



**FUNDAÇÃO EDSON QUEIROZ
UNIVERSIDADE DE FORTALEZA
CENTRO DE CIÊNCIAS TECNOLÓGICAS
MESTRADO EM INFORMÁTICA APLICADA**



Carla Ilane Moreira Bezerra

**MiniDMAIC: Uma Abordagem para Análise e Resolução de
Causas de Problemas em Projetos de Desenvolvimento de
Software**

Fortaleza
2009

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.



**FUNDAÇÃO EDSON QUEIROZ
UNIVERSIDADE DE FORTALEZA
CENTRO DE CIÊNCIAS TECNOLÓGICAS
MESTRADO EM INFORMÁTICA APLICADA**



Carla Ilane Moreira Bezerra

MiniDMAIC: Uma Abordagem para Análise e Resolução de Causas de Problemas em Projetos de Desenvolvimento de Software

Dissertação apresentada ao Curso de Mestrado em Informática Aplicada da Universidade de Fortaleza (UNIFOR), como requisito parcial para obtenção do Título de Mestre em Informática Aplicada.

Orientador: Prof. Adriano Bessa Albuquerque, D.Sc.

Fortaleza
2009


-
- B574m Bezerra, Carla Ilane Moreira.
MiniDMAIC : uma abordagem para análise e resolução de causas de
problemas em projetos de desenvolvimento de software / Carla Ilane Moreira
Bezerra. - 2009.
134 f.
- Dissertação (mestrado) – Universidade de Fortaleza, 2009.
“Orientação: Prof. Adriano Bessa Albuquerque.”
1. Desenvolvimento de software. 2. Resolução de problemas – Análise.
3. Controle de qualidade . I. Título.
- CDU 681.3.06
-


Carla Ilane Moreira Bezerra

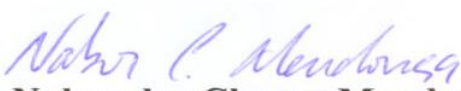
**MiniDMAIC: Uma Abordagem para Análise e Resolução de
Causas de Problemas em Projetos de Desenvolvimento de
Software**

Data de Aprovação: 15/12/2009

Banca Examinadora:


Prof. Adriano Bessa Albuquerque, D. Sc.
(Prof. Orientador – UNIFOR)


Prof. Alexandre Marcos Lins de Vasconcelos, Ph. D.
(Membro - UFPE)


Prof. Nabor das Chagas Mendonça, Ph. D.
(Membro - UNIFOR)

**Dedico este trabalho a
milha família e amigos.**

AGRADECIMENTOS

Agradeço primeiramente a Deus, por ter sempre guiado minha vida para bons caminhos, dando-me saúde para conseguir alcançar todos os objetivos que eu almejei.

Ao meu marido Emanuel, pelo amor, apoio e companheirismo, sempre incentivando e possibilitando atingir meus objetivos. E ao Lucas por ter proporcionado tanta alegria à nossa família.

Aos meus pais, Luzia e Carlos, por acreditarem em mim, possibilitando as melhores opções de estudos e formando a pessoa que sou hoje, e aos meus irmãos, Luciana e Rildo, pela amizade e amor que me proporcionam, mesmo quando estamos distantes.

Ao Professor Adriano Bessa, por ter me recebido como sua orientanda com tanto carinho e dedicação, e ter me apoiado e incentivado a continuação e melhoria do trabalho. Ao Professor Plácido por ter sempre acreditado em mim como aluna, desde os tempos da graduação na UECE. Ao Professor Nabor, pelo enorme incentivo e contribuição no mestrado, tendo me acolhido como orientanda no momento em que fiquei sem orientador, e agora também participando da minha banca. E ao Professor Alexandre Vasconcelos, por ter aceitado o convite de participar da minha banca, sua presença na defesa do meu trabalho é de grande importância pelo grande profissional que é e por conhecer parte da minha vida acadêmica pelos congressos a fora.

Aos meus amigos, Ciro e Márcia, pela fundamental contribuição na elaboração do trabalho.

Ao Atlântico, minha segunda casa, e aos meus queridos chefes Carlo Giovano e Gabriela Telles, pelo grande apoio e pela oportunidade da elaboração e aplicação do trabalho no Atlântico. Também, aos meus amigos do Atlântico: Luciana, Paula, Sérgio, Sofia, Joice, Cícilia e Thaís.

À FUNCAP, pelo apoio financeiro durante boa parte do meu mestrado.

Ao professor e amigo Arnaldo Dias Belchior (*in memoriam*), por ter me dado a oportunidade de ser sua orientanda e me possibilitado uma grande experiência acadêmica e profissional. Hoje professor, fecha-se o ciclo dos seus orientados, seja onde quer que esteja, espero que esteja orgulhoso dos seus queridos alunos.

Resumo da dissertação apresentada ao Corpo Docente do Curso de Mestrado em Informática Aplicada da Universidade de Fortaleza, como parte dos requisitos necessários para a obtenção do grau de Mestre em Informática Aplicada.

MiniDMAIC: Uma Abordagem para Análise e Resolução de Causas de Problemas em Projetos de Desenvolvimento de Software

Autor: Carla Ilane Moreira Bezerra

Orientador: Adriano Bessa Albuquerque, D.Sc.

O tratamento dos problemas e defeitos encontrados nos projetos de software ainda é deficiente na maioria das organizações. As análises em geral não focam suficientemente no problema e suas possíveis origens, levando a tomadas de decisões erradas que acabam por não resolver o problema. Também é difícil a implementação de um processo de Análise e Resolução de Causas (CAR) em projetos, conforme prescrito pelo nível 5 do CMMI, devido aos recursos limitados com os quais os mesmos têm que trabalhar. Este trabalho propõe o MiniDMAIC, uma abordagem para análise e resolução de causas de defeitos e problemas em projetos de desenvolvimento de software com base no método DMAIC do *Six Sigma* e na área de processo de CAR do CMMI. O MiniDMAIC é uma estratégia que visa a simplificação do modelo DMAIC com o intuito de tratar as causas e resolução de problemas em projetos de desenvolvimento de software de forma mais prática, rápida, com menos riscos e custos, prevenindo recorrências futuras e implantando melhorias no processo de desenvolvimento. A abordagem definida neste trabalho foi aplicada em um projeto real de desenvolvimento de software em uma empresa brasileira de pesquisa e desenvolvimento que foi avaliada nível 5 do CMMI obtendo o cumprimento de todas as práticas do CMMI com auxílio da abordagem MiniDMAIC para obter a aderência de práticas da área de processo de Análise e Resolução de Causas.

Palavras-chave: Análise e Resolução de Causas, CMMI, *Six Sigma*, DMAIC.

Abstract of the dissertation presented to the board of faculties of the Master Program in Applied Informatics at the University of Fortaleza, as partial fulfillment of the requirements for the Master's degree in Applied Informatics.

MiniDMAIC: An Approach for Causal Analysis and Resolution in Software Development Projects

Autor: Carla Ilane Moreira Bezerra

Orientador: Adriano Bessa Albuquerque, D.Sc.

Dealing with problems and defects on software projects is still deficient on the majority of the organizations. The analysis, in general, does not focus sufficiently on the problems and their possible origins, leading to wrong decision making that do not solve the problems. It is also difficult to implement a Causal Analysis and Resolution process on projects, as established on CMMI level 5, due to scarce resources. This work proposes the MiniDMAIC, an approach to analysis and resolution of causes of defects and problems in software development projects, based on the DMAIC method from *Six Sigma* and on the process area CAR of the CMMI. The MiniDMAIC is an strategy that aims at the simplification of the DMAIC model to deal with the causes and problem resolutions in software development projects in a more practical and rapid, way with less risks and costs, preventing future recurrences, implementing improvements on the development process. The approach defined on this work was executed on a real project in a research and development Brazilian organization, that was successfully evaluated on CMMI level 5. The MiniDMAIC helped the organization to obtain the adherence to all practices of the process area Causal Analysis and Resolution.

Keywords: Causal Analysis and Resolution, CMMI, *Six Sigma*, DMAIC.

SUMÁRIO

LISTA DE FIGURAS	xv
LISTA DE TABELAS	xvii
1 INTRODUÇÃO.....	1
1.1 MOTIVAÇÃO	1
1.2 OBJETIVOS	3
1.3 METODOLOGIA	4
1.4 ESTRUTURA DO TRABALHO	5
2 ANÁLISE E RESOLUÇÃO DE CAUSAS	6
2.1 VISÃO GERAL	6
2.2 CONCEITO DE PROBLEMA E DEFEITO	7
2.3 CLASSIFICAÇÃO DE DEFEITOS	8
2.4 ANÁLISE E RESOLUÇÃO DE CAUSAS	9
2.5 FERRAMENTAS UTILIZADAS NA ANÁLISE DE CAUSAS	14
2.6 ANÁLISE E RESOLUÇÃO DE CAUSAS NO MR-MPS	17
2.7 ANÁLISE E RESOLUÇÃO DE CAUSAS NO CMMI.....	19
2.7.1 Nível 5 de Maturidade.....	21
2.7.2 Área de Processo: Análise e Resolução de Causas	23
2.8 CONCLUSÃO DO CAPÍTULO	26

3	O <i>SIX SIGMA</i> E O MÉTODO DMAIC.....	27
3.1	VISÃO GERAL.....	27
3.2	HISTÓRIA DO <i>SIX SIGMA</i>	28
3.3	O QUE É O <i>SIX SIGMA</i> ?.....	29
3.3.1	Papéis e Responsabilidades.....	33
3.3.2	Ferramentas e Metodologias	34
3.4	O MÉTODO DMAIC	36
3.4.1	Definir	39
3.4.2	Medir	43
3.4.3	Analisar	47
3.4.4	Melhorar	50
3.4.5	Controlar.....	51
3.5	DMAIC X CMMI	54
3.6	CONCLUSÃO DO CAPÍTULO	56
4	ABORDAGEM MINIDMAIC	57
4.1	INTRODUÇÃO.....	57
4.2	ABORDAGEM DEFINIDA – MINIDMAIC.....	58
4.2.1	Fase: Definir.....	62
4.2.2	Fase: Medir.....	65
4.2.3	Fase: Analisar.....	66
4.2.4	Fase: Melhorar.....	67
4.2.5	Fase: Controlar	69

4.2.6	Disponibilizar as Oportunidades de Melhoria para os Ativos Organizacionais	70
4.3	DMAIC X MINIDMAIC	71
4.4	CAR X MINIDMAIC	73
4.5	CONCLUSÃO DO CAPÍTULO	75
5	EXPERIÊNCIA DE USO	76
5.1	INTRODUÇÃO	76
5.2	IMPLEMENTAÇÃO DO MINIDMAIC NO JIRA	77
5.3	EXPERIÊNCIA DE USO DO MINIDMAIC	77
5.3.1	Caracterização do MiniDMAIC para organização	77
5.3.2	Caracterização do Projeto Piloto	80
5.3.3	Executando as Fases “Definir” e “Medir” do MiniDMAIC	81
5.3.4	Executando Fase “Analisar” do MiniDMAIC	84
5.3.5	Executando a Fase “Melhorar” do MiniDMAIC	86
5.3.6	Executando a Fase “Controlar” do MiniDMAIC	88
5.3.7	Disponibilizar as Oportunidades de Melhoria para os Ativos Organizacionais	90
5.3.8	Lições Aprendidas e Oportunidades de Melhoria	91
5.4	CONCLUSÃO DO CAPÍTULO	93
6	CONCLUSÃO E TRABALHOS FUTUROS	94
6.1	CONSIDERAÇÕES FINAIS	94
6.2	PRINCIPAIS CONTRIBUIÇÕES	95
6.3	LIMITAÇÕES	95

6.4	TRABALHOS FUTUROS	96
	BIBLIOGRAFIA	97
	APÊNDICE A – PASSOS DO MINIDMAIC NA FERRAMENTA JIRA	107
	APÊNDICE B – TEMPLATE ANÁLISE DE CAUSAS.....	114

LISTA DE FIGURAS

Figura 2.1: Exemplo de Diagrama de Causa e Efeito (ROBITALLE, 2004).	16
Figura 2.2: Exemplo de Gráfico de Pareto.....	16
Figura 2.3: Níveis de Maturidade do Modelo MR-MPS (SOFTEX, 2009a).....	17
Figura 2.4: Níveis de Maturidade da Representação por Estágios do CMMI.	21
Figura 2.5: Diagrama de Contexto da Análise e Resolução de Causas (AHERN et al., 2003).....	23
Figura 3.1: Representação das Fases do DMAIC.	37
Figura 3.2: Áreas de Processo do CMMI e Áreas do DMAIC (SIVIY et al., 2007).	55
Figura 4.1: Fases do MiniDMAIC.	62
Figura 5.1: Gráfico de Controle do Projeto para <i>Baseline</i> Densidade de Defeitos em Testes Sistêmicos.....	82
Figura 5.2: Classificação dos Defeitos Encontrados em Testes Sistêmicos do Projeto..	83
Figura 5.3: Gráfico de Controle do Projeto para <i>Baseline</i> Densidade de Defeitos em Testes Sistêmicos com os Resultados Finais após Execução da Ação MiniDMAIC.....	89
Figura B.1: Tela Inicial de Criação da Ação MiniDMAIC no Jira.....	108
Figura B.2: Tela do Final da Fase “Medir” da Ação MiniDMAIC no Jira.....	110
Figura B.3: Tela do Final da Fase “Analisar” da Ação MiniDMAIC no Jira.....	111

Figura B.4: Tela do Final da Fase “Melhorar” da Ação MiniDMAIC no Jira..... 112

Figura B.5: Tela do Final da Fase “Controlar” da Ação MiniDMAIC no Jira..... 113

LISTA DE TABELAS

Tabela 2.1: Processo de Resolução de Problemas da Norma ISO/IEC 12207 (2008)....	13
Tabela 2.2: Resumo dos Atributos de Processo e Resultados Esperados do Nível A do MR-MPS para Análise e Resolução de Causas (SOFTEX, 2009c).	18
Tabela 3.1: Sumário das Fases e Passos do DMAIC (TAYNTOR, 2003).	38
Tabela 3.2: Ações para Cálculo do Nível Sigma	46
Tabela 3.3: Mapeamento das Fases e Passos do DMAIC com as Práticas das Áreas de Processo do CMMI (SIVIY et al., 2007).	55
Tabela 4.1: Objetivos Definidos por Tayntor (2003) para o Método DMAIC.	57
Tabela 4.2: Representação da Atividade “Definir” do MiniDMAIC.	63
Tabela 4.3: Representação da Atividade “Medir” do MiniDMAIC.	65
Tabela 4.4: Representação da Atividade “Analisar” do MiniDMAIC.	66
Tabela 4.5: Representação da Atividade “Melhorar” do MiniDMAIC.	67
Tabela 4.6: Representação da Atividade “Controlar” do MiniDMAIC.	69
Tabela 4.7: Comparação dos Passos do DMAIC Definidos por Tayntor (2003) com os Passos da Abordagem MiniDMAIC.	71
Tabela 4.8: Relacionamento dos Passos do MiniDMAIC com as Práticas Específicas de CAR.	73
Tabela 5.1: Caracterização dos Sub-Projetos do Projeto Piloto.	81

Tabela 5.2: Relacionamento das Causas e Ações Identificadas para Tratamento das Causas dos Defeitos.....	86
Tabela 5.3: Informações Consolidadas dos MiniDMAICs.....	90

1 INTRODUÇÃO

Este capítulo apresenta a introdução da pesquisa com a descrição da problemática que motivou e justificou o desenvolvimento desse trabalho. Os objetivos e metodologia aplicada são apresentados em seguida. Ao final, é apresentada a forma de organização textual dessa dissertação.

1.1 MOTIVAÇÃO

A busca incessante por produtos e serviços cada vez melhores leva muitas organizações a pesquisar e implantar várias técnicas, ferramentas e normas capazes de alavancar a qualidade de seus produtos. Dentro do contexto da busca por excelência, surge como solução para diminuir custo e melhorar a qualidade da implementação iniciativas tais como o CMMI (2006) e o *Six Sigma* (TAYNTOR, 2003).

O *Capability Maturity Model Integration* (CMMI) é um modelo de maturidade de desenvolvimento de produtos desenvolvido pelo *Software Engineering Institute* (SEI), que está cada vez mais sendo adotado nas empresas, uma vez que esse modelo busca orientar as organizações na implementação de melhorias contínuas em seu processo de desenvolvimento. Já o *Six Sigma* segundo Rotondaro et al. (2002) é um programa de qualidade estruturado que incrementa a qualidade por meio de melhoria contínua dos processos envolvidos na produção de um bem ou serviço.

Para atingir práticas do CMMI um grande número de organizações vem adotando o *Six Sigma* como estratégia. Essa metodologia não apenas cobre práticas dos níveis elevados de maturidade como também dos níveis mais baixos (SIVIY et al., 2005).

O *Six Sigma* e o CMMI têm objetivos compatíveis e o *Six Sigma* é, na maioria dos casos, extremamente compatível com outras iniciativas de qualidade que podem já estar implantada na organização. O *Six Sigma* pode ser executado em níveis macro e micro de uma organização e pode ser bem sucedido tanto com ferramentas gráficas elementares como ferramentas estatísticas avançadas (DENNIS, 1994).

Um dos aspectos fundamentais da melhoria da qualidade é a análise e resolução de problemas. Para isso, pode ser utilizado um método formal de resolução de problemas, que pode trazer inúmeros benefícios, tais como (BANAS QUALIDADE, 2009):

- Evitar que os solucionadores de problemas passem direto para a conclusão;
- Garantir a análise da causa-raiz;
- Desmistificar o processo de solução de problemas;
- Estabelecer ferramentas analíticas a usar e determinar quando utilizá-las.

Nesse contexto, a utilização de ferramentas da metodologia *Six Sigma*, como o DMAIC, tem se destacado. Ao contrário de outras metodologias para resolução de problemas, que focam apenas na eliminação do problema em si, o método DMAIC empregado pelo *Six Sigma* abrange desde a seleção dos problemas que merecem um tratamento mais aprofundado até o controle dos resultados obtidos no decorrer do tempo.

O método DMAIC mostra passo a passo como tais problemas devem ser tratados, agrupando as principais ferramentas da qualidade, ao mesmo tempo em que cria uma rotina padronizada na solução de problemas com uma comprovada eficiência na aplicação em organizações de software.

Apesar de adequados para o nível organizacional, os métodos formais de resolução de problemas podem ser inviáveis no nível de projetos. Um dos grandes desafios enfrentados por empresas que almejam o nível 5 do CMMI é justamente a implantação da área de processo de Análise e Resolução de Causas (*Causal Analysis and Resolution – CAR*) no contexto de projetos de software, uma vez que estes geralmente possuem recursos bastante limitados. Com isso, são tomadas apenas ações imediatas para resolução dos problemas e, na maioria das vezes, os mesmos voltam a acontecer.

