a periodicidade semanal. Após a realização da auditoria e resolução das não conformidades (momento que é possível garantir a integridade dos dados), os dados eram coletados e os indicadores gerados e analisados, conforme Figuras 14 e 15.



Figura 14 - Indicadores (parte 1)

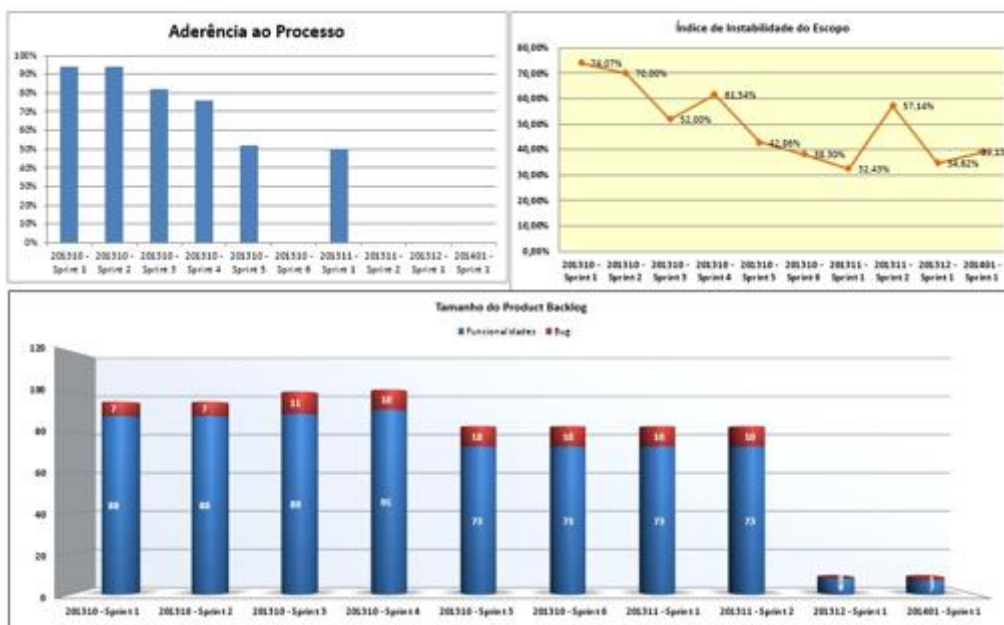


Figura 15 - Indicadores (parte 2)  
Fonte: Autoria própria.

Para os Indicadores fora da meta definida para organização, foram identificadas as causas dos desvios e definidas ações corretivas.

### 4.3.3 Fase 2

As metas definidas para esta fase eram o estabelecimento do conceito de projetos e auditorias para os mesmos. Assim, foi abordado, nesta fase, o grupo de áreas de processos do CMMI formado por PP e PMC. Inicialmente foi ministrado um curso de Gestão de Projetos, apresentando conceitos, práticas e métodos. Foram enfatizados os resultados requeridos, pontos chaves, elementos essenciais, limites de aplicação, formas de aplicação recomendadas, ilustradas através de análise de casos. Após o curso, foram definidos os processos de gestão de projetos. Os mesmos foram validados, implantados e monitorados em projetos pilotos, avaliados e posteriormente corrigidos e adaptados.

Foi definido que um projeto seria composto por um conjunto de quatro Sprints. A estratégia foi fixar o tempo e equipe e variar os parâmetros escopo, risco, qualidade, custo, comunicação. Esta estratégia foi adotada devido a quantidade variável de demandas que podiam surgir para os diferentes produtos, além do fato de que uma única equipe é responsável pela manutenção e evolução de todos os produtos da empresa.

Para gerenciar os projetos foi definido o papel do Gerente de Projetos(GP). O GP é o responsável por definir junto a equipe um planejamento e garantir que o projeto se mantenha alinhado ao mesmo, identificando riscos e definindo ações para garantir o sucesso do projeto.

O ciclo de vida do projeto foi dividido nas seguintes fases: (i) planejamento, (ii) execução e, (iii) encerramento, conforme Figura 16.



Figura 16 - Ciclo de Vida do Projeto

Na fase (i) o Gerente de Projeto realiza o planejamento do projeto, levando em conta os seguintes parâmetros: equipe, esforço, treinamentos, tamanho, cronograma,

riscos, escopo, custo, orçamento, gerência de configuração, recursos e comunicação. O plano de projeto (Figura 17) é apresentado a equipe buscando o comprometimento de todos os envolvidos com o planejado.

### Plano do Projeto

#### Processo

Este projeto herda todas as definições do Plano Padrão de Projeto (versão 1) e está de acordo com a Política de Desenvolvimento em sua versão 1.