Apesar de todos os benefícios do uso da metodologia *Six Sigma* em conjunto com o CMMI, a implementação da Análise e Resolução de Causas em projetos de software muitas vezes torna-se inviável, pelos seguintes motivos:

- Projetos DMAIC têm duração entre 3 a 6 meses. No entanto, projetos de desenvolvimento de software requerem rapidez na resolução de seus problemas, não podendo esperar muito tempo.
- Devido à grande necessidade da utilização de ferramentas estatísticas, o DMAIC pode tornar-se excessivamente caro; a economia obtida pode ser menor do que o custo para alcançar melhorias, e os projetos geralmente possuem recursos limitados.
- O nível de qualificação em estatística esperado da equipe DMAIC é bastante rigoroso; no entanto, no contexto de projetos de desenvolvimento de software, outros atributos como domínio do negócio e de gerência de projetos podem trazer maiores resultados do que o fato de ter uma equipe com grande conhecimento em estatística.

Diante deste contexto, este trabalho visa a elaboração de uma abordagem baseada no DMAIC do *Six Sigma*, denominada MiniDMAIC, para tratar a Análise e Resolução de Causas do CMMI, procurando reduzir as desvantagens acima descritas de uso do DMAIC. Também visa a implementação da abordagem em uma ferramenta de *workflow*, utilizando-a em projetos de desenvolvimento de software em uma organização.

1.2 OBJETIVOS

Este trabalho tem como objetivo principal realizar um estudo sobre análise e resolução de causas de problemas que ocorrem durante o projeto de desenvolvimento de software, propondo uma abordagem que atenda às práticas específicas da área de processo Análise e Resolução de Causas (nível 5 do CMMI), tomando como base o método DMAIC do *Six Sigma*.

Os objetivos específicos desse trabalho são:

- Pesquisar sobre o método DMAIC no contexto de análise e resolução de causas e como esse método poderia contribuir na implementação da área de processo Análise e Resolução de Causas (*Causal Analysis and Resolution – CAR*) do nível 5 do CMMI, focando no contexto de projetos de desenvolvimento de software.
- Definir uma abordagem de análise e resolução de causas para projetos de desenvolvimento de software com base no método DMAIC para atender às práticas da área de processo Análise e Resolução de Causas do CMMI. A ideia é adaptar o método DMAIC, suprimindo e adaptando passos do método para a realidade de projetos de software, deixando a abordagem MiniDMAIC mais simples e fácil de ser executada. A abordagem deve focar também nas lições aprendidas da análise de causas nos projetos para melhoria do processo organizacional e servir como fonte para outros projetos da organização que possuam os mesmos problemas enfrentados em projetos anteriores.
- Aplicar a abordagem proposta em uma empresa de software avaliada nível 5 de maturidade CMMI, e apresentar os resultados obtidos.

1.3 METODOLOGIA

Para atingir seus objetivos propostos, o desenvolvimento deste trabalho foi dividido em quatro etapas:

- (1) A primeira etapa consistiu na realização de uma revisão da literatura, especialmente em Análise e Resolução de Causas de Problemas, CMMI, *Six Sigma* e DMAIC.
- (2) A segunda etapa consistiu no desenvolvimento da abordagem proposta, o MiniDMAIC, a partir da revisão bibliográfica e experiência da autora do trabalho na execução de projetos DMAIC no contexto de uma organização de desenvolvimento de software.
- (3) A terceira etapa envolveu a aplicação da abordagem proposta em um projeto de desenvolvimento de software, e a análise dos resultados obtidos, para o refinamento dessa abordagem.

- (4) A partir dos resultados da experiência de uso, a quarta etapa consistiu na avaliação da abordagem proposta, ressaltando suas principais dificuldades em sua utilização, bem como seus pontos fortes e fracos.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está organizado em cinco capítulos, além desta introdução, como descrito resumidamente a seguir.

No Capítulo 2, **Análise e Resolução de Causas**, são apresentados métodos e ferramentas que auxiliam a análise causal, além de trabalhos similares à abordagem definida neste trabalho. Também é dado um enfoque na área de Análise e Resolução de Causas do nível 5 do CMMI.

No Capítulo 3, **O Six Sigma e o Método DMAIC**, é apresentado o *Six Sigma*, detalhando-se o método DMAIC.

No Capítulo 4, **MiniDMAIC**, é apresentada a abordagem proposta, que toma como base o referencial teórico apresentado anteriormente.

No Capítulo 5, **Experiência de Uso**, é apresentado o contexto da organização onde foi executada a abordagem, relatando a experiência obtida com a aplicação do MiniDMAIC.

No Capítulo 6, **Conclusão**, são apresentadas as principais conclusões e as contribuições deste trabalho, bem como suas limitações e perspectivas futuras. Além destes capítulos, o trabalho também apresenta os apêndices relacionados a implementação da abordagem na ferramenta *workflow* JIRA e o *template* de análise de causas para auxiliar a execução do MiniDMAIC.

2 ANÁLISE E RESOLUÇÃO DE CAUSAS

Neste capítulo é apresentada a definição de problemas e defeitos, bem como a forma de classificá-los e priorizá-los como um insumo para a análise e resolução de causas dos problemas. Também é apresentada uma pesquisa bibliográfica sobre a análise e resolução de causas, relatando estudos já feitos sobre esta área e a descrição da área de processo Análise e Resolução de Causas do modelo CMMI, que é o foco deste trabalho.

2.1 VISÃO GERAL

Projetos de desenvolvimento de software possuem muitos problemas, no entanto, muitas vezes devido aos recursos limitados do projeto, como tempo e custo, são tomadas apenas ações imediatas para resolução dos problemas e, na maioria das vezes, os mesmos voltam a acontecer.

Resolver um problema é uma atividade que necessita de tempo para: (i) estudar os sintomas, (ii) confirmar o problema, (iii) conduzir estudos preliminares, (iv) reunir a equipe, (v) diagnosticar as causas, (vi) pesquisar uma boa solução e (vii) implementá-la. À medida que os problemas não resolvidos vão se acumulando, os analistas e os gerentes experimentam vários tipos de pressões. É nesta condição que muitos problemas são sobrepostos aos outros, passando a frente da fila de tarefas por questões políticas e hierárquicas. Neste cenário, os analistas gastam mais tempo em explicações do porque prazos não foram cumpridos do que analisando e resolvendo problemas efetivamente (BOHN, 2000).

Existe uma grande deficiência na análise dos problemas encontrados e de suas verdadeiras causas (ROBITAILLE, 2004). As análises, quando realizadas, em geral não focam suficientemente o problema e suas possíveis origens. Consequentemente, são tomadas decisões erradas, que acabam por não resolverem o problema e, em alguns casos, até mesmo

agravando-o ainda mais, pois isto gera retrabalho, insatisfações e aumenta o custo ocasionado pela falta de qualidade.

Portanto, analisar e resolver problemas não é algo trivial e geralmente é uma das atividades mais difíceis e dispendiosas em qualquer tipo de empresa. Sendo assim, por décadas as organizações têm utilizado muitas ferramentas e metodologias diferentes para o tratamento dos problemas, todas elas com diferentes graus de eficácia. Entretanto, não faltam ferramentas para se buscar a solução dos problemas, e o que ocorre é a falta de disciplina, de coerência e de uma metodologia uniforme para o tratamento dos problemas, principalmente, no que concerne a problemas relacionados ao desenvolvimento de software.

A análise e resolução de causas dos problemas é uma excelente abordagem, pois estabelece ações efetivas que atacam a causa raiz dos problemas identificados. No entanto, para utilizarmos essa abordagem é importante saber a diferença entre os conceitos de problema e defeito, e como podemos classificar os defeitos para auxiliar na análise da causa raiz.

2.2 CONCEITO DE PROBLEMA E DEFEITO

Para entender melhor o problema é necessário saber o que é realmente o problema, o quanto urgente ele é, qual sua importância e quem é responsável por ele. Campos (1992) define problema como o “resultado indesejável de um processo” e, ainda, como “um item de controle com o qual não se está satisfeito”.

Segundo Pang (2002) defeitos são qualquer desvio entre a expectativa do cliente, e as especificações e/ou a implementação. Para Liker (2004), um defeito é uma não-conformidade percebida pelo cliente, seja este interno ou externo.

Existe uma diferença entre os conceitos de erro, defeito e falha. A terminologia padrão para Engenharia de Software do IEEE 610.12-1990 (*IEEE Standard Glossary of Software Engineering Terminology*) define os três conceitos abaixo:

- Erro: É um defeito cometido por um indivíduo ao tentar entender uma determinada informação, resolver um problema ou utilizar um método ou uma ferramenta.

- Defeito (ou Falta): É uma manifestação concreta de um erro num artefato de software. Um erro pode resultar em diversos defeitos.
- Falha: É o comportamento operacional do software diferente do esperado pelo usuário. Uma falha pode ter sido causada por diversas faltas e algumas faltas podem nunca causar uma falha.

2.3 CLASSIFICAÇÃO DE DEFEITOS

Sullivan e Chillarege (1991, 1992) apresentam estudos acerca do impacto de defeitos de software na disponibilidade de sistemas da IBM, baseando-se em dados sobre defeitos registrados durante cinco anos. A conclusão de que a distribuição de falhas, quando classificadas pelo tipo de defeito, é dependente do estágio do desenvolvimento do produto, permitiu a Chillarege et al. (1992) estenderem o conceito e propor o esquema de classificação ODC (*Orthogonal Defect Classification*), que é composta por categorias ortogonais, como tipo de defeito, origem e impacto.

O problema de se injetar defeitos que realmente representam falhas e que podem ter sido efetivados por programadores durante o desenvolvimento foi, pela primeira vez, publicado por Christmansson e Chillarege (1996) no âmbito do projeto ODC da IBM. A classificação das falhas, da maneira como foi proposta por Christmansson, é baseada em um conhecimento prévio das falhas reveladas no sistema, informação que, raramente, se encontra disponível para os sistemas existentes e, muito menos, para os sistemas que acabam de ser desenvolvidos.

Chillarege et al. (1992) propõem a seguinte classificação ortogonal dos tipos de defeitos:

- **Função** – Defeito que afeta a capacidade, as interfaces com o usuário final, as interfaces do produto, a interface com a arquitetura de hardware ou as estruturas globais de dados.
- **Lógica** – Defeito que afeta a funcionalidade de um módulo de código e pode ser corrigido pela re-implementação de um algoritmo ou estrutura de dados local, sem necessidade de solicitar uma alteração de projeto de alto nível.

- **Interface** – Defeito que afeta a interação dos componentes através de macros, chamadas de instruções e / ou parâmetros.
- **Verificação** – Defeito na lógica do programa, que falha ao validar dados e valores apropriadamente, antes de utilizá-los.
- **Atribuição** – Defeito na estrutura de dados ou na inicialização do bloco de código.
- **Sincronia/sequência** – Defeito que envolve a sincronia de recursos compartilhados e em tempo real.
- **Algoritmo** – Defeito que envolve a eficiência ou a correção do algoritmo ou da estrutura de dados, mas não do projeto.

Uma outra forma de classificarmos os defeitos é a da fase do ciclo de vida do desenvolvimento de software em que corresponde a fase em que o defeito é encontrado ou inserido. Iremos neste trabalho utilizar as seguintes fases do ciclo de vida de desenvolvimento de software: requisitos, análise e projeto, implementação, testes e disponibilização.

Chillarege (2006) diz que a classificação ortogonal de defeitos (ODC) pode auxiliar na análise de causa raiz facilitando a identificação da causa raiz do problema e diminuindo o custo da análise de causas, pelo fato de análise ser quantitativa pelos parâmetros definidos no ODC. Neste caso, vale ressaltar que o autor tratou de defeitos como sendo falhas, ao considerarmos as definições de defeito, erro e falha apresentados na IEEE 610.12-1990.

A abordagem proposta nesta dissertação define um *template* de análise de causas que propõe a classificação de defeitos por origem (fase em que o defeito é encontrado ou inserido), impacto e tipo de defeitos com base na classificação ortogonal de defeitos.

2.4 ANÁLISE E RESOLUÇÃO DE CAUSAS

A análise e resolução de causas de problemas melhora a qualidade e produtividade por meio da prevenção da introdução de defeitos nos produtos e da resolução de problemas na execução do processo. Como consequência, ajudam a diminuir a probabilidade de ocorrência de desvios no alcance dos objetivos de qualidade e de desempenho do processo do projeto (SEI, 2006).

Ainda segundo o SEI (2006), a análise de causas também pode ser feita em problemas não relacionados a defeitos. Por exemplo, a análise de causa pode ser usada para melhorar a qualidade de atributos, tais como, o tempo de ciclo. Propostas de melhoria, simulações, modelos dinâmicos de sistemas, análises de engenharia, novas diretrizes de negócio ou outros itens podem iniciar essa análise.

Segundo Kalinowski (2009), a primeira abordagem para análise de causas encontrada foi a descrita por Endres (1975), na IBM. Esta abordagem lida com a análise individual de defeitos de software, de modo que possam ser categorizados e suas causas identificadas, permitindo a tomada de ações para prevenir sua ocorrência em projetos futuros ou ao menos assegurar sua detecção nestes projetos. A análise dos defeitos nesta abordagem ocorre ocasionalmente, assim como as ações corretivas.

A técnica RCA (*Root Cause Analysis*) (AMMERMAN, 1998), que é uma das técnicas utilizada para analisar a causa raiz de um problema, tem por objetivo a formulação de recomendações para eliminar ou reduzir a incidência dos erros mais recorrentes e de maior custo nos projetos de desenvolvimento de software da organização. Segundo Robitaille (2004), o RCA tem o propósito de investigar os fatores não tão visíveis que tem contribuído para identificação de não conformidades ou problemas potenciais.

Triz (ALTSHULLER, 1999) é outra metodologia desenvolvida para análise de causas. Ela é uma metodologia sistemática orientada ao ser humano e baseada em conhecimento. Sua teoria define os problemas onde a sua solução faz surgir o aparecimento de novos problemas.

Card (2005) apresenta uma abordagem de análise causal de defeitos que está resumida em seis passos:

- **Selecionar a amostra do problema** – Em organizações maduras, a análise causal pode ser disparada por uma instabilidade detectada em um gráfico de controle estatístico de processos (FLORAC e CARLETON, 1999). Os dados relacionados a esta instabilidade tornam-se então alvo da análise e uma amostra destes dados deve ser selecionada para a reunião de análise causal. Em organizações de menor maturidade, a análise causal pode ser disparada de uma maneira mais informal, por exemplo, pela observação de algumas tendências ou comportamentos estranhos em métricas coletadas relacionadas a defeitos.

- **Classificar problemas selecionados** – Entre as informações básicas a serem coletadas a respeito dos defeitos estão o tipo do defeito, o momento (ou fase de desenvolvimento) da sua introdução, o momento da sua detecção, e o esforço necessário para removê-lo. O evento que disparou a detecção do defeito é também considerado pela abordagem ODC (*Orthogonal Defect Classification*) criada pela IBM (CHILLAREGE et al., 1992), que sugere registrar os seguintes atributos para os defeitos: tipo do defeito, omitido/incorrecto, gatilho (*trigger* - como foi detectado), origem (nova função ou correção) e impacto. O uso de tabelas de índice-cruzado, histogramas ou gráficos de Pareto pode ajudar a agrupar defeitos com base na sua respectiva classificação. Erros sistemáticos (tópico do próximo passo) possuem maior probabilidade de serem encontrados entre os tipos de defeitos mais comuns.
- **Identificar erros sistemáticos** – Um erro sistemático resulta em problemas similares se repetindo em diferentes ocasiões (CARD, 2005). Normalmente, erros sistemáticos estão associados com uma atividade ou parte do produto específica. Encontrar erros sistemáticos indica a existência de oportunidades de melhoria para o processo.
- **Identificar as principais causas** – Muitos fatores podem contribuir para um erro sistemático e tratar todos eles muitas vezes não é economicamente viável. Assim, deve haver um esforço no sentido de encontrar as causas principais. Neste passo é que a compreensão de sistemas causais, relacionando causas, problemas e sintomas, se torna uma habilidade útil. Entre as técnicas utilizadas para encontrar as principais causas temos o uso dos diagramas de causa e efeito (ou diagramas de Ishikawa) (ISHIKAWA, 1976).
- **Desenvolver itens de ação** – Uma vez que a causa principal de um defeito sistemático foi encontrada, itens de ação devem ser desenvolvidos para promover a prevenção (ou detecção mais cedo, em casos que a prevenção não é possível) dos erros sistemáticos. Normalmente, o número de ações é pequeno. As ações devem ser tão pontuais e específicas quanto possível, facilitando o acompanhamento objetivo do estado de sua implantação.
- **Documentar os resultados da reunião de análise causal** – Registros dos resultados da reunião de análise causal precisam ser mantidos para assegurar que

as ações serão implementadas. De acordo com CARD (2005), para o sucesso de um programa de análise causal é essencial que uma equipe de ação seja formada. Os resultados da reunião de análise causal devem ser fornecidos para esta equipe, que tem entre suas responsabilidades encontros regulares a respeito das ações propostas. Uma das formas de alcançar este resultado é formar uma equipe responsável pela realização dos planos de ação. Os resultados da reunião de análise causal devem ser fornecidos para esta equipe, que pode ter dentre suas responsabilidades (CARD, 2005): (i) priorizar os itens de ação; (ii) resolver conflitos e combinar propostas relacionadas (no caso de análises causais atuando simultaneamente em diferentes projetos); (iii) planejar a implementação dos itens; (iv) alocar recursos para implementar os itens de ação; (v) monitorar o progresso da implementação e a eficiência das ações (por meio de reuniões periódicas, por exemplo); e (vi) comunicar ações e sua situação para os interessados.

Já Kalinovski (2009) descreve uma abordagem denominada DBPI (*Defect Based Process Improvement*), e se baseia em uma rica revisão sistemática para elaboração da abordagem para análise de causas. Entre as principais contribuições e diferenças para o processo de prevenção de defeitos destacadas pelo autor são:

- Detalhar apoio para a realização eficiente de análise causal de defeitos, baseado no conhecimento obtido na revisão sistemática e nas diretrizes obtidas a partir de seus resultados (KALINOVSKI et al., 2008). Isto é feito através do refinamento de atividades em tarefas e indicando técnicas específicas que podem ser utilizadas para realizar estas tarefas.
- Prover um mecanismo para permitir a atualização dos relacionamentos de causa e efeito da organização com base nos resultados das reuniões de análise causal, permitindo um maior entendimento do contexto organizacional. Isto endereça a oportunidade de pesquisa identificada na revisão sistemática e destacada em (KALINOVSKI et al., 2008).

Gonçalves (2008b) propõe uma abordagem de análise causal desenvolvida com base no método PDCA¹ e que aplica a metodologia multicritério de apoio à decisão com o intuito

¹ O método PDCA é um instrumento utilizado para o controle e melhoria de processos, e é composto de quatro fases: o planejamento (*Plan*), a execução (*Do*), o controle (*Check*), e ação corretiva (*Act*) (DEMING, 1986).

de auxiliar na análise das causas de problemas complexos no âmbito de organizações de software.

A ISO/IEC 12207 (2008) descreve um arcabouço de processo de resolução de problemas para analisar e resolver os problemas (incluindo não conformidades) de qualquer natureza ou fonte, que são descobertos durante a execução do desenvolvimento, operação, manutenção ou outros processos. O objetivo é prover os meios em tempo adequado e de forma responsável e documentada para garantir que todos os problemas encontrados sejam analisados e resolvidos e tendências sejam identificadas. Na Tabela 2.1 é descrito o arcabouço de processo de resolução de problemas da ISO/IEC 12207 (2008).

Tabela 2.1: Processo de Resolução de Problemas da Norma ISO/IEC 12207 (2008).

Atividade	Tarefas	Requisitos
Implementação do processo.	Um processo de resolução de problema deve ser estabelecido para tratar todos os problemas (incluindo não-conformidades) detectados nos produtos de software e atividades.	<p>O processo deve ser de ciclo fechado (<i>closed-loop</i>), garantindo que: todos os problemas detectados sejam prontamente relatados e incluídos no processo de resolução de problema; a ação seja iniciada nos problemas detectados; as partes relevantes sejam alertadas da existência do problema, quando apropriado; as causas sejam identificadas, analisadas e, quando possível, eliminadas; a resolução e sua aplicação sejam alcançadas; a situação seja rastreada e relatada; e os registros dos problemas sejam mantidos, conforme estipulado no contrato.</p> <p>O processo deveria conter um esquema para categorizar e priorizar os problemas. Cada problema deveria ser classificado por categoria e prioridade para facilitar a análise de tendência e resolução de problema.</p> <p>A análise deve ser executada para detectar tendências nos problemas relatados</p> <p>As resoluções de problemas e suas aplicações devem ser avaliadas para: verificar se os problemas foram resolvidos, se as tendências adversas foram revertidas e se as alterações foram implementadas corretamente nos produtos de software e atividades apropriados; e determinar se problemas adicionais foram introduzidos.</p>
Resolução do problema	Quando problemas (incluindo não-conformidades) forem detectados em um produto de software ou em uma atividade, um relatório de problema deve ser preparado para	

	descrever cada problema detectado. O relatório de problema deve ser usado como parte do processo de ciclo fechado (<i>closed-loop</i>) descrito acima: a partir da detecção do problema, ao longo da investigação, análise e resolução do problema e sua causa, e para detectar tendências.	
--	---	--

A maioria das pesquisas citadas neste trabalho propõe abordagens para análise de causas focando no nível organizacional. No entanto, muitas vezes é necessária a realização de análises de causas dentro de projetos, de forma que esta seja rápida e eficaz, atacando as causas raízes do problema. Em organizações que buscam os altos níveis de maturidade de modelos de melhoria de processo como o CMMI e MPS.Br, esta prática é solicitada dentro do projeto para manter a aderência ao modelo.

Além disso, das abordagens pesquisadas que envolviam análise e resolução de causas, nenhuma é baseada no método DMAIC. A abordagem proposta neste trabalho tem como objetivo tornar eficaz a análise de causas no contexto de projetos fornecendo um conjunto de passos estruturados com base no método DMAIC, de forma que estes sejam simples de serem executados.

2.5 FERRAMENTAS UTILIZADAS NA ANÁLISE DE CAUSAS

Algumas ferramentas são destacadas por Robitaille (2004) na Análise de Causa Raiz:

- **Cinco Porquês:** desenvolvida por *Sakichi Toyoda* (fundador da Toyota), é baseada na realização de 5 iterações, perguntando o porquê daquele problema, sempre questionando a causa anterior. E, na prática, não é necessário fazer 5 perguntas, pode ser mais ou menos, o importante é chegar à causa do problema. Um exemplo de cinco porquês é ilustrado abaixo:
 - Problema: Os clientes estão reclamando muito dos atrasos nas entregas.
 - Porque há atrasos? Porque o produto nunca sai da fábrica no momento que deveria.

- Porque o produto não sai quando deveria? Porque as ordens de produção estão atrasando.
 - Porque estas ordens atrasam? Porque o cálculo das horas de produção sempre fica menor do que a realidade.
 - Porque o cálculo das horas está errado? Porque estamos usando um software ultrapassado.
 - Porque estamos usando este software? Porque o engenheiro responsável ainda não recebeu treinamento no software mais atual.
- **Brainstorming:** é um método menos estruturado do que os 5 porquês e mais vantajoso quando a causa de um problema é instável e elusiva. É particularmente útil em situações onde ações corretivas falharam na resolução de problemas porque os donos do processo também assumiram que eles não sabiam a causa ou não investigaram o suficiente os fatores de contribuição. O *brainstorming* começa situando o problema usando evidências substanciais, depois as pessoas podem classificar os fatos conhecidos durante o processo, e então é feita uma avaliação preliminar. O tempo alocado para o *brainstorming* segundo Robitaille (2004) não deve passar de 5 minutos por exercício e todas as idéias das pessoas são aceitas. A filtragem das idéias é executada em atividades seguintes com um processo de eliminação. Uma vez terminado esse exercício pode ser investigada a causa raiz do problema. O *brainstorming* vai ajudar a decidir quais documentos ou registros revisar, quem deve ser entrevistado e que processos devem ser modificados.
 - **Diagrama de Causa e Efeito (Espinha de Peixe ou Diagrama de Ishikawa):** identifica todos os processos e fatores que contribuem para um problema ou inversamente ao atendimento de uma meta desejada. Na Figura 2.1 mostra-se um exemplo de diagrama de causa e efeito proposto por Robitaille (2004). O diagrama ilustra a definição do problema, devem ser atribuídos os principais títulos ou tipos de causas (máquina, método, material, medição, pessoal e ambiente) às principais espinhas do peixe, para direcionar e estruturar a sessão de *brainstorming*, técnica que auxilia a identificação das causas no diagrama de causa e efeito.

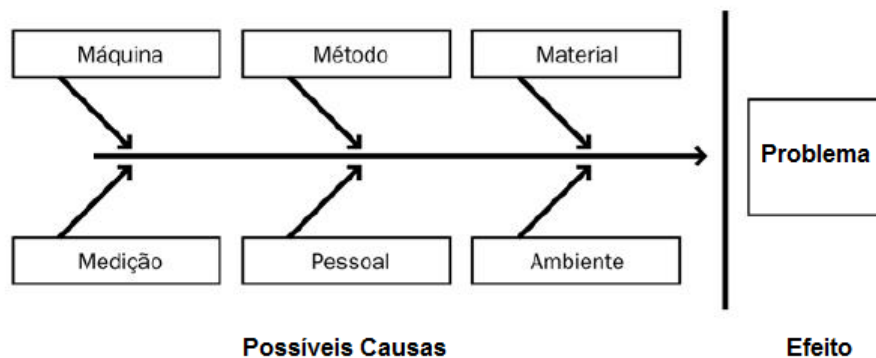


Figura 2.1: Exemplo de Diagrama de Causa e Efeito (ROBITALLE, 2004).

- Diagrama de Pareto:** é uma ferramenta simples que apresenta graficamente a quantidade de problemas e suas causas. Werkema (2006) destaca que o princípio de Pareto estabelece que um problema pode ser atribuído a um pequeno número de causas². Um exemplo de gráfico de análise de Pareto é mostrado na Figura 2.2. Neste exemplo não tem identificado o problema, mas as barras maiores identificam o maior número de problemas e indicam onde as possíveis causas do problema podem ser encontradas.

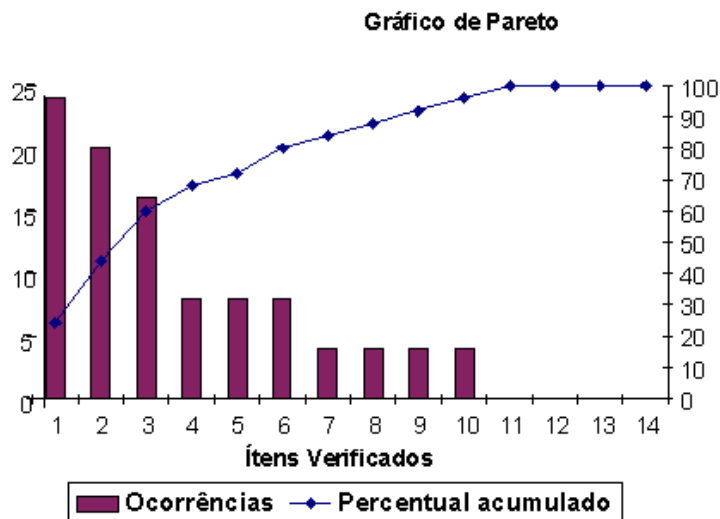


Figura 2.2: Exemplo de Gráfico de Pareto.

² Lei de Pareto: uma quantidade consideravelmente pequena de causas, tipicamente, causará a grande maioria dos problemas. Normalmente, é referenciada como princípio 80/20 (80% dos problemas se devem a 20% das causas) (SOFTEX, 2009a).

2.6 ANÁLISE E RESOLUÇÃO DE CAUSAS NO MR-MPS

O modelo MR-MPS é um modelo de referência brasileiro para melhoria do processo de software. O modelo MPS baseia-se nos conceitos de maturidade e capacidade de processo para a avaliação e melhoria da qualidade e produtividade de produtos de software e serviços correlatos (SOFTEX, 2009a).

O MR-MPS contém os requisitos que os processos das unidades organizacionais devem atender para estar em conformidade com o Modelo. Ele contém as definições dos níveis de maturidade, processos e atributos do processo (SOFTEX, 2009a).

Além disso, o MR-MPS define níveis de maturidade que são uma combinação entre processos e sua capacidade. A capacidade do processo é a caracterização da habilidade do processo para alcançar os objetivos de negócio, atuais e futuros; estando relacionada com o atendimento aos atributos de processo associados aos processos de cada nível de maturidade (SOFTEX, 2009a). Os níveis de maturidade do MR-MPS são exemplificados na Figura 2.4.

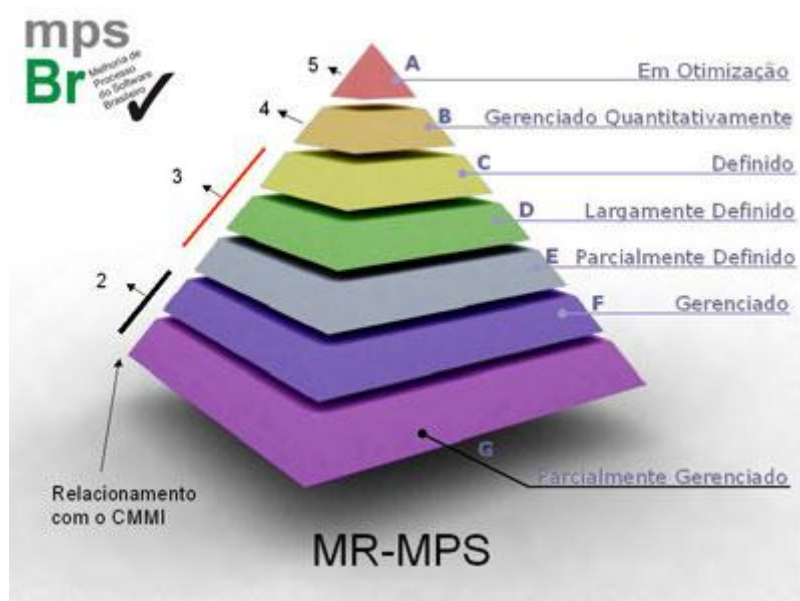


Figura 2.3: Níveis de Maturidade do Modelo MR-MPS (SOFTEX, 2009a).

No nível A do MR-MPS (SOFTEX, 2009c), melhorias de processos e de tecnologias são selecionadas e implementadas nos processos selecionados para controle estatístico, como forma de contribuir para o alcance de objetivos de melhoria de processo. Nesse nível, melhorias de processo e de tecnologias devem ser identificadas de tal forma que, quando implementadas, removam causas comuns de variação de processo, causas raiz de defeitos e outros problemas (SEI, 2006, apud SOFTEX, 2009c).

Segundo o MR-MPS (SOFTEX, 2009c), é importante ressaltar que, enquanto a identificação das causas de defeitos é realizada com base nos processos definidos para os projetos, as melhorias de processo identificadas a partir dessas análises têm como objetivo melhorar o conjunto de processos padrão da organização para prevenir a ocorrência desses defeitos em projetos futuros.

O nível A do MR-MPS trata a análise e resolução de causas nos resultados de atributos de processos conforme a Tabela 2.2.

Tabela 2.2: Resumo dos Atributos de Processo e Resultados Esperados do Nível A do MR-MPS para Análise e Resolução de Causas (SOFTEX, 2009c).

Atributo de Processo	Resultados	Resumo
AP 5.1 - O processo é objeto de melhorias e inovações	RAP 37 - Defeitos e outros problemas são identificados, classificados e selecionados para análise	Este resultado esperado tem como objetivo manter os registros dos problemas encontrados a partir da realização das diferentes atividades dos demais processos.
	RAP 38 - Defeitos e outros problemas selecionados são analisados para identificar sua causa raiz e soluções aceitáveis para evitar sua ocorrência futura	O objetivo deste resultado esperado é identificar erros sistemáticos e suas causas principais. Também é objetivo deste resultado esperado, desenvolver e registrar itens de ação. Os itens de ação devem ser desenvolvidos para promover a prevenção (ou detecção mais cedo, em casos que a prevenção não é possível) dos erros sistemáticos que resultam no tipo de problema analisado.
	RAP 39 - Dados adequados são analisados para identificar causas comuns de variação no desempenho do processo	Para alcançar este resultado esperado, oportunidades de melhoria devem ser derivadas para remover a origem dos problemas (causa raiz) que resultam em variação no desempenho e na capacidade do processo. Essas melhorias devem ser identificadas a partir do entendimento de causas de problemas existentes ou de potenciais problemas de processo induzidos por objetivos de melhoria de processo.
	RAP 42 - Uma estratégia de implementação é estabelecida e executada para alcançar os objetivos de melhoria do processo e para resolver problemas	A estratégia de implementação deve conter informações sobre as melhorias que serão implementadas, a forma mais adequada de implantação das melhorias para minimizar interrupções nos processos, a identificação de ativos de processo que necessitam ser alterados, o registro de necessidades de treinamentos para garantir que os membros da organização estarão capacitados para realizar suas atividades de acordo com as melhorias

		implantadas, a identificação de medidas de retorno do investimento da implementação das melhorias, entre outras. Uma das formas de alcançar este resultado é formar uma equipe responsável pela realização dos planos de ação.
AP 5.2 - O processo é otimizado continuamente	RAP 45 - As ações implementadas para resolução de problemas e melhoria no processo são acompanhadas com medições para verificar se as mudanças no processo corrigiram o problema e melhoraram o seu desempenho	Uma vez que o processo modificado pela implementação dos itens de ação é implantado no projeto, o efeito das mudanças deve ser verificado para obter evidência de que a mudança no processo corrigiu o problema e melhorou o seu desempenho. O objetivo deste resultado esperado é, então, avaliar a efetividade das mudanças, nos projetos, com relação aos resultados atuais e realizar ajustes, quando necessário, para garantir o alcance dos objetivos de melhoria do processo.
	RAP 46 - Dados da análise de causas de problemas e de sua resolução são armazenados para uso em situações similares	A organização deve manter dados da análise de causas de problemas e dos itens de ação, de modo que outros projetos e unidades organizacionais possam realizar mudanças apropriadas em seus processos e atingir resultados similares.

2.7 ANÁLISE E RESOLUÇÃO DE CAUSAS NO CMMI

O *Capability Maturity Model Integration* (CMMI) (CHRISISS et al., 2006) é um modelo de maturidade de desenvolvimento de produtos desenvolvido pelo *Software Engineering Institute* (SEI), que está cada vez mais sendo adotado nas empresas, uma vez que este modelo busca orientar as organizações na implementação de melhorias contínuas em seu processo de desenvolvimento.

Em 1997, o SEI verificou a necessidade de integração dos modelos CMM (SW-CMM; SE-CMM-EIA/IS; IPD-CMM) para que fosse implementado um processo único de melhoria para toda a organização, tendo também o intuito de reconciliar esses modelos com outras metodologias e padrões, tais como a ISO 9001, 15504, entre outros.

Sua estrutura inclui elementos comuns e melhores práticas dos modelos acima citados, bem como os requisitos e métodos específicos do CMMI, provendo para a organização a possibilidade de selecionar elementos que melhor se apliquem à sua estratégia de negócios. O âmbito do CMMI não é somente software, mas desenvolvimento multidisciplinar, embora apresente forte fundamentação em software.

Segundo Chrissis et al. (2006), o propósito do CMMI é estabelecer um guia para melhorar o processo da organização e sua capacidade para gerenciar o desenvolvimento, aquisição e manutenção de produtos e serviços. Por ser um guia de referência, necessita ser interpretado e adaptado para a realidade de cada organização.

Há duas diferentes representações no CMMI: a representação contínua e a representação por estágios.

A representação contínua utiliza níveis de capacidade para medir a melhoria do processo da organização. Essa representação contém seis níveis de capacidade: incompleto, executado, gerenciado, definido, gerenciado quantitativamente e, em otimização. Suas áreas de processo são independentes dos níveis de maturidade, ficando relacionadas apenas com a capacidade do processo, ou seja, uma determinada área de processo em particular poderá ter sua capacidade avaliada independente das outras áreas de processo, de acordo com o objetivo estratégico da organização. Por exemplo, a área de processo de gerência de requisitos poderia ser mais bem explorada e apresentar um nível de capacidade maior do que a área de processo de subcontratação se a organização não utiliza muito este artifício.

Na representação por estágios, as áreas de processo comuns são agrupadas em diferentes níveis de maturidade. São cinco os níveis de maturidade que estão representados na Figura 2.4: inicial, gerenciado, definido, gerenciado quantitativamente e, em otimização. O fato de a organização estar no nível de maturidade dois significa que todas as áreas de processo obrigatórias para aquele nível foram satisfeitas igualmente.

Este trabalho tem como foco a representação por estágios do CMMI, uma vez que é apresentada a utilização da metodologia proposta no contexto dos níveis de maturidade.

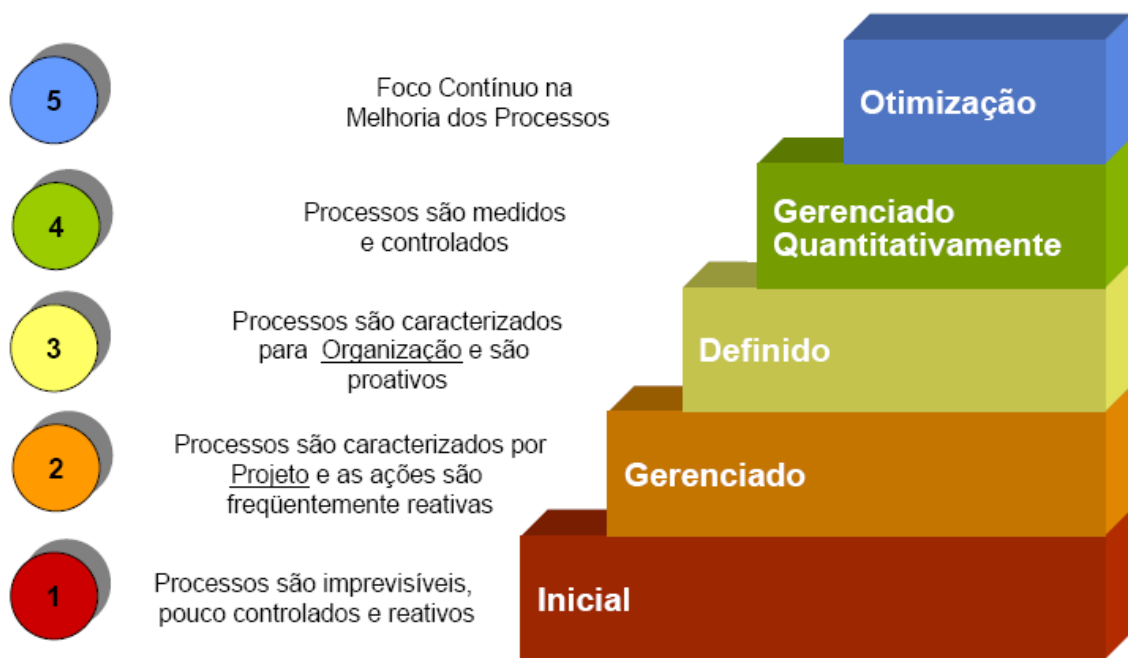


Figura 2.4: Níveis de Maturidade da Representação por Estágios do CMMI.