#### Equipe e Esforço

Considerando a duração do projeto baseada em 04 sprints semanais, foram realizadas as seguintes estimativas:

Membro	Papel	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Previsto no Projeto
	Product Owner	5h	5h	5h	5h	20h
	Scrum Master	8h	8h	8h	8h	32h
	Time	40h	40h	40h	24h	144h
	Time	40h	40h	40h	24h	144h
	Time	30h	30h	30h	18h	108h
	Time	40h	40h	40h	24h	144h
	Time	8h	40h	40h	24h	92h
	SQA	4h	4h	4h	4h	16h
<b>Total</b>	-	<b>175h</b>	<b>207h</b>	<b>207h</b>	<b>131h</b>	<b>720h</b>

Esforço calculado considerando as horas de gerência, desenvolvimento, pesquisa, manutenção, suporte, auditoria, cerimônias.

#### Treinamentos

Treinamento	Carga Horária	Data Início	Data Fim
Selenium	20 horas	03/01	07/01

#### Tamanho

O cálculo do tamanho estimado no projeto é feito com base em 80% do esforço total da equipe disponível para o projeto dividido pelo custo médio do pontos (em horas). A velocidade estimada na sprint é obtida pela divisão do tamanho estimado do projeto pela quantidade de sprints.

**Figura 17 - Plano de Projeto**

Na fase (ii) são executadas as *Sprints* conforme descrito anteriormente. As cerimônias de fim das *Sprints* passam a ser marcos do projeto, onde este é acompanhado e monitorado. A cada marco do projeto, as medidas coletadas e indicadores analisados, servem como instrumentos para avaliar o desempenho do projeto. Neste momento os riscos são reavaliados e ações são estabelecidas para garantir que desvios do planejamento sejam corrigidos e que o projeto possa atingir seus objetivos. As informações de monitoramento e controle do projeto são registradas no *Redmine*, conforme a Figura 18.

Wiki » Projetos Equipe » Projetos em Andamento » Projeto D201402 » [Editar](#) [Observar](#) [Bloquear](#) [Renomear](#) [Excluir](#) [Histórico](#)

### Acompanhamento da Sprint 13

Parâmetros de Acompanhamento	Status	Causa do Desvio (apenas para alerta ou crítico)	Observação e/ou Ticket
Apointamento de Horas	✓		
Velocidade	⚠	Na Sprint teve uma história que inicialmente o time não tinha conhecimento de como implementar a solução e estimaram em 40 pontos. Porém a solução foi mais simples do que prevista e foi diagnosticado que a história foi super estimada	Não foi definida ação por se tratar de um problema pontual.
Efetividade	✓		
Custo do Ponto	✓		
Esforço por categoria	✓		
Instabilidade do Escopo	✖	Necessidade de apoio ao cliente no uso do sistema de matrícula.	Não foi definida nenhuma ação, pois a instabilidade do escopo alta já estava prevista devido ao período de matrícula das escolas
Aderência ao Processo	⚠	Desconhecimento de algumas práticas do processo e apontamento indevido de horas.	#44500
Comprometimento dos Envolvidos	✓		
Recursos	⚠	O Oracle não estava instalado na máquina de um membro do time	Não foi definida uma ação por ser um problema pontual
Riscos	✓		O risco #44505 ocorreu, porém já tinha sido tomada uma ação no planejamento para minimizar o impacto.
Reuniões Diárias	⚠	Ritmo intenso da equipe.	#44506

Legenda:

OK
  Alerta
  Crítico

Significado dos Status

**Figura 18 - Acompanhamento do Projeto**

Na fase (iii) é realizado o encerramento do projeto. Os resultados do projeto são apresentados a alta direção. Em seguida o projeto é avaliado para verificar se os objetivos iniciais foram atendidos. Ainda nessa fase são levantadas as lições aprendidas do projeto.

Durante esta etapa do projeto de melhoria, as auditorias de qualidade foram incrementadas, de forma que foram definidas auditorias para o planejamento, acompanhamento e encerramento do projeto. A análise de indicadores também passou a consolidar informações a respeito dos projetos, facilitando a definição de uma base histórica de medições para a organização.

As informações de cada projeto são organizadas individualmente no *Redmine*, conforme apresentado na Figura 19.

## Documentação do Projeto

### Planejamento do Projeto

- Plano de Projeto

### Acompanhamento do Projeto

- Acompanhamento da Sprint 13
- Acompanhamento da Sprint 14
- Acompanhamento da Sprint 15
- Acompanhamento da Sprint 16