No modelo CMMI todas as áreas de processo possuem práticas específicas e práticas genéricas a serem atingidas, para obter o cumprimento de cada área de processo. Segundo o CMMI (2006), as práticas específicas e genéricas são definidas como:

- **Práticas Específicas (SP):** Um componente esperado do modelo que é considerado importante para atingir o objetivo específico associado. As práticas específicas descrevem as atividades esperadas para atingir as metas específicas de uma área de processo.
- **Prática Genéricas (GP):** Um componente esperado do modelo que é considerado importante para atingir o objetivo genérico associado. As práticas genéricas associadas ao objetivo genérico descrevem as atividades que são esperadas para se atingir o objetivo genérico e contribuir com a institucionalização dos processos associados à área de processo.

2.7.1 Nível 5 de Maturidade

O foco do nível 5 de maturidade está na melhoria contínua dos processos, mas agora o conjunto de processos padrão da organização é o documento que fornece o foco principal.

Usando o processo padrão da organização (que agora deve refletir o verdadeiro funcionamento da organização), a melhoria do processo pode ser tratada de uma maneira muito mais eficiente. O nível 4 foca nas causas especiais³ de variação no processo da organização. O nível 5 tenta encontrar as causas comuns⁴, e tratá-las, resultando em inúmeras melhorias. Medições são usadas para selecionar as melhorias e estimar os custos e benefícios em atender as melhorias propostas. As mesmas medições podem ser usadas para justificar esforços para futuras melhorias (KULPA et al., 2003).

No nível 5, o foco é constantemente revisar e melhorar os processos, mas estas melhorias devem ser introduzidas de maneira disciplinada e ordenada para gerenciar e manter a estabilidade do processo.

As principais características deste nível são (KULPA et al., 2003):

- A melhoria contínua do desempenho dos processos por meio de processos incrementais e inovadores e melhorias tecnológicas.
- Os objetivos quantitativos da melhoria dos processos, para a organização, são estabelecidos e revisados continuamente.
- Os gerentes são aptos a estimar e monitorar a eficácia das mudanças.
- As análises têm forte relação com o tratamento de causas comuns e variação do processo.
- As medições são usadas para selecionar melhorias e inovações, estimar e provar (acompanhar) o custo/benefício destas mudanças.

O nível 5 possui duas áreas de processo:

3 As causas especiais provocam desvios que excedem os limites estabelecidos para a variação do comportamento do processo. São eventos que não fazem parte do curso normal do processo e provocam mudança no padrão de variação. A remoção das causas especiais leva a ajustes no processo. As causas especiais são também chamadas de atribuíveis, pois podem ser identificadas, analisadas e utilizadas como diretrizes para prevenção de ocorrências futuras (SOFTEX, 2009b).

4 As causas comuns provocam desvios dentro de um limite aceitável para o comportamento do processo. São variações que pertencem ao processo, ou seja, é o resultado de interações normais dos componentes do processo: pessoas, máquinas, ambientes e métodos (SOFTEX, 2009b).

- Inovação e Desenvolvimento Organizacional (*Organizational Innovation and Deployment – OID*)
- Análise e Resolução de Causas (*Causal Analysis and Resolution – CAR*)

Neste capítulo, iremos detalhar apenas a área de processo de Análise e Resolução de Causas, por ser o principal foco deste trabalho.

2.7.2 Área de Processo: Análise e Resolução de Causas

A área de processo de Análise e Resolução de Causas compõe o nível 5 do CMMI DEV na representação por estágios. Segundo Chrissis et al. (2006), o propósito desta PA (área de processo) é identificar as causas de defeitos e de outros problemas e tomar ações para prevenir que ocorram no futuro, proporcionando uma melhoria da qualidade e da produtividade de um produto ou serviço. A Figura 2.5 mostra o diagrama de contexto desta área de processo.

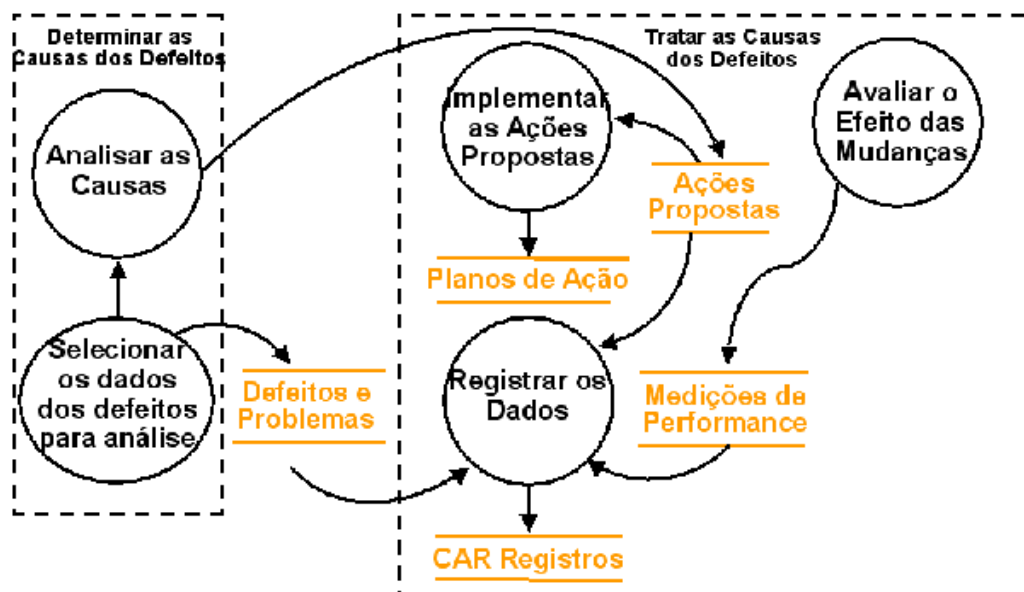


Figura 2.5: Diagrama de Contexto da Análise e Resolução de Causas (AHERN et al., 2003).

A análise de causas também pode ser usada para resolver problemas não relacionados a defeitos. Por exemplo, a análise de causas pode ser usada para analisar as melhorias sugeridas e novas diretrizes de negócio (KASSE, 2004). Vale ressaltar que a resolução de alguns problemas não gera benefício econômico direto, mas é estratégica para a empresa, pois

os problemas estão vinculados à sua imagem, à sua visão, às suas políticas e aos seus objetivos.

Os tipos de defeitos e outros problemas encontrados são analisados para evitar tendências, com base no entendimento do processo definido e como ele é implementado. Por exemplo, a análise e resolução de causas poderá ser utilizada para melhorar os atributos de qualidade como o tempo de ciclo de vida.

Algumas vezes pode ser impraticável executar a análise e resolução de causas em todos os defeitos. Nestes casos, são feitas análises de custos/benefício entre: os investimentos estimados e o retorno em qualidade, produtividade e ciclo de vida, e posteriormente os alvos de defeitos são selecionados (CHRISISS et al., 2006).

A fim de ter um processo efetivo para encontrar a causa da raiz de um defeito de modo que o mesmo possa ser removido, é importante obter dados relevantes do defeito através de múltiplas fontes (KASSE, 2004). Segundo o SEI (2006) alguns exemplos de dados de defeitos relevantes:

- Defeitos reportados pelo cliente.
- Defeitos reportados pelos usuários finais.
- Defeitos encontrados em revisão por pares.
- Defeitos encontrados nos testes.

Não só os defeitos são insumos na análise de causas. O SEI (2006) cita alguns exemplos de dados de problemas relevantes, que também podem ser insumo para análise de causas:

- Relatórios de problemas de gerenciamento de projeto que requerem ações corretivas.
- Problemas de capacidade de processo.
- Medições de duração de processo.
- Medições de valor agregado pelo processo (ex: índice de desempenho de custo).
- Medições de utilização de recursos ou de tempo de resposta.

Para selecionar defeitos para análise posterior, outras questões podem ser levadas em consideração, tais como (KASSE, 2004):

- Frequência da ocorrência.
- Similaridade entre defeitos.
- Custo da análise.
- Tempo e recursos necessários.
- Questões de segurança.

Nas seções a seguir, serão descritos os objetivos específicos da PA de CAR segundo o modelo CMMI (2006), bem como suas práticas específicas. Não serão detalhados os objetivos e práticas genéricas por não serem o foco deste trabalho, uma vez que os mesmos envolvem várias outras áreas de processo.

2.7.2.1 Determinar as Causas dos Defeitos (SG1)

As causas-raiz de defeitos e de outros problemas são determinadas de forma sistemática. Causa-raiz é a origem de um defeito que, quando removida, faz com que o defeito diminua, ou seja, eliminado. Este objetivo específico possui as seguintes práticas específicas:

- **SP 1.1 - Selecionar os dados dos defeitos para análise:** consiste em obter os defeitos relevantes e determinar que outros defeitos e problemas sejam analisados.
- **SP 1.2 - Analisar as causas:** consiste em realizar a análise de causas de defeitos e de outros problemas selecionados e propor ações para tratá-las. O objetivo é obter soluções para os problemas identificados a partir de análise dos dados relevantes visando a geração de ações propostas para implementação.

2.7.2.2 Tratar as Causas dos Defeitos (SG2)

As causas-raiz dos defeitos e de outros problemas são tratadas de forma sistemática para prevenir sua recorrência. Os projetos executados de acordo com um processo bem definido analisam sistematicamente as operações em que ainda ocorrem problemas e implementam mudanças de processo para eliminar as causas-raiz dos problemas selecionados. Este objetivo específico possui as seguintes práticas específicas:

- **SP 2.1 - Implementar as ações propostas:** consiste em implementar ações propostas selecionadas que foram desenvolvidas durante análise de causa. As ações propostas descrevem as tarefas necessárias para remover as causas-raiz dos defeitos ou dos problemas analisados e para evitar recorrência. Recomenda-se implementar em larga escala apenas as mudanças que realmente possam agregar valor.
- **SP 2.2 - Avaliar os Efeitos das Mudanças:** uma vez que o processo modificado é implementado no projeto, o efeito das mudanças deve ser verificado para reunir evidências de que a mudança do processo está corrigindo o problema e melhorando o desempenho.
- **SP 2.3 - Registrar os dados:** os dados são registrados de forma que outros projetos e a organização possam fazer mudanças de processo apropriadas e alcançar resultados similares.

A aplicação efetiva das práticas de CAR requer que a organização tenha um processo que é quantitativamente entendido e gerenciado.

Defeitos ou outros problemas devem primeiro ser selecionados para análise, usando-se, por exemplo, a análise de Pareto. As causas raiz dos defeitos selecionados podem ser então determinadas, usando-se, por exemplo, diagrama de causa e efeito. As propostas para remoção das causas raiz são então avaliadas e implementadas e os resultados são medidos (MUTAFELIJA et al., 2003).

Vale ressaltar que, de acordo com o CMMI (SEI, 2006), os dois objetivos específicos apresentados anteriormente devem ser implementados tanto no nível de projetos quanto no nível organizacional.

2.8 CONCLUSÃO DO CAPÍTULO

Este capítulo apresentou conceitos, ferramentas e métodos de análise e resolução de causas de problemas, focando na área de processo Análise e Resolução de Causas do nível 5 de maturidade do CMMI. Também foram conceituados os problemas e defeitos e forma de classificação e priorização de defeitos como um insumo da análise e resolução de causas dos problemas. O capítulo seguinte apresenta o *Six Sigma* e o método DMAIC no contexto da análise e resolução de causas.

3 O SIX SIGMA E O MÉTODO DMAIC

Este capítulo apresenta uma descrição sobre o *Six Sigma* e suas ferramentas, e mostra detalhes do método DMAIC.

3.1 VISÃO GERAL

Uma metodologia que está se consolidando no mundo dos negócios e possibilitando grandes conquistas para as empresas é o *Six Sigma* (também conhecido como Seis Sigma), que é um método que se concentra na diminuição ou eliminação da incidência de erros, defeitos e falhas em um processo. A metodologia *Six Sigma* visa também reduzir a variabilidade do processo, podendo ser aplicada na maioria dos setores da atividade econômica (SMITH e ADAMS, 2000).

Alcançar o *Six Sigma* significa reduzir defeitos, erros e falhas⁵ aproximadamente a zero e atingir a quase perfeição no desempenho dos processos. A metodologia associa um rigoroso enfoque estatístico a um arsenal de ferramentas que são empregadas com o objetivo de caracterizar as fontes da variabilidade do processo (WATSON, 2001).

Diferentemente do que se acredita, o *Six Sigma* não trata a qualidade no sentido tradicional, ou seja, a conformidade com normas e requisitos internos. Na verdade, o programa redefine qualidade como o valor agregado por um esforço produtivo e busca que a empresa alcance seus objetivos estratégicos (PYZDEK, 2003).

A metodologia *Six Sigma* procura definir as causas óbvias e não óbvias que afetam o processo de forma a eliminá-las, ou melhorá-las e controlá-las (ROTONDARO et al., 2002).

⁵ Na metodologia *Six Sigma*, os defeitos, os erros e as falhas são qualquer desvio de uma característica que gere insatisfação ao cliente (BLAUTH, 2003).

Algumas áreas de Tecnologia da Informação (TI) acreditam que o *Six Sigma* possa ser aplicado apenas para processos de manufatura e engenharia e que não apresenta relevância para o ciclo de vida de desenvolvimento de sistema e para a área de TI em geral. No entanto, o *Six Sigma* tem aplicabilidade para a engenharia de software (TAYNTOR, 2003).

As técnicas para garantir que os requisitos do cliente sejam entendidos, que o impacto de solicitações de mudanças seja medido e avaliado e que o processo de desenvolvimento seja executado de forma mais confiável irão beneficiar todas as áreas de TI.

O cliente percebe de forma clara, quando aquilo que recebe, varia, seja nos prazos de entrega seja nas especificações dimensionais ou de características físicas. Ele percebe, fica insatisfeito e muda de fornecedor (ROTONDARO et al., 2002). A metodologia *Six Sigma* visa também reduzir a variabilidade do processo, podendo ser aplicada na maioria dos setores da atividade econômica (SMITH e ADAMS, 2000).

3.2 HISTÓRIA DO SIX SIGMA

O *Six Sigma* nasceu no departamento de garantia da qualidade da Motorola, em 1986, fruto da aplicação dos conceitos de Deming (1990) sobre variação de processos, cujo objetivo era atacar as causas destas variações como forma de melhorá-los. Tais variações eram medidas por meio de seu *desvio padrão* (medida estatística representada pela letra grega σ) e a meta de melhoria adotada por esta empresa foi de *Six Sigma*, que representa uma variação das características de resposta do processo analisado de seis desvios padrões em torno de seu valor central, e que também foi adotada como nome desta abordagem de melhoria. Embora esta fosse uma abordagem que originalmente destinava-se à parte operacional, o presidente da Motorola, Bob Galvin, atento ao potencial dessa abordagem, logo a adaptou a todas as atividades de negócio, elegendo esta como a maneira de fazer negócio de sua empresa (ECKES, 2001).

A estratégia auxiliou a Motorola no alinhamento de seus processos às necessidades de seus clientes, na mensuração e no aprimoramento do desempenho de processos críticos e na documentação destes, levando-a a triplicar a produtividade e obter economias da ordem de US\$ 11 bilhões até o ano de 1997. O trabalho resultou na obtenção do prêmio *Malcolm Baldrige National Quality Award* (Prêmio Nacional da Qualidade dos EUA) em 1988.

Posteriormente, esta estratégia recebeu contribuições importantes de organizações como IBM, Xerox, Texas Instruments, AlliedSignal (atualmente Honeywell) e GE, onde o processo de melhoria DMAIC foi padronizado como o principal método de aprimoramento, para atingir a qualidade *Six Sigma*.

Foi criada a *Six Sigma Academy* (SSA) nos EUA para estimular sua divulgação. No Brasil, a procura pelo *Six Sigma* vem crescendo a cada dia, e desde 1998 diversas empresas estão adotando o programa com suporte de consultorias nacionais. Entre as empresas no Brasil que estão investindo no *Six Sigma*, podemos citar: Ambev, Oi, Votorantim, Johnson & Johnson, entre várias outras.

3.3 O QUE É O SIX SIGMA?

O *Six Sigma* é uma implementação rigorosa, concentrada e altamente eficaz de princípios e técnicas comprovadas de qualidade. Ao incorporar elementos do trabalho de muitos pioneiros da qualidade, essa metodologia busca o desempenho virtualmente livre de erros (PYZDEK, 2003).

Baseado em poderosas análises estatísticas e, empregado por profissionais habilitados para resolver os mais intrincados problemas, o *Six Sigma* se propõe a chegar próximo da perfeição, ou seja, 3,4 de haver defeitos para cada milhão de oportunidades.

Sigma (σ) é uma letra do alfabeto grego utilizada pelos estatísticos para mensurar a variância em qualquer série de dados. O desempenho de uma empresa pode ser medido pelo nível sigma de seus processos empresariais. Tradicionalmente, as empresas aceitavam níveis de desempenho de 3-Sigma ou 4-Sigma (o sigma representa o desvio padrão) como normais, apesar de saberem que esses processos criam entre 6,2 mil e 67 mil problemas por milhão de oportunidades. O padrão *Six Sigma*, de 3,4 problemas por milhão de oportunidades, é uma resposta ao aumento do nível de expectativa dos clientes e à crescente complexidade dos produtos e processos modernos (PYZDEK, 2003).

A metodologia *Six Sigma* pode ser definida como um sistema flexível para a liderança e o desempenho dos negócios, que possibilita o alcance de benefícios após a sua implementação (PANDE, 2001).