### Encerramento do Projeto

- Termo de Encerramento

Figura 19 - Documentação do Projeto no *Redmine*

#### 4.3.4 Fase 3

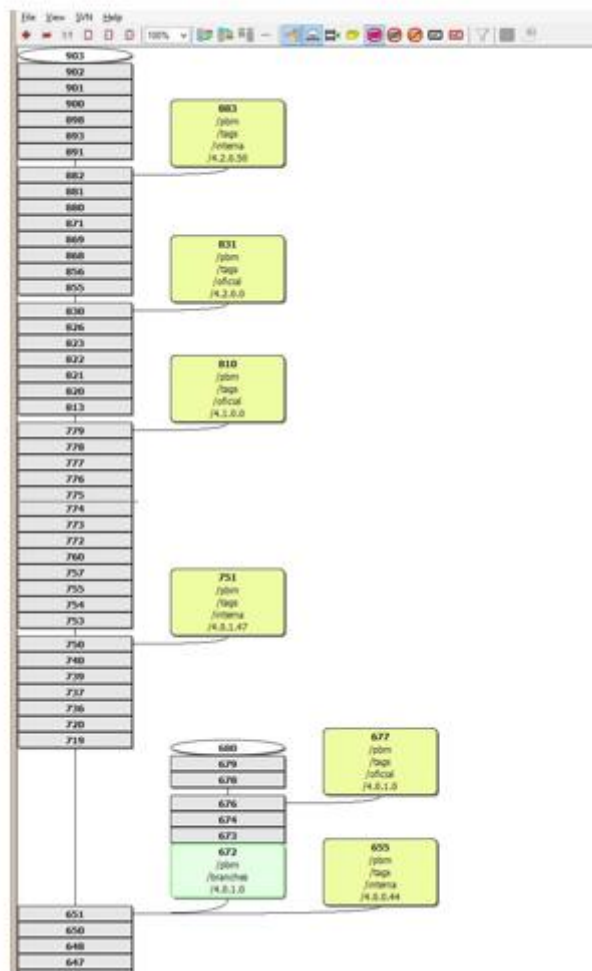
Para esta fase, as principais metas estão relacionadas a definição e institucionalização de práticas relacionadas a gerência de configuração. Inicialmente um curso de gerência de configuração foi ministrado para difundir os conhecimentos e termos básicos relacionados a esta área de processo.

A empresa já adotava o *Subversion* (SVN) como ferramenta de controle de versão. O SVN é uma ferramenta de controle de versão que permite que se trabalhe com diversas versões de arquivos organizados em um diretório e localizados local ou remotamente, mantendo-se suas versões antigas e os registros de quem e quando manipulou os arquivos. É especialmente útil para se controlar versões de um software durante seu desenvolvimento, ou para composição colaborativa de um documento.

A primeira atividade realizada foi reestruturar os repositórios da empresa de forma que cada produto fosse um projeto dentro da estrutura do SVN, e treinar o time para gerenciar o repositório de forma que os registros da ferramenta pudessem ser entendidos de forma clara (Figura 20). Em seguida, foi configurada a integração entre o SVN e o *Redmine*, de modo que as alterações feitas nos produtos de software estejam justificadas em termos de tarefas (estórias e bugs) previamente



acordadas. Assim as mudanças podem ser rastreadas em relação a requisitos e outros fatores de mudança registrados no *Redmine*. Esta rastreabilidade cruzada entre *Redmine* e SVN foi obtida através da padronização de mensagens de comentários durante operações de *commit* (operação de envio de dados ao repositório).



**Figura 20 - Gráfico do Histórico de Revisões do SVN**

Em seguida mecanismos para controlar as configurações estáveis (*baselines*) do produto e as liberações foram estabelecidos através de tickets do *Redmine* ligados somados a funcionalidade de *tags* (rótulos) do SVN. Permitindo o controle da situação dos itens de configuração relacionados ao produto de software (Figura 21).

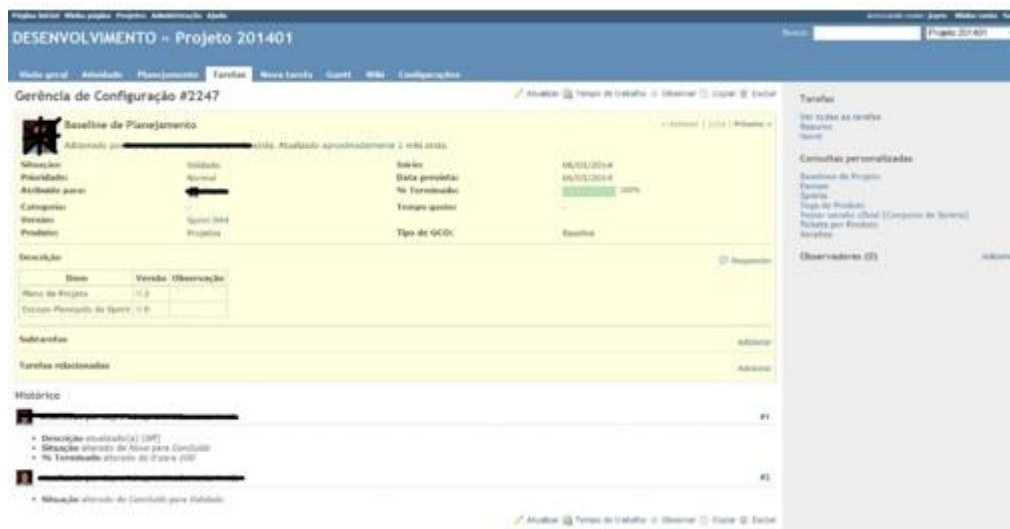
The screenshot shows the Redmine interface for project configuration. The main header is 'DESENVOLVIMENTO - Projeto 201402'. The page title is 'Gerência de Configuração #2672'. The interface is divided into several sections:

- Tag oficial:** Shows project details like 'Atualizado 1 dia atrás', 'Validado', 'Data prevista: 24/02/2014', and 'Terminado: 100%'.
- Descrição:** A text area containing project goals and objectives.
- Subtarefas:** A list of tasks with columns for 'Situação', 'Observação', and 'NE Realizada'.
- Configuração:** A table with columns for 'Id', 'Área', 'Item', 'Situação', 'Observação', and 'NE Realizada'.

Id	Área	Item	Situação	Observação	NE Realizada
1	000	O nome da versão do produto atende ao padrão definido (X.Y.Z.GP)?	OK		
2	000	O escopo para a versão foi definido? (Dados de histórico previous para a versão fixam, adicionadas à descrição da versão)	OK		
3	000	O escopo definido foi finalizado? (Dados de histórico estão com estado "validado", "rejeitado" ou "cancelado")	OK		
4	000	Existe versão planejada no projeto de versão?	OK		
5	000	As tarefas a Tag no 50% correspondem a versão planejada?	OK		
6	000	A Tag definida no 50% está em conformidade com o nome da versão planejada?	OK		
7	000	Para Tags de Versões Oficiais: Foram realizadas as testes sobre o conjunto de funcionalidades desenvolvidas ao longo do tempo?	OK		
8	000	Para Tags de Versões Oficiais: O resultado dos testes realizados sobre o conjunto de funcionalidades desenvolvidas ao longo do tempo foi satisfatório?	OK		
9	000	Para Tags de Versões Oficiais: Caso o resultado dos testes não tenha sido satisfatório, foram criadas tickets de bugs, como solicitado, para registrar as pendências identificadas?	Não se aplica		

Figura 21 - Baseline de Produto

*Baselines* também foram institucionalizadas em relação ao conceito de projetos mensais estabelecido. Assim *baselines* são definidas durante os marcos de planejamento e encerramento dos projetos. Mudanças em relação ao planejamento (plano de projeto e escopo) são controladas através do *redmine* (Figura 22).



**Figura 22 - Baseline de Projeto**

Os padrões definidos para a gestão de configuração também foram incluídos nas auditorias de qualidade e procedimentos foram definidos para a validação das *baselines*, garantindo que estejam íntegras e consistentes.

#### **4.3.5 Fase 4**

Para esta fase, as principais metas definidas estavam relacionadas a definição dos planos e políticas referentes às práticas institucionalizadas até então. Isto refletiu na concepção dos documentos (Figura 23): (i) Política Organizacional, (ii) Processo de Desenvolvimento, (iii) Plano de Garantia da Qualidade, (iv) Guia de Medição e Análise, (v) Plano de Gestão de Configuração, (vi) Papéis e Responsabilidades.

## Documentos Organizacionais

- Política de Desenvolvimento
- Processo de Desenvolvimento
- Papéis

## Artefatos Organizacionais

- Plano Padrão de Projeto
- Guia de Medição
- Plano Padrão de Gerência de Configuração
- Plano de Garantia da Qualidade

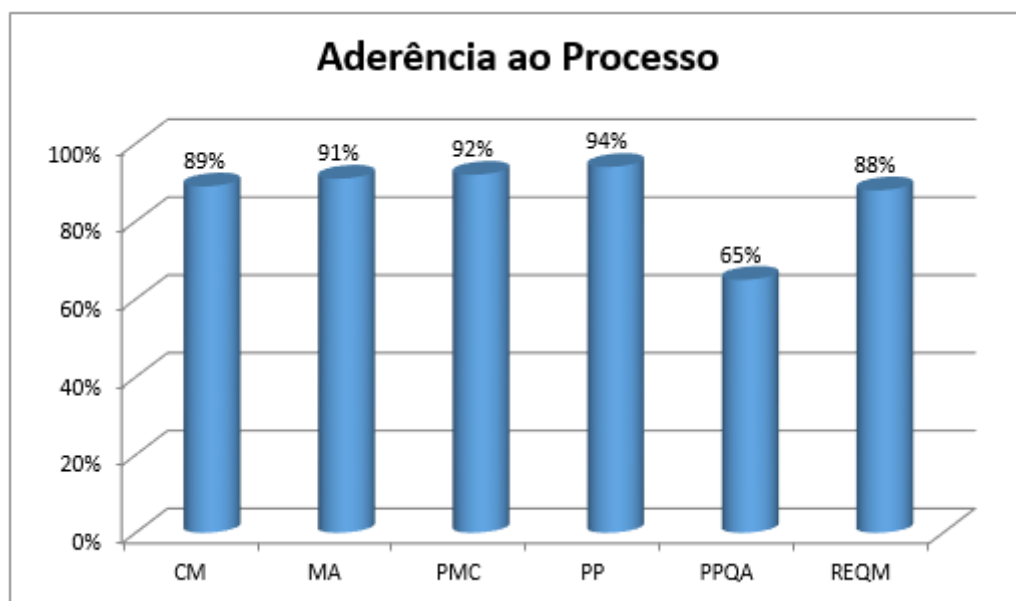
**Figura 23 - Documentos Organizacionais mantidos no Redmine**