Pande (2001) afirma ainda que o *Six Sigma* proporciona:

- A geração de um sucesso sustentado, pois desenvolve as habilidades e a cultura necessárias a uma revitalização constante nas empresas.
- A determinação das metas de desempenho, que é a base sobre a qual está alicerçada a metodologia *Six Sigma*, em virtude do nível de desempenho aproximar-se da perfeição.
- A intensificação do valor para os clientes, considerando que o foco no cliente é o ponto vital do método, que busca compreender o significado e a importância do valor para o consumidor.
- O aprimoramento das melhorias, que é garantido pela utilização de várias ferramentas de gestão empresarial disciplinadas pela estrutura do método.
- A promoção da aprendizagem, em virtude de o *Six Sigma* aumentar o desenvolvimento e acelerar o compartilhamento de idéias inéditas dentro das empresas.
- A execução de mudanças estratégicas, pois a sua incorporação possibilita a compreensão detalhada dos processos e procedimentos das empresas, oferecendo, assim, a capacidade de implementação de simples ajustes a complexas mudanças.

Em termos tradicionais, o *Six Sigma* foca na prevenção de defeitos, na redução dos tempos de ciclo e na economia de custos. Ao contrário dos cortes de custos descuidados, que reduzem valor e qualidade, o *Six Sigma* identifica e elimina custos do desperdício, ou seja, que não agregam valor aos clientes. Em geral, esses custos são extremamente elevados em empresas que não o utilizam. Empresas que operam em níveis 3-Sigma ou 4-Sigma geralmente gastam entre 25% e 40% de suas receitas para reparar ou resolver problemas. Isso é conhecido como o custo da qualidade ou, mais precisamente, o custo da má qualidade. Empresas que operam em *Six Sigma* geralmente gastam menos de 5% de suas receitas para consertar problemas (PYZDEK, 2003).

Ao analisar, no entanto, as “Organizações *Six Sigma*”, percebe-se que não existem empresas que atingiram a qualidade no nível *Six Sigma* em todas as áreas e departamentos. O fato de usar medidas sigma ou alguma outra ferramenta não qualifica uma empresa como sendo uma “Organização *Six Sigma*”. Para tornar-se uma organização *Six Sigma* é necessário que a empresa assuma o desafio de medir e melhorar todos os seus processos, o que pode ser inviável.

Vários fatores contribuem para o sucesso do *Six Sigma*, entre os quais podem ser destacado (SNEE, 1999):

- **Liderança da direção:** à medida que os principais alvos sejam organizacionais, é necessário o envolvimento da direção na escolha de projetos, definição de metas, reuniões de revisão e acompanhamento dos indicadores.
- **Enfoque disciplinado:** utiliza-se um caminho simples e estruturado nos projetos, constituído de cinco fases: definir, medir, analisar, melhorar e controlar.
- **Rápida conclusão de projetos:** as melhorias são planejadas projeto a projeto e devem ser realizadas em curto prazo, de 3 a 6 meses.
- **Clara definição de sucesso:** existem metas muito claras disseminadas por toda a organização. Por isso é muito fácil saber se está no caminho correto ou não, e se os objetivos propostos foram alcançados ou não.
- **Infraestrutura definida:** uma condição básica de "ingresso" neste sistema é ter uma infraestrutura bem definida. É fundamental ter uma equipe disponível e treinada. Como as regras, as atividades e a hierarquia são claras, isto facilita o sucesso de implantação.
- **Enfoque estatístico:** a estatística é utilizada cientificamente na redução de fontes de falhas, que é a variabilidade, reduzindo a relevância das considerações sobre sentimentos ou opiniões. A luz da verdade predomina sobre os julgamentos pessoais.
- **Uso de ferramentas de forma integrada:** durante muito tempo os métodos para qualidade eram implantados por ondas: fase do método A, do método B, etc. Com o *Six Sigma* avaliam-se as ferramentas disponíveis e utiliza-se a mais adequada na melhoria de um dado processo. Neste contexto, preponderam os resultados sobre as ferramentas.
- **Foco nos clientes e nos processos:** existe um balanço adequado nas ações, à medida que se procura simultaneamente aprimorar processos aumentando a satisfação do cliente. Assim, evita-se escolher projetos que possam trazer ônus ao cliente.

- **Resultados rápidos:** a alocação de equipe para melhorias e foco em projetos de curto prazo, com um método adequado, cria uma condição única de trabalhar em atividades com grande potencial de retorno financeiro. O retorno financeiro nos projetos de *Six Sigma* tem demonstrado uma excepcional relação custo benefício.

Em suma, o sucesso que as empresas conseguem com a utilização da metodologia *Six Sigma* deve-se principalmente ao uso de poderosas ferramentas que, aplicadas em situações específicas e de maneira correta, proporcionam um significativo auxílio à melhoria de sistemas administrativos e de manufatura, à gestão da qualidade e ao gerenciamento de processo.

Por outro lado, algumas empresas não tiveram sucesso com a utilização do *Six Sigma*, que, de acordo com pesquisas realizadas por Eckes (2001), levou-se as seguintes causas:

- O foco do programa é orientado apenas para a redução de custos.
- As melhorias do processo não são consideradas parte integrante do trabalho usual da empresa.
- O método DMAIC não é seguido correto e disciplinadamente.
- Quando a liderança da empresa não consegue diferenciar a causa comum da causa especial de variação.
- Quando a empresa não entende e assimila o conceito e a importância do cliente.
- O papel da liderança não é compreendido, e a criação e o gerenciamento de sistemas de gestão de processos são negligenciados.

Além dos fatores de fracasso citados por Eckes (2001), Andrietta et al. (2002) destacam dois dos fatores principais que favorecem ao fracasso da implantação da metodologia *Six Sigma*. O primeiro é quando a empresa encara a metodologia como uma série de projetos, pois é um grave erro entender o *Six Sigma* como um projeto. Na verdade, os projetos devem ser compreendidos como uma alavanca tática para um programa *Six Sigma*, mas muito mais importante é o enfoque sobre a gestão do processo e a prática de uma disciplina rigorosa em todas as etapas dos processos. O segundo fator de fracasso é o descaso da liderança da empresa, que deve estar comprometida ativamente com todas as etapas da implantação e condução da metodologia, e não apenas com a alocação de recursos para o programa. O envolvimento da liderança no *Six Sigma* é um fator determinante no

gerenciamento de sistemas de gestão de processos, e mais importante do que a criação de projetos táticos da estratégia das empresas.

Por se tratar de um programa de melhoria da qualidade na qual a capacitação das pessoas que estão ligadas aos processos da organização é de fundamental importância na aplicação dos conhecimentos complexos em que se baseia o *Six Sigma*, somente empresas que possuam um grau de maturidade suficientemente alto, o que inclui uma elevada preocupação com a capacitação de seus colaboradores, consegue empregá-lo com uma maior probabilidade de sucesso (CABRERA, 2006).

3.3.1 Papéis e Responsabilidades

Para Rotondaro et al. (2002), a constituição da equipe *Six Sigma* é um elemento fundamental para o sucesso do programa.

A metodologia *Six Sigma* criou uma organização própria com papéis chaves para seu desenvolvimento, tomando como base a graduação utilizada nas artes marciais. Tal estrutura foi criada para fortalecer a disseminação da cultura dentro da organização e buscar o comprometimento em todos os níveis. Os títulos dados são de *Sponsor*, *Champion*, *Master Black Belt* (MBB), *Black Belt* (BB) e *Green Belt* (GB). Existem outras empresas que acrescentaram os títulos *Yellow Belt* (YB) e *White Belt* (WB) aos termos já citados. Isto se deve à forte tendência das organizações em personalizar e criar mais níveis de profissionais em seus programas *Six Sigma*. A função desempenhada por cada um destes títulos está descrita a seguir (HARRY e SCHROEDER, 2000):

- ***Sponsor (Patrocinador)*** – podem ser líderes informais que utilizam o *Six Sigma* em seu trabalho diário e comunicam suas mensagens em todas as oportunidades. São os donos dos processos e sistemas que ajudam a iniciar e coordenar as atividades de melhoria *Six Sigma* nas áreas pelas quais são responsáveis.
- ***Champion*** – representante da alta administração que deve demonstrar o alto comprometimento da empresa com o processo de implementação. É uma função característica de empresas grandes com várias divisões. Deve difundir a cultura do *Six Sigma* e garantir o comprometimento do pessoal ao andamento dos projetos. Avalia periodicamente o programa.

- **Master Black Belt (MBB)** – especialista em estatística provido de alto conhecimento do próprio negócio, que oferece consultoria interna aos *Black Belts* e multiplicação dos conceitos dentro da organização. Esta também é uma função típica de empresas de grande porte.
- **Black Belt (BB)** – é o líder de grandes projetos dentro de uma área da empresa. Função normalmente exercita por funcionários experientes e com visão global da empresa. Normalmente, dedica-se 100% do seu tempo ao *Six Sigma*. Com isto a empresa dispõe de funcionários altamente treinados e dedicados integralmente para a melhoria contínua dos seus processos.
- **Green Belt** – membro das equipes multifuncionais, com dedicação geralmente não integral aos projetos. São funcionários especializados em suas tarefas, com um grau menor de conhecimento.
- **Yellow Belt** – membro de equipe de projeto BB e GB. Tem por objetivo contribuir para o êxito dos projetos e também para a disseminação da cultura da metodologia *Six Sigma* por toda a empresa.
- **White Belt** – são os demais membros da organização, que recebem o conhecimento básico sobre a metodologia para poder prover informações e contribuir com o sucesso dos projetos.

Os projetos são conduzidos pelos líderes (*Black Belts*) em equipes multifuncionais trabalhando sob estrutura matricial.

3.3.2 Ferramentas e Metodologias

As ferramentas e técnicas utilizadas pela metodologia *Six Sigma* não são novidades no contexto da qualidade, mas a forma que este programa estrutura e foca na redução da variabilidade dos processos faz desta uma metodologia bastante eficiente e diferenciada (CABRERA, 2006).

O conhecimento prévio de técnicas e ferramentas é necessário para se determinar qual destas é a mais apropriada, ressaltando que a aplicação apropriada de ferramentas é crítica para a eficácia do projeto *Six Sigma* (CABRERA, 2006).

O *Six Sigma* tem como uma de suas principais características a forte estruturação, seqüência e uso das ferramentas apropriadas em cada etapa, o que também garante uma forte taxa de sucesso nos projetos, quando seguidas.

Entre as ferramentas mais usadas destacam-se:

- O Controle Estatístico de Processos (CEP) e os Gráficos de Controle, para a identificação de problemas;
- Os diagramas SIPOC (*Supplier, Input, Process, Output, Customer*) e os gráficos de causa-efeito para definição do problema e análise da causa-raiz;
- O Planejamento de Experimentos (*Design of Experiment - DOE*), para análise de soluções ótimas e validação de resultados;
- A Análise de Modo e Efeitos de Falha (*Failure Mode and Effect Analysis - FMEA*), para priorização e prevenção de problemas;
- O Desdobramento da Função da Qualidade (*Quality Function Deployment - QFD*), para produtos, serviços e projeto de processos.

Segundo Pande (2001), quais ferramentas usar, quando e por quê, é um dos maiores desafios das organizações e das equipes que se envolvem em um esforço *Six Sigma*. Essas ferramentas são utilizadas pelos métodos disponíveis no *Six Sigma*, de acordo com as características de cada fase.

O *Six Sigma* apresenta dois possíveis métodos: DMAIC⁶ (*Define, Measure, Analyse, Improve and Control*) e DCOV⁷ (*Define, Characterize, Optimize, Verify*). O primeiro é para tratar problemas existentes e a segunda para prevenir a ocorrência de problemas. Também apresenta outras variações, como o DFSS⁸ (*Design For Six Sigma*), uma extensão do *Six*

6 O método DMAIC é usado como a abordagem padrão para a condução dos projetos *Six Sigma* de melhoria de desempenho de produtos e processos.

7 O DCOV é um método do *Six Sigma* que oferece um quadro de avaliação objetiva e medição das atividades fundamentais para o desenvolvimento de uma combinação produto.

8 O DFSS é uma metodologia baseada em ferramentas analíticas que fornecem aos usuários a capacidade de prevenir e prever defeitos no projeto de um novo produto, serviço ou processo.

Sigma para o projeto de novos produtos e processos, que utiliza o método DMADV⁹ (*Define, Measure, Analyse, Design, Verify*) para assegurar a repetição e melhoria contínua, e o *Lean Six Sigma*, que constitui a integração da estratégia *Lean*¹⁰ com o *Six Sigma* visando a eliminação de desperdícios e de causas de defeitos nos processos administrativos, processos de produção ou processos de manufatura.

Portanto, seja qual for o modelo adotado, é essencial que os projetos de melhoria sejam conduzidos com disciplina e organização. Esses modelos são vistos, inicialmente, como verdadeiros guias na solução de problemas e implementação de melhorias.

O método que será detalhado a seguir, o DMAIC, é funcional, mostrando claramente o caminho a ser seguido após um projeto ser escolhido. Esse é o método utilizado neste trabalho, que foi criado de forma mais padronizada pela General Electric e, de acordo com Tayntor (2003), é o mais utilizado nas empresas que implementam o *Six Sigma*, assim como, é o mais indicado para o desenvolvimento de software.

3.4 O MÉTODO DMAIC

Muitos modelos de melhorias têm como referência o ciclo do PDCA (*Plan, Do, Check, Act*), originalmente concebido por Deming. A filosofia desse ciclo é sua aplicação contínua, ou seja, a última etapa de um ciclo determina o início de um novo ciclo. Na estratégia *Six Sigma*, o ciclo DMAIC tem as mesmas características.

O DMAIC é um método de qualidade baseado em dados para melhoria de processos, reconhecido em todo o mundo como o meio de se estruturar os projetos de melhoria na busca do padrão *Six Sigma*. As empresas têm utilizado esta ferramenta como a principal estrutura para que o CEP (Controle Estatístico de Processos) determine os pontos de oportunidade de melhoria nos processos estudados e possibilitem a aplicação da estratégia com a finalidade de elevar o nível Sigma.

Em sua totalidade, o método DMAIC consiste em definir, medir, analisar, melhorar e controlar, conforme representado na Figura 2.1. Na fase “**definir**”, é necessário identificar o

9 O DMADV é o método para implantação do *Design for Six Sigma* (DFSS), sendo utilizado em projetos cujo escopo é o desenvolvimento de novos produtos e processos.

10 Filosofia que busca minimizar o capital de giro necessário para produzir um produto ou fornecer um serviço.

problema e então definir as oportunidades existentes para solucioná-lo de acordo com os requisitos dos clientes. Na fase “**medir**”, deve-se verificar a situação atual através de medições quantitativas do desempenho, para que as decisões posteriores sejam baseadas em fatos. Na fase “**analisar**”, deve-se determinar as causas do desempenho encontrado e analisar as oportunidades existentes. Após ter feito essa análise é possível perceber pontos para melhoria do desempenho, assim as melhorias podem ser implantadas na fase “**melhorar**”. Na fase “**controlar**”, deve-se assegurar a melhoria, através de um controle do desempenho do processo implantado.



Figura 3.1: Representação das Fases do DMAIC.

Antes que seja iniciada a primeira fase, a de definição, o projeto de melhoria deve ser selecionado, sendo um dos pontos mais importantes da metodologia *Six Sigma*. A seleção correta de um projeto de melhoria pode ser extremamente significativa no negócio, uma vez que seu processo pode tornar-se mais eficiente em um horizonte de tempo de três a seis meses. Conseqüentemente, seus funcionários ficarão satisfeitos em ver resultados financeiros advindos destes esforços (CABRERA, 2006).

Para Rotondaro et al. (2002), é fundamental que na seleção do projeto haja uma relação clara com um requisito especificado do cliente e que o projeto seja economicamente vantajoso. Ainda assim, Pande (2001) ressalta que não se pode utilizar o DMAIC para qualquer coisa. Um projeto de melhoria *Six Sigma*, segundo o autor deve ter as seguintes características:

- Há uma lacuna entre o desempenho atual e o desempenho necessário/desejado.
- A causa do problema não é claramente compreendida.
- A solução não é predeterminada, nem é a solução ótima aparente.

Além disso, devem ser observados os critérios de viabilidade, tais como: os recursos necessários, a habilidade disponível, a complexidade, a probabilidade de sucesso e o apoio ou engajamento, e os critérios de impacto organizacional, tais como: os benefícios da aprendizagem, os benefícios entre as áreas, ou seja, se o projeto vai poder ajudar a quebrar barreiras entre grupos na organização e melhorar a gerência de todo o processo.

Embora cada projeto tenha o seu próprio tempo de realização, os primeiros projetos dentro da metodologia *Six Sigma* devem ser planejados para 120 a 160 dias de duração. Conforme relata Eckes (2001), quando este prazo ultrapassa os 160 dias a probabilidade de atingir as metas diminui. Ele afirma, ainda, que cerca de metade deste tempo deve ser dedicado às atividades de definição e mensuração, enquanto que a outra metade deve ser dedicada à análise e melhoria.

As etapas do método DMAIC serão descritas de forma mais detalhada a seguir, de acordo com os passos definidos por Tayntor (2003). A Tabela 3.1 apresenta um sumário das fases e passos do DMAIC.

Tabela 3.1: Sumário das Fases e Passos do DMAIC (TAYNTOR, 2003).

Fase	DMAIC (Passos)
Definir	Definir o Problema
	Formar a Equipe
	Estabelecer um <i>Project Charter</i>
	Desenvolver o Plano do Projeto
	Identificar os Clientes
	Identificar os Artefatos Resultantes
	Identificar e Priorizar os Requisitos do Cliente
Medir	Determinar o que Medir
	Conduzir as Medições
	Calcular o Nível Sigma Atual
	Determinar a Capacidade do Processo
	<i>Benchmark</i> dos líderes do Processo
Analisar	Determinar as Causas de Variação
	<i>Brainstorming</i> de Idéias para Melhoria de Processo
	Determinar as Melhorias que tem Maior Impacto nos Requisitos do Cliente
	Desenvolver o Mapa do Processo Proposto

	Avaliar os Riscos Associados com o Processo Revisado
Melhorar	Obter a Aprovação das Mudanças Propostas
	Finalizar o Plano de Implementação
	Implementar as Mudanças Aprovadas
Controlar	Estabelecer as Métricas Chaves
	Desenvolver a Estratégia de Controle
	Celebrar e Comunicar o Sucesso
	Implementar o Plano de Controle
	Medir e Comunicar as Melhorias

3.4.1 Definir

A fase *definir* consiste em delimitar as metas das atividades de melhoria com os problemas e as situações a serem melhoradas. A definição deve ser específica, mensurável, descrever o impacto sobre o negócio ou a atividade da empresa e não concluir prematuramente as causas e soluções. Nessa fase, a equipe do projeto é formada, o termo de abertura do projeto (*Project charter*) é criado, necessidades e requisitos do cliente são determinados e verificados, e um mapa de alto nível do processo corrente é criado.

As principais ferramentas e técnicas que têm sido utilizadas nesta fase são:

- **QFD (*Quality Function Deployment*)** – é uma técnica para dar prioridades aos comentários e requisitos dos clientes e traduzi-los em projetos e especificações de um produto, serviço e/ou processo (ALLEN, 2006).
- **Voz do Cliente (VoC – *Voice of Customer*)** – consiste em identificar quais as necessidades e expectativas do cliente ouvindo destes tais definições (DENOVE e POWER, 2006).
- **SIPOC (*Supplier, Input, Process, Output, Customer*)** – é uma técnica utilizada para entender melhor algumas relações existentes no processo desde o fornecedor até o atendimento do cliente (GUPTA, 2006).

Os passos dessa fase serão apresentados a seguir.

3.4.1.1 Definir o Problema

É muito comum que se inicie a tentativa de solução de um problema sem uma definição exata de sua real importância e o impacto na qualidade do produto ou serviço prestado. No entanto, esta atividade é primordial uma vez que a definição do problema determina a questão tática que o grupo pretende melhorar.

Segundo Eckes (2001), deve ser feita uma descrição do problema citando desde quando o mesmo existe. O problema deve ser específico, atingível e mensurável, pois isso fornece a magnitude do problema e, além disso, é necessária a descrição do seu impacto ou consequências sobre os negócios.

Eckes (2001) afirma também que a partir daí, metas e objetivos factíveis devem ser determinados (da definição do problema) e acordados entre os membros da equipe e seu líder. As metas devem sempre ser estabelecidas sobre os resultados (fins) e nunca sobre os meios de um processo.

3.4.1.2 Formar a Equipe

Dentro dos projetos de melhoria, a distribuição dos papéis dentro da equipe deve ser realizada cuidadosamente. Não devem ser selecionadas simplesmente as pessoas disponíveis e interessadas no projeto, mas aquelas mais qualificadas para o trabalho e que tenham impacto direto sobre os objetivos estratégicos do projeto em questão (ECKES, 2001).

De acordo com Tayntor (2003), obtém-se maior probabilidade de sucesso ao envolver todos os *stakeholders* importantes; portanto, deve-se juntar a equipe correta. Além disso, é de suma importância o forte comprometimento da alta direção.

3.4.1.3 Estabelecer o Project Charter

A primeira tarefa de maior importância da equipe é o desenvolvimento do termo de abertura do projeto (PMI, 2008), que é um dos documentos mais importantes em um projeto usando a metodologia *Six Sigma*, pois dá uma visão geral do projeto e serve como acordo entre a direção e a equipe *Six Sigma* no que se refere aos resultados esperados do projeto.

O *Project charter* é usado para estabelecer a direção do projeto e definir as medidas de sucesso. Isso inclui uma clara definição das funções e responsabilidades da equipe. Além disso, no *Project charter* são identificados todos os atores envolvidos no projeto com seus devidos papéis, objetivos e metas, a razão de ser do projeto ou justificativa, assim como

premissas, orçamento, entre outras informações relevantes. Aprovando o documento, o *champion* do projeto e o dono do processo se comprometem com o projeto e concordam em garantir que a organização concederá os recursos apropriados.

3.4.1.4 Desenvolver o Plano do Projeto

No desenvolvimento do plano do projeto, é estabelecido um cronograma contendo as atividades que devem ser executadas em cada fase do ciclo DMAIC, compostas por várias tarefas menores com seus respectivos responsáveis e prazos. Neste passo, também são desenvolvidos planos de comunicação, riscos, qualidade, entre outros.

3.4.1.5 Identificar os Clientes

Para Deming (1990), o cliente é o que há de mais importante. Devem-se cultivar relações de longo prazo com os clientes e compreender continuamente as necessidades deles ao projetar e construir produtos.

De acordo com Eckes (2001), o cliente é o destinatário do produto ou serviço. Com essa definição em mente (de quem é o cliente) parte-se para a determinação de necessidades e requisitos. Deve-se lembrar que as necessidades do cliente são os resultados de um processo que estabelece a relação entre o consumidor e o fornecedor e os requisitos são as características que determinam se o cliente está contente com o resultado.

Segundo Pande (2001), para medir ou alcançar o *Six Sigma* uma compreensão clara das necessidades dos clientes e atenção dada a elas são obrigatórias, já que o desempenho *sigma* baseia-se na definição do cliente.

3.4.1.6 Identificar os Artefatos Resultantes

Além de considerar o produto e os artefatos de saída mais importantes e tangíveis, devem-se considerar os produtos e serviços que podem ser usados para avaliar o desenvolvimento do produto. Artefatos de saída não são apenas os tangíveis, mas também os intangíveis, como velocidade de processamento ou acurácia de informações (TAYNTOR, 2003).

É importante manter atualizada durante a execução do projeto a documentação das atividades realizadas. Essa documentação deve ficar acessível aos participantes da equipe e, eventualmente, às áreas técnicas da empresa.

3.4.1.7 Identificar e Priorizar os Requisitos do Cliente

Entender os requisitos de forma clara antes de realizar qualquer mudança no processo é muito importante não é apenas pelo princípio de focalizar no cliente, mas também para se tomar decisões baseadas em fatos.

O ponto de vista do cliente é uma das premissas básicas para a análise e o direcionamento de todas as atividades que a equipe de trabalho executará na fase de definição. Identificar os parâmetros relevantes para o cliente são entradas de informações importantes para a atividade de mapeamento dos processos. Informações como o tempo em que o cliente final considera como aceitável para a entrega de um produto e o que significa qualidade do produto na visão do cliente, são parâmetros relevantes para os processos.

Os requisitos do cliente devem ser específicos e facilmente medidos, relacionados diretamente a um atributo ou serviço e descrever qual é a necessidade, e não como ela será implementada (GEORGE et al., 2005).

3.4.1.8 Documentar o Processo Atual

Apesar de existirem várias formas de documentar um processo, a maioria das organizações *Six Sigma* utiliza mapas de processo ao invés de simples descrição. Um mapa de processo oferece uma representação pictórica do processo analisado, mostrando uma sequência de tarefas, além de suas entradas e saídas (TAYNTOR, 2003).

O mapeamento dos processos é uma das atividades chave da fase de definição, permitindo assim que, através do resultado do mapeamento, seja possível identificar áreas de retrabalho, atividades que não agregam valor ao resultado final, atividades que estão fora dos padrões de mercado no quesito tempo e uso de recurso, e também problemas com recursos humanos específicos dentro de um processo (STAMATIS, 2004).

O desenvolvimento de mapas de processo é um processo tipicamente iterativo. A equipe desenvolve um mapa inicial, que em seguida é revisado por pessoas chaves que entendem o processo para avaliá-lo (TAYNTOR, 2003).

O processo sob a perspectiva de modelagem estatística pode também ser interpretado como um conjunto de fatores (causas) que geram uma ou mais respostas (efeitos), podendo ser afetadas por fatores não controláveis (ruído) (ROTONDARO et al., 2002).

3.4.2 Medir

A fase *medir* consiste em mensurar o sistema existente, estabelecendo métricas válidas e confiáveis para ajudar a monitorar o progresso rumo às metas definidas no passo anterior. A medição impõe a determinação de uma medida básica da eficiência do processo e deve ter um plano de coleta de dados, que possa assegurar que a amostragem seja representativa e aleatória. Nessa fase, o nível corrente do sigma é criado.

Para Pande (2001), embora possa exigir mais esforço, a maioria das coisas que acontecem em uma empresa pode ser medida, e o requisito número um para a medição é a capacidade de “observar”. A medição consome recursos, atenção e energia, o que significa que não se deseja realizar qualquer medição que não seja necessária.

O que diferencia a metodologia *Six Sigma* de tantos outros programas de melhoria da qualidade que vêm sendo implantados em empresas, nas últimas décadas, é a ênfase na tomada de decisões baseadas em dados e fatos e não nas experiências individuais. O que é medido é realizado, por isso, Eckes (2001) afirma que se uma pessoa não realiza medições e faz afirmações a respeito de um processo, apenas é uma pessoa com uma opinião em particular.

As principais ferramentas e técnicas utilizadas nesta fase são:

- **Histograma** – é uma representação visual dos dados que nos possibilita enxergar três propriedades do conjunto de dados do processo: a forma, a tendência central e a dispersão (MONTGOMERY, 2004).
- **FMEA (*Failure Mode and Effect Analysis*)** – técnica de análise de produtos ou de processos, utilizada para identificar possíveis modos de falha, determinando o efeito de cada uma sobre o desempenho do produto ou processo (PALADY, 1997).
- **Análise de Variância (Anova - *Analise of Variance*)** – é um teste estatístico que visa verificar se existe uma diferença significativa entre as médias e se os fatores exercem influência em alguma variável dependente. Permite a comparação de médias oriundas de grupos diferentes, também chamados tratamentos, como por exemplo médias históricas de questões de satisfação, empresas que operam simultaneamente com diferentes rendimentos, entre muitas outras aplicações. (VIEIRA, 2006).

Os passos dessa fase serão apresentados a seguir.

3.4.2.1 Determinar o que Medir

As medições podem auxiliar a direcionar a maior parte dos problemas críticos em desenvolvimento de software além de prover suporte para se planejar, monitorar, controlar e validar processos de software, respondendo às mudanças de forma efetiva.

Para Pande (2001), é de suma importância a existência de um processo para a medição que envolva a seleção do que medir, o desenvolvimento das definições operacionais, a identificação da fonte de dados, preparação do plano de coletas e amostragem e, finalmente, a implementação e o refinamento da medição. Vale salientar que é importante compreender a diferença entre medidas contínuas e discretas, já que isso pode impactar não só em como definir as medidas, mas também como os dados serão coletados e aquilo que se pode aprender com eles.

Ainda segundo Pande (2001), além do treinamento, a medição é provavelmente o maior “investimento” que qualquer organização pode fazer em sua iniciativa *Six Sigma*. O desenvolvimento em longo prazo de uma “infraestrutura” de medição, no entanto, é uma das pedras fundamentais de um sistema organizacional pleno *Six Sigma*.

Dada a importância da medição no *Six Sigma*, é imprescindível a geração de fatos e dados confiáveis alinhados à estratégia organizacional. Não se trata, portanto, em “medir por medir”, mas sim em medir o que é relevante. É através da análise de fatos e dados que se consegue transformar informação em conhecimento para tomada de decisões e alcance de metas. Porém, esses dados precisam ser confiáveis para que as conclusões sejam válidas. Portanto, o primeiro passo para qualquer estudo, é questionar a confiabilidade de toda e qualquer informação utilizada (fatos e dados).

Sendo assim, o processo deve ser consistente e assegurar que as informações coletadas estejam íntegras. Os procedimentos devem estar bem descritos, pois o mesmo procedimento deve ser aplicado outras vezes para acompanhamento do desempenho.

3.4.2.2 Conduzir as Medições

Usando a lista de elementos a serem medidos, inicia-se a coleta de dados. Neste passo, deve-se coletar a maior quantidade de dados e informações sobre o problema. Cálculos são realizados para determinar quão próximo o processo atual está dos requisitos do cliente

identificados na fase de definição. As medições obtidas devem ser estudadas para entender as possíveis variações existentes, como forma de não tomar decisões apenas em cima de médias, lembrando que seus objetivos são eliminar defeitos e qualquer variação pode ser vista como defeito (TAYNTOR, 2003).

Sendo assim, é importante que a equipe de trabalho esteja ciente sobre os motivos pelos quais as variações podem ocorrer durante as medições. Todos os processos apresentam variabilidade, ou seja, as características de um produto apresentam variações inevitáveis, devido a variações sofridas pelos fatores que compõem um processo produtivo. Haverá sempre a variabilidade no produto; portanto, é importante que essa variabilidade seja controlada.

3.4.2.3 Calcular o Nível Sigma Atual

A forma de cálculo da escala sigma é muito subjetiva. Pode-se alterar a escala sigma de seu processo sem necessariamente aumentar a sua qualidade. Para isto, basta que a fórmula de cálculo seja alterada, aumentando o número de oportunidades de defeitos no processo (HARRY e SCHROEDER, 2000).

Como o nível sigma é uma métrica de defeitos, é importante ratificar a definição de termos chaves antes de iniciar o cálculo:

- **Defeito:** falha em relação à especificação.
- **Unidade:** menor medição de artefatos de saída.
- **Defeito por Unidade (DPU):** número total de defeitos em uma amostra, dividido pelo número total de unidades na mesma.
- **Defeituosa:** unidade com um ou mais defeitos, identificado no final do processo.
- **Oportunidade:** chance de criar um defeito em uma única unidade.
- **Defeitos por Milhão de Oportunidades (DPMO):** métrica mais importante para determinar o nível sigma.

Para calcular o nível sigma do processo, a equipe calcula o DPMO de acordo com os passos da Tabela 3.2 (TAYNTOR, 2003).

Tabela 3.2: Ações para Cálculo do Nível Sigma

Passo	Ação
1	Contar o número de unidades processadas
2	Contar o número de unidades com defeitos (defeituosas)
3	Calcular a taxa de defeitos (passo 2 / passo 1)
4	Contar oportunidades de defeitos em cada unidade
5	Calcular os defeitos por oportunidade (passo 3 / passo 4)
6	Calcular DPMO (passo 5 x 1.000.000)
7	Converter DPMO em um valor sigma, usando tabela de conversão

3.4.2.4 Determinar a Capacidade do Processo

Segundo Carvalho (2002), o controle estatístico tem por objetivo conhecer a estabilidade do processo estudado, monitorando seus parâmetros ao longo do tempo. Para o autor, a capacidade do processo já observa outra questão: para um processo cujo comportamento seja conhecido, pode-se dizer se ele é capaz de produzir itens ou prestar o serviço segundo as especificações determinadas pelo cliente.

Carvalho (2002) destaca que, embora a restrição da distribuição seja normal para os dados, não há referências nos trabalhos dos principais autores quanto à necessidade do processo estar sob controle, para que o estudo da capacidade possa ser feito. Os cálculos necessitam ser feitos em todos os processos que estão sendo acompanhados por meio da estratégia *Six Sigma*.

3.4.2.5 Benchmarking dos Líderes do Processo

Algumas organizações consideram este passo apenas na fase de análise. Uma vez que a equipe entendeu o processo, pode-se realizar a atividade de *benchmarking*. O objetivo é identificar empresas que desempenham o processo da melhor forma, quantificar desempenho dos líderes e aprender o que diferencia seus líderes de seus competidores (ROTONDARO et al., 2002).

O *Six Sigma* é usado como parâmetro para comparar o nível de qualidade de processos, operações, produtos, características, equipamentos, máquinas, divisões e departamentos, entre outros.

Para ilustrar o “*Six Sigma – Benchmarking*”, considere um processo com um nível de qualidade de aproximadamente quatro sigmas, ou seja, 6.210 defeitos por milhão de oportunidades (DPMO). Neste caso, pode ser feito um estudo *benchmarking* na busca de uma referência de qual o nível de qualidade é encontrado em processos similares dos concorrentes, ou internamente em outras unidades da empresa. Caso a qualidade esteja abaixo da média do mercado, projetos de melhoria devem ser desenvolvidos rapidamente em busca de novos padrões de qualidade. Em alguns casos, pode ser encontrado o que se chama de *benchmarking* interno, onde o nível mais alto encontrado para um processo similar pertence a uma outra unidade da própria empresa.

Vale ressaltar que fazer *benchmarking* deve ser uma prática contínua em uma empresa que busca uma qualidade *Six Sigma*.

3.4.3 Analisar

A fase *analisar* consiste em examinar as informações obtidas. Para Pande (2001), essa é a fase mais importante do ciclo DMAIC, pois determina e valida a raiz do problema original, que é o alvo da busca de melhoria. Nela, a equipe analisa os dados e o próprio processo com o intuito de determinar as causas raiz do desempenho pobre do sigma do processo.

Segundo Eckes (2001), muitas vezes, porém, a equipe tem uma noção preconcebida da razão do problema, o que a faz passar pela fase da análise de forma superficial, dando soluções precipitadas de melhoria.

Após a primeira fase do processo de melhoria, onde todas as idéias factíveis para a identificação da raiz do problema são geradas, ocorre o afinilamento, que vai então gerar um rol das causas potenciais mais prováveis e, na última fase, a equipe reforça o afinilamento para uma, duas ou três causas mais prováveis apenas.

As principais ferramentas e técnicas utilizadas nesta fase são:

- **Diagrama de Causa-Efeito** – também conhecido como diagrama de Ishikawa ou Diagrama de Espinha de Peixe, consiste em uma representação gráfica que mostra

a relação entre um conjunto de causas/subcausas que provocam um determinado efeito/problema. Desta forma, ilustra claramente as várias causas que afetam um processo por classificação e relação das causas (ISHIKAWA, 1998).

- **Análise de regressão e correlação:** compreendem a análise de dados amostrais para saber se e como um certo conjunto de variáveis está relacionado com outra variável (JOGLEKAR, 2003).
- **DOE (*Design of Experiment*)** – é uma técnica utilizada para se planejar experimentos, ou seja, para definir quais dados, em que quantidade e em que condições devem ser coletados durante um determinado experimento, buscando, basicamente, satisfazer dois grandes objetivos: a maior precisão estatística possível na resposta e o menor custo (WERKEMA, 1996).

Os passos dessa fase serão apresentados a seguir.

3.4.3.1 Determinar as Causas da Variação

Não há como imaginar situações em que não haja variação, pois esta é parte intrínseca do cotidiano. Por isso, a variação está sempre presente no ambiente, sejam entre produtos, pessoas, serviços, processos, organizações, natureza, etc. Evidentemente que, em qualquer ambiente produtivo, a presença da variação pode decorrer por diferentes causas, algumas delas comuns, outras não.

As causas comuns são as variações inerentes a um processo e estarão presentes mesmo que todas as operações sejam efetuadas com métodos padronizados. Determinam a sua “variabilidade característica” e, geralmente, vêm de várias fontes de pequenas variações. A eliminação dessas variações é mais complexa e requer o conhecimento e a análise de todo o processo e mudanças estruturais – procedimentos, pessoas, equipamento etc.

Quando apenas as causas comuns estão atuando em um processo, a quantidade de variabilidade se mantém em uma faixa estável, conhecida como faixa característica do processo. Neste caso, diz-se que o processo está sob controle estatístico, apresentando um comportamento estável e previsível.

As causas especiais de variação são variações que surgem ocasionalmente no processo e, em geral, a eliminação destas está ao alcance diretamente envolvido na execução das atividades. Uma vez identificada uma causa especial, deve-se prevenir a sua reincidência por

meio de uma ação preventiva. Quando um processo está operando sob atuação de causas especiais, diz-se que ele está fora de controle estatístico e, neste caso, sua variabilidade é bem maior do que a variabilidade natural.

Uma vez identificado que o processo está fora de controle, seus problemas devem ser tratados e deve ser identificada a causa raiz do mesmo. Normalmente, ocupa-se uma grande parte de tempo resolvendo problemas recorrentes, porque, muitas vezes, não se atua na causa-raiz do problema, mas em seus efeitos.

Existem várias formas de identificar a verdadeira causa-raiz dos problemas, desde a utilização de métodos estatísticos, ao questionamento sistemático e recorrente do porquê da origem do problema. Neste passo, deve-se selecionar uma ferramenta adequada para investigação da causa raiz do problema. Quando se conhece a raiz do problema, pode-se implantar uma solução definitiva, que, às vezes, pode ser simples, ou mais complexa na resolução do problema em questão.

3.4.3.2 *Brainstorming* de Idéias para a Melhoria de Processo

Após a equipe possuir uma lista completa de causas de variação no processo, membros começam a considerar formas de eliminar essa variação, realizando um *brainstorming* para gerar idéias, além de classificá-las em termos de custo e tempo necessário para implementar cada mudança (TAYNTOR, 2003).