O documento (i) descreve todas as premissas da organização em relação a seus projetos e processos. O (ii) descreve o processo de desenvolvimento, bem como o ciclo de vida e fases dos projetos. O (iii) descreve o processo de garantia da qualidade, realização de auditorias, comunicação de não conformidades, plano de escalonamento de não conformidades. O (iv) tem por objetivo definir o conjunto de indicadores e medições a serem coletados e analisados para os projetos. O (v) é apresentado um conjunto de diretrizes a serem seguidas nos projetos de software no contexto da Gerência de Configuração. O (vi) documenta todos os papéis da organização, bem como suas responsabilidades e qualificações necessárias.

### **4.3.6 Fase 5**

Esta fase teve como meta identificar o grau de institucionalização do processo e estabelecer um mecanismo para garantir que a institucionalização seja mantida.

Para isto, um novo *Gap Analysis* foi conduzido para identificar o nível de atendimento das práticas institucionalizadas na empresa em relação aos requisitos do Nível 2 de maturidade do CMMI. Os resultados são observados na Figura 24.



**Figura 24 - Resultado do segundo Gap Analysis Final**

Os resultados apontaram que havia evidências para todas as Práticas Específicas do Nível 2 do CMMI-DEV, no entanto algumas falhas na institucionalização de práticas, sobretudo em relação a garantia da qualidade. Um plano de ação foi definido para sanar determinados problemas e auditorias externas mensais foram estabelecidas para garantir que a institucionalização do processo fosse mantida e averiguar se as auditorias de qualidade da empresa são realizadas e se as Não Conformidades geradas estavam de fato sendo contempladas.

Com implementação da auditoria externa, após dois projetos foi diagnosticada a aderência de 100% das práticas da empresa em relação as exigências do nível 2 do CMMI.

#### **4.4 Mapeamento dos resultados obtidos com o CMMI-DEV**

Afim de avaliar os resultados obtidos, os quadros 1, 2, 3, 4, 5 e 6 apresentam uma análise do grau de atendimento das Práticas Específicas do Nível 2 de maturidade do CMMI-DEV, para cada PA, em relação às evidências geradas na implantação da melhoria de processo na empresa apresentada. O Quadro 5.1 apresenta as Áreas de Processo (PA) do Nível 2 do CMMI-DEV com suas

respectivas Práticas Específicas (SP), as evidências produzidas com base nos ativos propostos e o grau de atendimento das práticas.

A análise do grau de atendimento aponta se as evidências geradas são suficientes para indicar o atendimento de uma prática específica. Os valores possíveis são: "Atende totalmente" (as evidências são suficientes), "atende parcialmente" (as evidências por si só não são suficientes para atender a prática específica, sendo necessárias evidências complementares) e "não atende" (a proposta não gerou evidência alguma que possa ser usada para indicar o atendimento de uma prática específica).

**Quadro 1 - Grau de atendimento das evidências geradas às práticas específicas da PA Project Planning**

<b>SP</b>	<b>Evidência Gerada</b>	<b>Grau de Atendimento</b>
SP 1.1 Estimar o escopo do projeto.	Planejamento do Projeto – Plano de Projeto – Planejamento do Escopo; Cerimônia de Planejamento de Sprint; <i>Backlog</i>	Atende Totalmente
SP 1.2 Estabelecer estimativas de produto de trabalho e atributos de tarefas.	Planejamento do Projeto - Plano de Projeto – Planejamento do Tamanho do Projeto; Cerimônia de Planejamento de Sprint.	Atende Totalmente
SP 1.3 Definir as fases do ciclo de vida.	Planejamento do Projeto – Plano de Projeto - Planejamento do Cronograma	Atende Totalmente
SP 1.4 Estimar esforço e custo	Planejamento do Projeto - Plano de Projeto – Planejamento do Esforço	Atende Totalmente
SP 2.1 Estabelecer o Orçamento e Cronograma	Planejamento do Projeto - Plano de Projeto – Planejamento do Cronograma	Atende Totalmente
SP 2.2 Identificar os Riscos do Projeto	Planejamento do Projeto – Plano de Projeto – Planejamento dos Riscos; Tickets do tipo Risco no <i>Redmine</i> .	Atende Totalmente
SP 2.3 Plano de	Planejamento do Projeto – Plano de	Atende

Gerenciamento de Dados	Projeto – Planejamento da Gerência de Configuração; Plano de Gerência de Configuração.	Totalmente
SP 2.4 Planejar os Recursos do Projeto	Planejamento do Projeto – Plano de Projeto – Planejamento da Gerência de Configuração; Plano de Gerência de Configuração.	Atende Totalmente
SP 2.5 Plano de conhecimentos e habilidades necessários.	Planejamento do Projeto - Plano de Projeto – Planejamento da Equipe e Treinamentos; Documento de Papeis e Responsabilidades.	Atende Totalmente
SP 2.6 Plano de Envolvimento das Partes Interessadas	Planejamento do Projeto - Plano de Projeto – Planejamento da Comunicação	Atende Totalmente
SP 2.7 Estabelecer o plano do projeto	Planejamento do Projeto - Plano de Projeto	Atende Totalmente
SP 3.1 Rever os planos que afetam o projeto.	Planejamento do Projeto – Apresentação do Plano de Projeto para Equipe	Atende Totalmente
SP 3.2 Conciliar o Trabalho e os Níveis de Recursos	Planejamento do Projeto – Apresentação do Plano de Projeto para Equipe	Atende Totalmente
SP 3.3 Obter Plano de Compromisso	Planejamento do Projeto – Apresentação do Plano de Projeto para Equipe	Atende Totalmente

**Quadro 2 - Grau de atendimento das evidências geradas às práticas específicas da PA Project Monitoring and Control**

<b>SP</b>	<b>Evidência Gerada</b>	<b>Grau de Atendimento</b>
-----------	-------------------------	----------------------------

SP 1.1 Monitorar parâmetros de planejamento de trabalho.	Execução do Projeto – Análise de Marco – Monitoramento dos Indicadores	Atende Totalmente
SP 1.2 Monitorar os compromissos.	Execução do Projeto – Análise de Marco – Monitoramento dos Compromissos dos Envolvidos	Atende Totalmente
SP 1.3 Monitorar os riscos.	Execução do Projeto – Análise de Marco – Monitoramento dos Riscos	Atende Totalmente
SP 1.4 Monitorar a Gestão de Dados	Execução do Projeto – Análise de Marco – Monitoramento dos Recursos	Atende Totalmente
SP 1.5 Monitorar de Participação das Partes Interessadas	Execução do Projeto – Análise de Marco – Monitoramento dos Compromissos com os Envolvidos	Atende Totalmente
SP 1.6 Conduzir Revisões de Progresso	Execução do Projeto – Análise de Marco; Cerimônia de Revisão de Sprint.	Atende Totalmente
SP 1.7 Conduzir Revisões de Marcos	Execução do Projeto – Análise de Marco	Atende Totalmente
SP 2.1 Analisar Questões	Execução do Projeto – Análise de Marco – Análise de Causa dos Desvios; Cerimônias de Retrospectiva.	Atende Totalmente
SP 2.2 Tomar uma Ação Corretiva	Execução do Projeto – Análise de Marco – Análise de Causa dos Desvios	Atende Totalmente
SP 2.3 Gerenciar Ações Corretivas	Execução do Projeto – Análise de Marco – Ações Corretivas; Tickets do Redmine do Tipo Ação.	Atende Totalmente

**Quadro 3 - Grau de atendimento das evidências geradas às práticas específicas da PA Requirement Management**



<b>SP</b>	<b>Evidência Gerada</b>	<b>Grau de Atendimento</b>
SP 1.1 Compreender os Requisitos	Execução do Projeto – Cerimônia de Planejamento de Sprint; Backlog	Atende Totalmente
SP 1.2 Obter Compromisso de Requisitos	Execução do Projeto – Cerimônia de Planejamento de Sprint	Atende Totalmente
SP 1.3 Gerenciar Mudanças nos Requisitos	Gerência de Configuração – Controle de Mudanças.	Atende Totalmente
SP 1.4 Manter Rastreabilidade Bidirecional dos Requisitos	Gerência de Configuração – Rastreabilidade Tickets do <i>Redmine</i> com Código no <i>Subversion</i>	Atende Totalmente
SP 1.5 Garantir o Alinhamento Entre o Trabalho do Projeto e Requisitos	Execução do Projeto – Cerimônia de Revisão de Sprint.	Atende Totalmente

**Quadro 4 - Grau de atendimento das evidências geradas às práticas específicas da PA Measurement and Analysis**

<b>SP</b>	<b>Evidência Gerada</b>	<b>Grau de Atendimento</b>
SP 1.1 Estabelecer Objetivos de Medição	Guia de Medição – Objetivos de Medição.	Atende Totalmente
SP 1.2 Especificar Medições	Guia de Medição – Definição dos Indicadores	Atende Totalmente
SP 1.3 Especifique a coleta de dados e procedimentos de armazenamento	Guia de Medição – Especificação dos Indicadores	Atende Totalmente
SP 1.4 Especificar Procedimentos de Análise	Guia de Medição – Especificação dos Indicadores	Atende Totalmente

SP 2.1 Obter Dados de Medição	Execução do Projeto – Análise de Marco – Coleta dos Indicadores	Atende Totalmente
SP 2.2 Analisar os Dados de Medição	Execução do Projeto – Análise de Marco – Analise dos Indicadores	Atende Totalmente
SP 2.3 Armazenamento de Dados e Resultados	Execução do Projeto – Análise de Marco – Planilha de Indicadores Armazenada no <i>Redmine</i>	Atende Totalmente
SP 2.4 Comunicar os Resultados	Execução do Projeto – Análise de Marco – Comunicação do Resultado do Acompanhamento	Atende Totalmente