3.4.3.3 Determinar as Melhorias que têm Maior Impacto nos Requisitos do Cliente

A partir de um conjunto de idéias para melhoria de processos, é feita uma análise sobre quais dessas idéias têm um maior impacto nos requisitos do cliente. Muitas vezes, são consideradas as melhorias que foram sugeridas e que apresentem menor custo e tempo; porém, sem se preocupar se as mesmas iriam satisfazer os requisitos do cliente. Isto não é um procedimento correto.

O custo benefício de uma mudança deve ser analisado criteriosamente, a fim de se concluir se o impacto da mudança é realmente benéfico tanto em redução de custos como em benefícios para os clientes da organização, considerando também o tempo necessário para implementar cada mudança.

3.4.3.4 Desenvolver o Mapa do Processo Proposto

Consiste do desenvolvimento de um mapa de processo para o processo revisado. O mapa representa uma análise de melhorias que devem e podem ser implementadas (processo revisado) (TAYNTOR, 2003).

3.4.3.5 Avaliar os Riscos associados com o Processo Revisado

A proposta de mudança não estará completa até que os riscos associados com o processo revisado sejam avaliados. Esta avaliação proporciona uma maior segurança na implementação das mudanças nos processos.

3.4.4 Melhorar

A fase *melhorar* consiste na soma das atividades relacionadas com a geração, seleção e implementação de soluções. A melhoria é conseguida com soluções que eliminem, atenuem ou minimizem as causas do problema. Nessa fase, a equipe gera e seleciona um conjunto de soluções para melhorar o desempenho sigma.

As principais ferramentas e técnicas utilizadas nesta fase são:

- **Pareto** – consiste em um gráfico de barras que ordena as frequências das ocorrências, da maior para a menor, permitindo a priorização dos problemas. Sua maior utilidade é a de permitir uma fácil visualização e identificação das causas ou problemas mais importantes, possibilitando a concentração de esforços sobre os mesmos (ALLEN, 2006).
- **Análise de Custo Benefício** – consiste numa técnica onde se comparam os custos de implementação das possíveis estratégias, com os benefícios que as mesmas geram. Ou seja, procura a melhor razão entre os benefícios e os custos (BREYFOGLE, 2003).
- **Matriz de Priorização** – é uma ferramenta para tomada de decisão que estabelece prioridades, que podem ou não ser baseadas em critérios com pesos definidos (FLORAC e CARLETON, 1999).

Os passos dessa fase serão apresentados a seguir.

3.4.4.1 Obter a Aprovação das Mudanças Propostas

Realiza-se a preparação final das mudanças propostas para aprovação. Pode-se fazer um documento com a avaliação de impacto, resumindo efeitos positivos e negativos propostos para a melhoria do processo (TAYNTOR, 2003).

3.4.4.2 Finalizar o Plano de Implementação

Deve-se desenvolver um plano simplificado para implementação das melhorias propostas. Ao invés de apenas desenvolver um plano geral do projeto, sugere-se fazer uma versão simplificada de alto nível, para ser usado em comunicações com *stakeholders* (TAYNTOR, 2003).

Neste passo são definidas as tarefas que serão executadas, seus responsáveis e prazos.

3.4.4.3 Implementar as Mudanças Aprovadas

Além do desenvolvimento e da implementação das soluções pela equipe, deve-se certificar de que todos os envolvidos entendam as mudanças que estão ocorrendo. Ao final deste passo, deve-se analisar se as metas definidas foram alcançadas.

Desenvolver e implementar as mudanças é o passo mais longo da fase de melhoria. Quando esta é completada, passa-se para a fase seguinte do modelo, a de controle (TAYNTOR, 2003).

3.4.5 Controlar

A fase *controlar* difere de outras iniciativas de qualidade, que se encerram no momento em que as soluções são implementadas. O *Six Sigma* reconhece a necessidade de uma fase adicional, conhecida como fase de controle, que consiste em controlar os processos aperfeiçoados, a fim de gerar um ciclo de melhoria contínua, ou seja, é a ação de garantir que as melhorias se sustentem ao longo do tempo (TAYNTOR, 2003).

Detectar uma mudança no comportamento do processo, o mais rapidamente possível, pode fazer com que ações corretivas adequadas sejam disparadas, e o processo, corrigido a tempo de evitar surpresas (ROTONDARO et al., 2002).

O controle deve acontecer nos níveis táticos e estratégicos, sendo garantido por meio de soluções e gráficos estatísticos, que alertem quando acontecerem mudanças ou surgirem

defeitos no processo. Nesta etapa, um conjunto de ferramentas e técnicas são aplicadas no processo melhorado, de modo que o desempenho melhorado do sigma mantenha-se.

As principais ferramentas e técnicas utilizadas nesta fase são:

- **Plano de Institucionalização** – consiste em um planejamento realizado para mostrar como as melhorias serão implementadas pela organização (CHRISSIS et al., 2006).
- **Cartas de Controle** – também conhecidas como gráficos de controle, mostram se o processo está ou não sob controle estatístico, sendo útil para o monitoramento da variabilidade e para a avaliação da estabilidade de um processo (MONTGOMERY, 2004).
- **Avaliação do Sistema de Medição** – consiste em verificar a adequação do sistema de medição da organização (FLORAC e CARLETON, 1999).

Os passos dessa fase são apresentados a seguir.

3.4.5.1 Estabelecer as Métricas Chaves

Na fase *medir*, foram conduzidas medições; porém, não foram desenvolvidas métricas. Neste contexto, uma medição possui uma única dimensão, capacidade, ou quantidade, enquanto uma métrica é um valor calculado a partir de várias medições. As métricas precisam ser simples de entender, objetivas, ter custo razoável e serem informativas (HALL e FELTON, 1997).

A seleção das métricas a serem coletadas deve ser cuidadosamente estudada e fortemente fundamentada, pois interfere diretamente na qualidade e relevância dos resultados obtidos com a mensuração. Sem uma política bem definida para a definição de um conjunto de métricas, a probabilidade de se escolher arbitrariamente as mais adequadas é remota.

Segundo Baker et al. (1990), Fenton e Pfleeger (1997) e Florac e Carleton (1999), as métricas servem para auxiliar três atividades básicas. A primeira é tornar clara a compreensão do que está acontecendo no desenvolvimento e na manutenção do processo de software, ajudando a deixar mais visíveis aspectos do produto e do processo. A segunda é oferecer **controle** sobre o que está acontecendo nos projetos de software, ajudando a prever o que acontecerá e a realizar mudanças no processo para conseguir atingir as metas estabelecidas. A terceira é encorajar a melhoria contínua dos processos e dos produtos.

É fundamental que sejam definidas métricas adequadas de forma que retratem o desempenho do processo no decorrer do tempo, a fim de que o mesmo seja monitorado de forma a facilitar e guiar a tomada de decisões.

3.4.5.2 Desenvolver a Estratégia de Controle

Métricas são uma forma de medir sucesso. Deve-se desenvolver uma estratégia de controle para ajudar a garantir esse sucesso. O primeiro passo para isso é a criação de um plano de controle formal para garantir que as melhorias do projeto são sustentadas. Além disso, devem-se realizar revisões pós-implementação periodicamente (TAYTOR, 2003).

3.4.5.3 Celebrar e Comunicar o Sucesso

Como forma de dinamizar o trabalho em equipe é encorajada a celebração da conclusão de um projeto, assim como de marcos, por todos os integrantes da equipe, além de ser uma forma de comunicar a toda a organização o sucesso alcançado (TAYTOR, 2003).

É extremamente importante dentro de um programa *Six Sigma* cultivar a integração entre os participantes. Os gerentes devem saber o que os empregados estão fazendo para contribuir ao sucesso da empresa, e os empregados devem saber o que eles estão fazendo para contribuir ao sucesso da empresa. A equipe deve demonstrar como o trabalho desenvolvido contribuiu para o sucesso da organização. Isto cria uma unidade altamente eficaz em todos os níveis dentro de uma atmosfera com moral elevado (LARSON, 2003).

3.4.5.4 Implementar o Plano de Controle

Devem-se institucionalizar melhorias através de modificações em sistemas, estruturas e processos, tudo isto acompanhado por um plano de controle, em que ficam registrados os responsáveis, o que está sendo mensurado, parâmetros de desempenho e medidas corretivas aplicadas (STAMATIS, 2004).

3.4.5.5 Medir e Comunicar as Melhorias

O último passo necessário para garantir que os ganhos foram mantidos é medir periodicamente as melhorias obtidas e comunicar os resultados obtidos. Isso ocorre mesmo com a equipe já dispersa, havendo reuniões periódicas para verificar possíveis processos a serem melhorados, já que o DMAIC é um processo iterativo (TAYNTOR, 2003).

3.5 DMAIC X CMMI

A partir do ponto de vista da definição de um processo, há uma sinergia natural entre as áreas de alta maturidade e as fases do método DMAIC do *Six Sigma*. As táticas do *Six Sigma* podem ser utilizadas para apoiar algumas áreas de processos de níveis de maturidade mais altos. Por exemplo, a Gerência Quantitativa de Projeto e Análise e Resolução de Causas podem se utilizar dos passos, subpassos e ferramentas do DMAIC (SIVIY et al., 2005).

Siviy et al. (2007) listam diversas organizações de software que obtiveram sucesso na implantação de modelos múltiplos como ISO 9001, ITIL CMMI e *Six Sigma* para melhoria de seus processos. Ainda destacam que organizações que tem implementado CMMI e *Six Sigma* em conjunto podem ter os seguintes benefícios:

- Alinhamento das atividades de melhoria de domínio específico com necessidades de negócio.
- Eficiência nos esforços de melhoria de sinergia de iniciativa – com *Six Sigma* aumentando o foco no negócio e o CMMI provendo conhecimento no domínio, mitigando o risco de implementadores em cada iniciativa reinventarem o que já existe.
- Execução acelerada de maturidade organizacional.
- Reconhecimento do uso de medidas de desempenho relevantes, construindo engenharia e pontos de vista do cliente.
- Mudança de cultura, para se tornar uma organização orientada a dados.
- Redução de custos.

Na Figura 3.2 Siviy et al. (2007) mostram a relação das áreas de processo do CMMI com os passos da metodologia DMAIC. Uma das áreas de processo destacadas no mapeamento é a de análise e resolução de causas, que tem relacionamento com as fases Analisar e Melhorar do DMAIC.

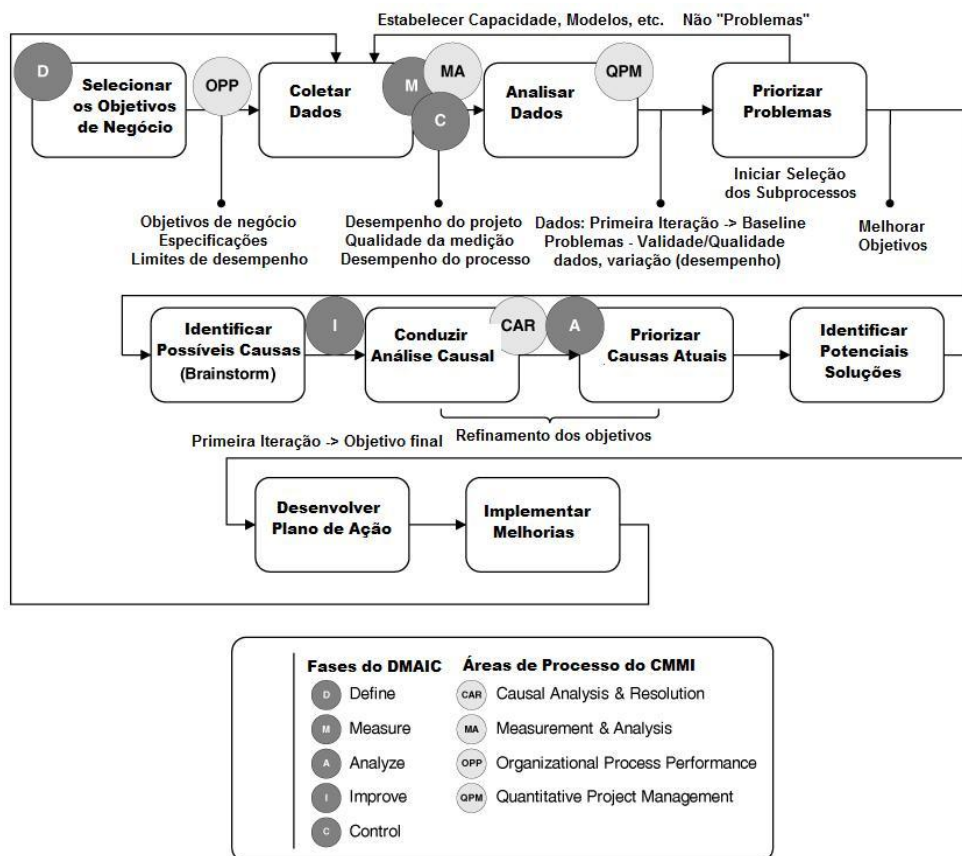


Figura 3.2: Áreas de Processo do CMMI e Áreas do DMAIC (SIVIY et al., 2007).

Siviy et al. (2007) fazem um mapeamento das fases do DMAIC com algumas áreas de processo do CMMI por práticas genéricas e específicas, que são aderentes ao método DMAIC. Esse mapeamento é ilustrado na Tabela 3.3.

Tabela 3.3: Mapeamento das Fases e Passos do DMAIC com as Práticas das Áreas de Processo do CMMI (SIVIY et al., 2007).

Fase	Passos	Práticas
Definir	- Definir o Escopo do Projeto. - Alinhar as melhorias do processo com os objetivos de negócio.	- Foco no Processo Organizacional (SG 1) - Desempenho do Processo Organizacional (SG1) - Inovação Organizacional (SG1) - GP 2.2, GP 3.1, GP 4.1, GP 5.1, GP 5
	- Estabelecer os projetos formais. - Estabelecer os projetos de melhoria.	- Foco no Processo Organizacional (SG 1) - Inovação Organizacional (SG1) - GP 4.1, GP 5.1
Medir e Analisar	Identificar (definir) os dados necessários; estabelecer repositórios.	- Medição e Análise (SG 1) - Definição do Processo Organizacional (SG 1) - Foco no Processo Organizacional (SG 1) - Análise e Resolução de Causas (SG 2) - Gerência Quantitativa de Projeto (SG 2)

		- GP 2.8, GP 3.2, GP 4.2, GP 5.1, GP 5.2
	- Consolidar a <i>Baseline</i> de Dados.	- Definição do Processo Organizacional (SG 1) - Desempenho do Processo Organizacional (SG1) - GP 2.8, GP 3.2, GP 5.2
	- Explorar os dados. - caracterizar o processo e o problema.	- Medição e Análise (SG 2) - Desempenho do Processo Organizacional (SG 1) - Análise e Resolução de Causas (SG 1) - Inovação Organizacional (SG2) - GP 2.8, GP 3.2, GP 5.2
Melhorar e Controlar	- Identificar melhorias das alternativas	- Análise de Decisão (SG 1) - Inovação Organizacional (SG 1) - Desempenho do Processo Organizacional (SG 1) - GP 5.1
	- Definir o método de controle e implementar.	- Medição e Análise (SG 2) - Desempenho do Processo Organizacional (SG 1) - Inovação Organizacional (SG 2) - Análise e Resolução de Causas (SG 2) - Gerência Quantitativa de Projeto (SG 2) - GP 2.8, GP 4.2, GP 5.1, GP 5.2

3.6 CONCLUSÃO DO CAPÍTULO

Este capítulo apresentou o *Six Sigma* e o método DMAIC detalhando cada uma de suas fases e também uma análise comparativa entre o método DMAIC e o CMMI. O capítulo seguinte apresenta a abordagem MiniDMAIC para implementação da área de processo de Análise e Resolução de Causas do CMMI no contexto de projetos de software e o mapeamento dos passos do MiniDMAIC com esta área de processo e o método DMAIC.

4 ABORDAGEM MINIDMAIC

Este capítulo apresenta uma abordagem para análise e resolução de causas de problemas de desenvolvimento de software no contexto de projetos.

4.1 INTRODUÇÃO

A abordagem proposta tem como base os passos definidos por Tayntor (2003) para o método DMAIC no âmbito do desenvolvimento de software. Alguns passos foram suprimidos do método e outros foram adaptados para tornar mais fácil a execução do processo nos projetos de desenvolvimento de software. Foi tomado o cuidado para que a essência do modelo não fosse alterada, conforme os objetivos que Tayntor (2003) define na Tabela 4.1.

Tabela 4.1: Objetivos Definidos por Tayntor (2003) para o Método DMAIC.

O Método DMAIC	
Fases	Objetivos
Definir	Identificar o problema e os clientes; definir e priorizar os requisitos do cliente; definir o processo atual.
Medir	Confirmar e quantificar o problema; medir as diversas etapas do processo atual; rever e esclarecer o escopo do problema, se necessário; definir os resultados desejados.
Analisar	Determinar a causa raiz do problema; propor soluções.

Melhorar	Priorizar soluções; desenvolver e implementar as soluções que trarão maiores benefícios.
Controlar	Medir as melhorias; comunicar e comemorar o sucesso; assegurar que as melhorias no processo serão implantadas.

A abordagem MiniDMAIC foi definida para organizações que possuam o processo aderente às práticas dos níveis 4 e 5 do modelo de maturidade CMMI.

Diante deste contexto, este trabalho visa a elaboração de uma abordagem baseada no DMAIC do *Six Sigma*, denominada MiniDMAIC, para tratar a área de processo de Análise e Resolução de Causas do CMMI em projetos de desenvolvimento de software, procurando reduzir as desvantagens do uso do DMAIC relatadas no capítulo de introdução deste trabalho. Também visa mostrar a aplicação da abordagem em projetos de desenvolvimento de software em uma organização a partir de uma ferramenta, onde foi implementada as práticas do MiniDMAIC.

4.2 ABORDAGEM DEFINIDA – MINIDMAIC

O MiniDMAIC é uma estratégia que visa à simplificação do método DMAIC com o intuito de tratar as causas e resolução de problemas em projetos de desenvolvimento de software de forma mais prática, rápida, com menos riscos e custos, prevenindo recorrências futuras, implantando melhorias no processo de desenvolvimento e assim, aumentando cada vez mais a satisfação dos seus clientes (GONÇALVES et al., 2008a, BEZERRA et al., 2009).

Esta abordagem foi definida inicialmente conforme Gonçalves (2008a) e foi aplicada em projetos pilotos de uma organização que, no período estava implantando os níveis 4 e 5 do modelo CMMI. Durante a execução da abordagem nos projetos pilotos foram identificadas algumas melhorias na abordagem e esta foi refinada.

Com base nas melhorias implantadas o MiniDMAIC foi executado em outros projetos de desenvolvimento de software tendo sido publicado um segundo trabalho com estudo de caso de alguns projetos que executaram a abordagem refinada (BEZERRA et al., 2009). Após este último trabalho, melhorias foram adicionadas à abordagem e foram validadas em

avaliação oficial CMMI nível 5 na organização em que foi aplicado o MiniDMAIC. Pode-se verificar que a abordagem apresentada neste trabalho passou por várias validações sendo refinada e executada em vários projetos de desenvolvimento de software, demonstrando eficácia na análise e resolução de causas no contexto destes projetos.