**Quadro 5 - Grau de atendimento das evidências geradas às práticas específicas da PA Configuration Management**

<b>SP</b>	<b>Evidência Gerada</b>	<b>Grau de Atendimento</b>
SP 1.1 Avaliar Objetivamente Processos.	Execução do Projeto – Auditoria de Qualidade	Atende Totalmente
SP 1.2 Avaliar Objetivamente Produtos de Trabalho	Execução do Projeto – Auditoria de Qualidade	Atende Totalmente
SP 2.1 Comunicar e Resolver Problemas de Não Conformidade	Execução do Projeto – Auditoria de Qualidade – Comunicação das Não Conformidades	Atende Totalmente
SP 2.2 Estabelecer Registros	Execução do Projeto – Auditoria de Qualidade – Tickets de Auditorias no <i>Redmine</i> / Tickets de Não Conformidades	Atende Totalmente

**Quadro 6 - Grau de atendimento das evidências geradas às práticas específicas da PA Product and Process Quality Assurance**

<b>SP</b>	<b>Evidência Gerada</b>	<b>Grau de Atendimento</b>
SP 1.1 Identificar Itens de Configuração	Planejamento do Projeto – Planejamento da Gerência de Configuração; Plano de Gerência de Configuração.	Atende Totalmente
SP 1.2 Estabelecer um Sistema de Gerência de configuração	Planejamento do Projeto – Planejamento da Gerência de Configuração; Plano de Gerência de Configuração; Ferramenta <i>Redmine</i> ; Ferramenta <i>Subversion</i> .	Atende Totalmente
SP 1.3 Criar ou liberar baselines	Planejamento do Projeto – Criação de <i>Baseline</i> de Planejamento; Encerramento do Projeto – <i>Baseline</i> de Encerramento do Projeto.	Atende Totalmente
SP 2.1 Acompanhar as solicitações de mudança.	Gerência de Configuração – Controle de Mudanças; Tickets do Tipo Ação no Redmine.	Atende Totalmente
SP 2.2 Controlar a Configuração de Itens	<i>Baselines</i> ; <i>Redmine</i> ; <i>Subversion</i>	Atende Totalmente
SP 3.1 Estabelecer Gerenciamento de Registros de Configuração	Log do <i>Redmine</i> ; Log do SVN.	Atende Totalmente
SP 3.2 Realizar Auditorias de	Execução do Projeto – Auditoria da Qualidade	Atende Totalmente

Configuração		
--------------	--	--

A análise dos quadros revela que as evidências geradas nesta abordagem, são suficientes para o atendimento do Nível 2 do CMMI-DEV.

#### 4.5 Benefícios observados para a empresa

A execução do projeto de melhoria de processo descrita neste trabalho, promoveu um conjunto de benefícios na empresa alvo. Em relação aos principais problemas inicialmente identificados (Seção 4.2), as seguintes melhorias puderam ser observadas:

- Necessidade de organização do trabalho: Com a implantação dos projetos, houve maior visibilidade das ações e a possibilidade de planejar as ações, inicialmente em unidades de tempo semanais (*sprint*) e posteriormente mensal (projeto);
- Definir um escopo e garantir o foco das atividades: Ao garantir a identificação de um *backlog* de atividades e a priorização deste ao longo das *sprints*, foi possível estabilizar o escopo de atividades, permitindo que o time se concentrasse nas atividades prioritárias;
- Insatisfação do cliente: Com a implantação das *Sprints*, criou-se a cultura de entregar software funcionando constantemente, agilizando a evolução dos produtos. Os bugs reportados pelos clientes passaram a ser analisados e priorizados, o que ajudou o time a dar vazão aos bugs realmente prioritários.
- Interferências externas: A entrada de novos requisitos nas sprints/projetos passou a ser controlada, assim as mudanças passaram a ter seu impacto identificado e controlado; Ao institucionalizar o papel de *Scrum Master*, a equipe passou a ter maior proteção a interferências externas, e estas passaram a ser mais visíveis para a alta gestão a partir da análise do indicador de instabilidade do escopo, assim ações puderam ser tomadas para garantir maior estabilidade.

Com a implantação dos indicadores foi possível medir a evolução da performance ao longo do programa de melhoria. A produtividade do time passou de 50 pontos por Sprint para 90 pontos, o que representou um aumento de 80% na produtividade. O

índice de retrabalho, esforço gasto para corrigir bugs, passou de 70% para 15%. A instabilidade do escopo diminuiu cerca de 30%.

Esses resultados mostram que o programa de melhoria não só garantiu a aderência as exigências do nível 2 do CMMI, mas garantiu a resolução de problemas que eram críticos para organização, conseqüentemente melhorando a performance e qualidades dos processos e produtos.

## 5 CONCLUSÃO

Ao longo deste trabalho pretendeu-se, sobretudo, focar o processo de desenvolvimento de software, dando destaque à necessidade da sua qualidade como fator determinante para a obtenção de melhores resultados, do próprio sucesso das organizações e da otimização das atividades inerentes a todo o processo de desenvolvimento.

No caso específico de processos de desenvolvimento de software, a melhoria da qualidade pode ser obtida através da implementação de Modelos de Maturidade, modelos esses que permitem à organização avaliar os seus processos de desenvolvimento e manutenção, implementar melhorias no seu modo de funcionamento e determinar os progressos obtidos.

No caso da empresa estudada, após a constatação de que existiam alguns problemas que vinham ameaçando a qualidade do software como a necessidade de organização do trabalho, falta de visibilidade, insatisfação dos clientes, atrasos na liberação de versões, muitas interferências externas, entre outros, resolveu avançar para a resolução destes mesmos problemas e também para a melhoria global dos processos e atividades da empresa através da adoção do modelo de maturidade CMMI, para guiar a melhoria do seu processo de desenvolvimento. O objetivo passou então pela certificação em nível 2 de CMMI, sendo necessário para isso a definição e implementação das áreas de processo correspondentes a esse nível.

Este trabalho permitiu demonstrar todo o caminho que a empresa passou para definir e implementar todas as mudanças na sua organização. Foram apresentados os processos, estratégias e ações necessárias para implementar o modelo e conseguir a aderência a exigência do nível 2 de maturidade do CMMI, consequentemente resolvendo todos os problemas que existiam na empresa.

Além de todos os ganhos alcançados pela empresa em estudo, o principal benefício deste trabalho diz respeito à validação da metodologia de implantação aplicada neste estudo de caso. Validação essa que permite que o método seja consolidado e disponibilizado, para que pequenas empresas, que não possuem recursos financeiros para contratar uma consultoria, consigam implantar um programa de melhoria de processos para melhorar a qualidade dos seus produtos.

## REFERÊNCIAS

- [ABNT 1994] Associação Brasileira de Normas Técnicas. NBR ISO 8402/1994 - Gestão da qualidade e garantia da qualidade - Terminologia. Rio de Janeiro: ABNT, 1994.
- [AGUIAR, 2004] AGUIAR, H. V. PEPPE – Processo de Implantação de Pequenas Empresas. Monografia apresenta a UFLA, 2004.
- [ANDRADE 2002] Andrade, P. I. Qualidade nos Processos do Ciclo de Vida do Produto com CMMI: Uma Aplicação Prática de Gerência de Configuração na COMPSIS. Dez. 2002.
- [CÂNDIDO 2004] Cândido, Edílson J. D. Uma simplificação da técnica análise de pontos de função para estimar tamanho de aplicativos web. Dissertação de Mestrado, USP, 2004.
- [CMMI 2014] *Capability Maturity Model Integration. Software Engineering Institute(SEI)*. Disponível em <http://www.sei.cmu.edu/cmmi/> . Acesso em: 05/01/2014.
- [HUMPHREY 1989] Humphrey W. S. *Managing the software Process*. Addison-Wesley, 1989.
- [IDEAL 2014] *The IDEAL<sup>SM</sup> Model. Software Engineering Institute(SEI)*. Disponível em <http://www.sei.cmu.edu/ideal/> . Acesso em 05/01/2014.
- [IEEE 1990] *Institute of Electrical and Electronics and Engineers*. IEEE Std 610.12-1990 - *Standard glossary of software engineering terminology*. Piscataway: IEEE, 1990.
- [LIBERATO, 2008] Liberato, T. E. M., Implementação do Modelo CMMI na Espírito Santo Informática. Dissertação de Mestrado, UTM, 2008.
- [PAULK 1995] Paulk, M.C.; Weber, C.V.; Curtis, B.; Chrissis, M.B.; E Outros. *The Capability Maturity Model: Guidelines for Improving the Software Process*. Estados Unidos: Addison-Wesley. 1995.
- [PRESSMAN 2002] R. S. Pressman. Engenharia de Software, 5ª ed. Rio de Janeiro, *Mc Graw Hill*, 2002.
- [RAMOS, 2011] RAMOS, E. S., Implantação do Modelo MPS.BR: Estudo de Caso da Empresa CIENTEC – Consultoria e Desenvolvimento de Sistemas. Trabalho de Conclusão de Curso, FUMEC, 2011.
- [Redmine 2014] Disponível em [www.redmine.org](http://www.redmine.org). Acesso 02/02/2014
- [ROUILLER 2001] Rouiller, A. C. Gerenciamento de Projetos de Software para empresas de Pequeno Porte. Tese de Doutorado, UFPE, 2001.

[SALVIANO 2003] Clenio F. Salviano. Melhoria e Avaliação de Processo com ISO/IEC 15504 (SPICE) e CMMI, 1ª ed, Editora UFLA, 2003.

[SCHWABER, 2013] SCHWABER, K.; SUTHERLAND, J, Guia do Scrum. Disponível em <https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide-Portuguese-BR.pdf>. Acesso em 01/02/2014.