A grande diferença entre o MiniDMAIC e o DMAIC é que o DMAIC, a partir de uma análise e resolução das causas para o problema definido, tem como objetivo maior a melhoria de um dos processos padrão da organização, implantando as melhorias de forma controlada na organização. O MiniDMAIC apenas trata as causas em nível do projeto e tem como objetivo a prevenção e tratamento dos problemas definidos através da análise e resolução da(s) causa(s) raiz(es) destes problemas, podendo apenas auxiliar na melhoria organizacional do processo (BEZERRA et al., 2009).

Além disso, o DMAIC exige a comprovação estatística das causas dos problemas e das melhorias alcançadas, o que não é necessário no MiniDMAIC, que levanta e prioriza as causas por meio de ferramentas mais simples, como Diagrama de *Ishikawa* e Gráfico de Pareto, e analisa as melhorias obtidas observando o progresso dos indicadores do projeto (BEZERRA et al., 2009).

As principais características do MiniDMAIC são:

- Curta duração.
- Necessidade de conhecimento básico de estatística.
- Associada a riscos.
- Custo baixo em relação ao método DMAIC.
- Adequado para projetos de desenvolvimento de software.

Os problemas que merecem um tratamento mais criterioso através da aplicação da abordagem MiniDMAIC podem ser definidos em nível organizacional (Ex.: limites de controle, quantidade de defeitos, etc.). No entanto, deve estar bem claro para a equipe do projeto a diferença entre os problemas que requerem apenas ações simples e imediatas, e os problemas que requerem o tratamento definido na abordagem MiniDMAIC. Ações simples

são adequadas para tratamento de itens simples de melhoria que podem ser executados tipicamente por uma pessoa com pouco esforço e a causa/solução é conhecida ou provável.

A execução do MiniDMAIC em um projeto de desenvolvimento de software também deve levar em conta o porte do projeto e a frequência com que ocorre a coleta dos indicadores em uma organização. Para organizações que possuam uma coleta mensal dos indicadores a execução deve levar em conta que o projeto deve ter no mínimo um mês de duração. Caso o projeto tenha iterações curtas, o tratamento do problema pela abordagem MiniDMAIC servirá para que o problema não ocorra em iterações posteriores. No caso de projetos com um mês de duração a ação pode terminar de ser executada no final do projeto. Apesar da ação não tratar o problema em tempo hábil para que melhorias surtam efeito no projeto, a execução desta ação pode ter benefícios que auxiliarão em outros projetos da organização.

Exemplos de problemas de projetos que merecem o tratamento através da abordagem MiniDMAIC são:

- Projeto fora de controle, onde os resultados dos indicadores com processos controlados estatisticamente não atendam os limites de especificação¹¹ definidos pelo projeto ou limites de *baseline*¹² organizacionais (Ex.: produtividade, desvio na entrega, densidade de defeitos, etc.).
- Problemas recorrentes no projeto.
- Número elevado de defeitos em testes sistêmicos.
- Elevado número de defeitos encontrados pelo cliente.

11 Limites de especificação são limites estabelecidos pelo projeto para o indicador com processos controlados estatisticamente com base nos objetivos do cliente ou do projeto. Estes podem ser estabelecidos com base nos limites estabelecidos pela organização.

12 O desempenho esperado para o processo pode ser utilizado para estabelecer os objetivos de qualidade e de desempenho e também, inicialmente, como uma *baseline* com a qual o desempenho dos projetos atuais pode ser comparado. Esta informação será utilizada para gerenciar quantitativamente os projetos. Cada projeto gerenciado quantitativamente fornecerá resultados atuais sobre o desempenho dos processos que se tornarão parte dos dados da *baseline* dos processos organizacionais (SOFTEX, 2009b).

Quando a análise de causa e defeito for realizada, a seleção de defeitos para análise deve levar em consideração os seguintes fatores:

- Tipos de defeitos mais comuns.
- Frequência da ocorrência.
- Similaridade entre defeitos.

Nesta abordagem, defeitos são tratados como sendo falhas, ao considerarmos as definições de defeito, erro e falha apresentados na IEEE 610.12-1990. Este conceito foi tomado, pois a abordagem MiniDMAIC tem como base para fase medir a classificação ortogonal de defeitos, que utiliza a mesma definição.

Como apoio à abordagem, podem ser utilizadas ferramentas como planilhas eletrônicas, ferramentas de gestão de projetos, entre outras.

Nos itens abaixo são descritas as fases do MiniDMAIC, que utiliza as mesmas fases do método DMAIC, e uma última fase incluída para disponibilização das oportunidades de melhoria, identificadas na execução da abordagem, para os ativos organizacionais. A Figura 4.1 mostra a sequência dos passos que a abordagem pode seguir.

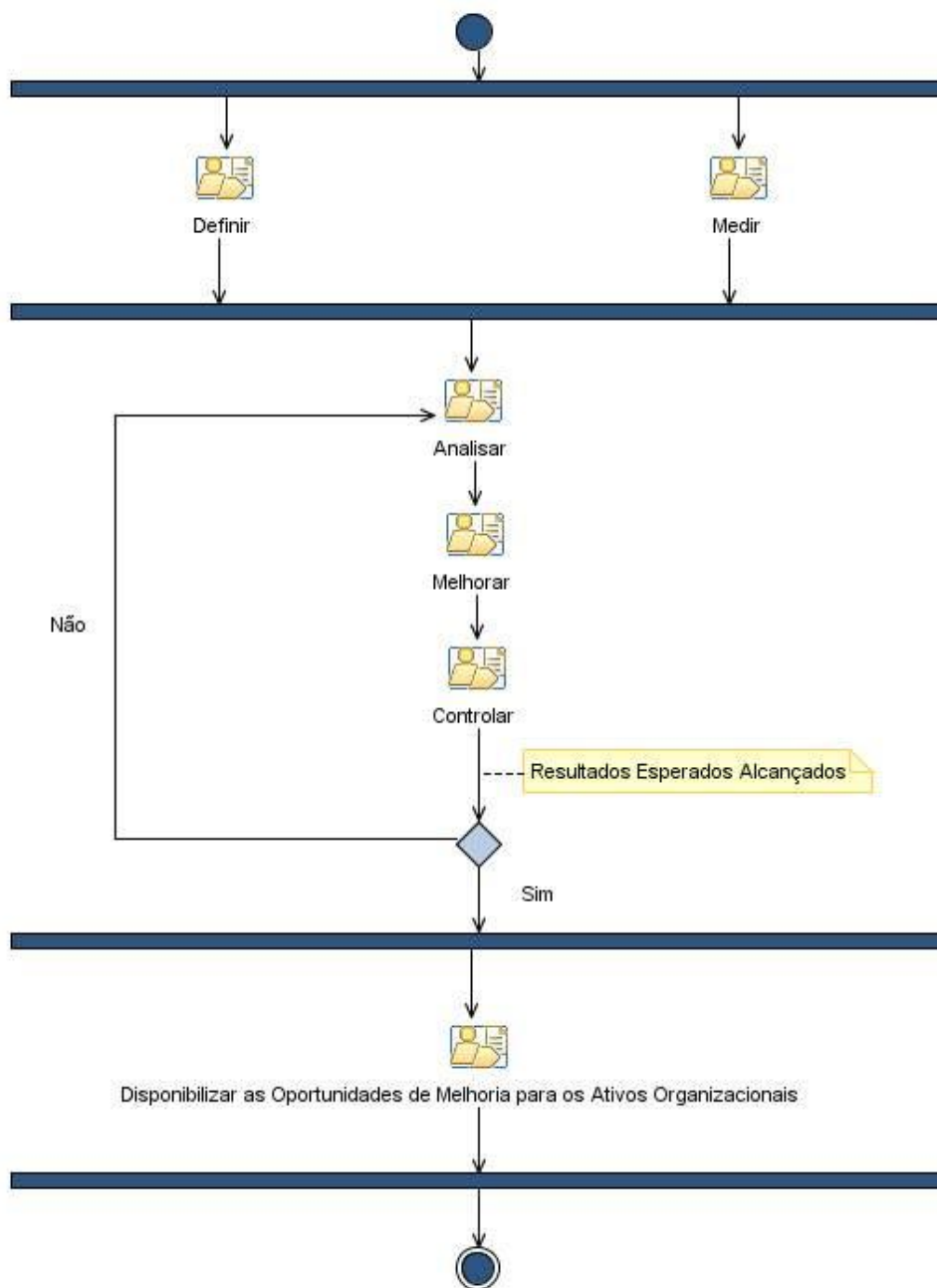


Figura 4.1: Fases do MiniDMAIC.

4.2.1 Fase: Definir

A fase “Definir” é uma fase de planejamento da ação e consiste na definição do problema, fontes, processos e subprocessos impactados e resultados esperados, além da formação da equipe, conforme a representação da Tabela 4.2.

Tabela 4.2: Representação da Atividade “Definir” do MiniDMAIC.

Objetivos: Consiste na definição do problema, fontes, processos e subprocessos impactados e resultados esperados, além da formação da equipe.	
Papéis Principais: Gerente do Projeto	Papéis Adicionais: Equipe do Projeto
Entrada: Fontes do Problema Base Histórica (MiniDMAICs de projetos anteriores)	Saída: Definição do Problema
Passos: <ol style="list-style-type: none"> 1. Definir o problema 2. Determinar a fonte do problema 3. Identificar os processos afetados 4. Identificar os riscos de não tratamento do problema 5. Definir os resultados esperados 6. Formar equipe e estimar prazo de execução 	

4.2.1.1 Passo: Definir o Problema

Deve ser definido o problema que será tratado para que esteja clara a sua importância e sejam definidos os seus objetivos. Uma busca deve ser feita na base histórica organizacional à procura de problemas similares que foram tratados em outros projetos com uma ação MiniDMAIC para ajudar na definição e resolução das causas raízes do problema. É importante descrever qual o impacto ou consequências do problema sobre o projeto. Esta descrição deve estar focando somente nos sintomas e não em causas ou soluções.

4.2.1.2 Passo: Determinar a Fonte do Problema

Esta etapa deve mostrar qual foi a fonte que mostrou a ocorrência do problema. Exemplos de fontes de problemas em projetos de desenvolvimento de software são:

- Indicadores do projeto.
- Relatório de testes sistêmicos.
- Resultado dos testes de integração.
- Relatório de testes do cliente.
- Problemas em revisão técnica que afetam os requisitos ou o funcionamento correto da aplicação.
- Reclamações do cliente.

4.2.1.3 Passo: Identificar os Processos Afetados

Identificar quais os processos e subprocessos que foram afetados pelo problema definido. Caso o problema seja resultado de um indicador fora de controle, identificar a *baseline* de processo associada. As *baselines* de processo selecionadas pelo projeto levam em conta os objetivos de desempenho do cliente.

4.2.1.4 Passo: Identificar os Riscos de não Tratamento do Problema

Podem ser identificados pelo gerente de projeto os riscos de não tratamento do problema para que estes sejam tratados e acompanhados de acordo com o processo definido que atende à área de processo de Gerência de Riscos (*Risk Management – RSKM*) do CMMI.

4.2.1.5 Passo: Definir os Resultados Esperados

Neste passo, são definidos os resultados esperados que devem ser atingidos com a execução da abordagem MiniDMAIC para o tratamento do problema em questão. Os resultados esperados devem ser definidos de forma quantitativa, podendo-se utilizar alguns indicadores associados ao problema definido.

4.2.1.6 Passo: Formar a Equipe e Estimar Prazo de Execução

Neste passo será definida a equipe que irá participar de cada fase do MiniDMAIC e a estimativa de prazo para execução de cada uma delas. Em uma ação MiniDMAIC não há necessidade de se ter *Black Belts* na liderança. Por serem simples e diretamente relacionadas ao projeto, é necessário apenas um conhecimento básico sobre *Six Sigma* e treinamento na abordagem MiniDMAIC. O mais relevante é o entendimento relacionado ao negócio do projeto e técnicas de gestão, sendo, portanto, indicado que o Gerente do Projeto lidere o MiniDMAIC. O tamanho da equipe MiniDMAIC pode variar de acordo com a necessidade do problema. Em algumas situações, podemos ter apenas o gerente do projeto e um membro da equipe contando com a participação de outras pessoas apenas em determinados passos, como por exemplo, contar com o apoio de um líder *Green Belt* (principalmente na fase medir e analisar).

4.2.2 Fase: Medir

A fase “Medir” consiste na coleta e análise das medições (existentes ou a definir) relacionadas ao problema para conhecimento da situação atual do projeto e dos processos relacionados, conforme a Tabela 4.3. Esta fase pode ocorrer em paralelo à fase “Definir” apoiando a definição do problema. Caso a medição já seja analisada no nível de projeto a análise deve ser verificada no relatório que possui a coleta e análise das medições. Caso a medição definida esteja dentro da ação MiniDMAIC esta deve ser coletada e analisada na fase “Medir”.

Tabela 4.3: Representação da Atividade “Medir” do MiniDMAIC.

Objetivos: Consiste na coleta e análise das medições (existentes ou a definir) relacionadas ao problema para conhecimento da situação atual do projeto e dos processos relacionados.	
Papéis Principais: Gerente do Projeto Equipe do Projeto	Papéis Adicionais: Não se aplica
Entrada: Definição do Problema Fonte do Problema	Saída: Plano de Medição Classificação dos defeitos/problema
Passos: 1. Planejar medições 2. Medir situação atual	

4.2.2.1 Passo: Planejar Medições

Neste passo deve ser analisado se existe a necessidade de uma nova medição que forneça maiores evidências para o problema em questão. Em grande parte das situações, as medições já estarão sendo conduzidas de acordo com o processo definido que atende à área de processo de Medição e Análise (*Measurement and Analysis – MA*). Pode também ser planejada uma nova medição que forneça maiores evidências para consolidar e ampliar o entendimento do problema e suas consequências.

4.2.2.2 Passo: Medir Situação Atual

As medições selecionadas no passo anterior devem ser executadas de acordo com o planejado. É necessário coletar as informações e medir a situação atual do projeto. Posteriormente, essas mesmas medidas serão utilizadas para aferir a melhoria obtida. No caso da coleta dos defeitos, recomenda-se o uso do *Template – Análise de Causas* (apresentado no

Apêndice B) de forma a priorizar os defeitos que merecem uma análise mais detalhada das causas.

4.2.3 Fase: Analisar

A fase “Analisar” consiste na identificação e priorização das causas raízes do problema a partir de técnicas para garantir que as causas raízes a serem atacadas realmente estão relacionadas ao problema, bem como, a definição de possíveis ações para solucionar o problema, conforme podemos observar na Tabela 4.4.

Tabela 4.4: Representação da Atividade “Analisar” do MiniDMAIC.

Objetivos: Consiste na identificação e priorização das causas raízes do problema a partir de técnicas para garantir que as causas raízes a serem atacadas realmente estão relacionadas ao problema, bem como, a definição de possíveis ações para solucionar o problema.	
Papéis Principais: Gerente do Projeto Equipe do Projeto	Papéis Adicionais: Não se aplica
Entrada: Definição do problema Classificação dos defeitos/problema	Saída: Possíveis causas priorizadas Ações candidatas
Passos: 1. Determinar as causas do problema 2. Priorizar as causas do problema 3. Definir ações candidatas	

4.2.3.1 Passo: Determinar as Causas do Problema

Este é um dos passos mais importantes do MiniDMAIC, uma vez que seu propósito é encontrar a causa raiz do problema. Caso esta atividade não seja feita corretamente, o resultado do MiniDMAIC pode ser comprometido, pois todas as atividades seguintes terão como base o resultado deste passo. Sendo assim, é importante que participem deste passo as pessoas que possuem conhecimento a respeito do problema e que possam contribuir com informações sobre suas causas. Exemplos de técnicas para determinar as causas do problema são: *brainstorming*, cinco porquês, diagrama de causa e efeito (ISHIKAWA, 1998), entre outras. Para execução deste passo pode ser utilizado o *Template – Análise de causas* (apresentado no Apêndice B). Caso sejam analisados defeitos, deve ser utilizada como insumo para esta fase, a classificação dos defeitos para verificar onde os defeitos estão mais concentrados.

4.2.3.2 Passo: Priorizar Causas do Problema

A priorização das causas do problema identificadas deve ser feita de acordo com o processo definido para a área de Processo Análise e Tomada de Decisão (*Decision Analysis and Resolution – DAR*). Outra forma de priorizar as causas é a partir da utilização do gráfico de Pareto (JURAN, 1991), onde 20% das causas podem contribuir com 80% dos defeitos. Caso seja adotado o gráfico de Pareto, as causas podem ser agrupadas de acordo com a criticidade de defeitos, a origem em que os defeitos foram inseridos e pelo tipo de defeito. Para execução deste passo pode ser utilizado o *Template – Análise de causas* (apresentado no Apêndice B).

4.2.3.3 Passo: Definir Ações Candidatas

Neste passo devem ser levantadas com a equipe do projeto através da técnica de *brainstorming* as possíveis ações que venham a tratar as causas do problema em questão. Cada ação de tratamento do problema deve ser associada à respectiva causa relacionada.

4.2.4 Fase: Melhorar

A fase “Melhorar” consiste na definição e análise de viabilidade das ações propostas, elaboração e execução do plano de ação, bem como no acompanhamento dos resultados obtidos, conforme podemos ver na representação da Tabela 4.5.

Tabela 4.5: Representação da Atividade “Melhorar” do MiniDMAIC.

Objetivos: Consiste na definição e análise de viabilidade das ações propostas, elaboração e execução do plano de ação, bem como no acompanhamento dos resultados obtidos.	
Papéis Principais: Gerente do Projeto Equipe do Projeto	Papéis Adicionais: Não se aplica
Entrada: Ações Candidatas	Saída: Ações a serem implementadas priorizadas Plano de ação definido e executado
Passos: 1. Priorizar ações 2. Elaborar e executar plano de ação 3. Acompanhar ações	

4.2.4.1 Passo: Priorizar Ações

As ações candidatas podem ser priorizadas de acordo com o processo definido para a área de processo de Análise e Tomada de Decisão (*Decision Analysis and Resolution – DAR*). Ou pode ser feita uma análise de viabilidade para implementação de cada ação. Uma causa priorizada pode ter uma ou mais ações, assim como uma ação pode estar atacando uma ou mais causas priorizadas na fase “Analisar”. As ações devem referenciar quais as causas atacadas. A análise de viabilidade deve verificar aspectos como complexidade, tempo e custo para execução da ação dentro do projeto. Conforme o passo Definir Ações Candidatas as ações priorizadas devem estar associadas às respectivas causas relacionadas.

4.2.4.2 Passo: Elaborar e Executar Plano de Ação

Um plano de ação para execução das ações priorizadas e aprovadas deve ser elaborado pelo gerente do projeto para tratamento e acompanhamento das ações. Este plano deve conter as seguintes informações:

- Tarefas a serem executadas.
- Responsável pela execução da tarefa.
- Esforço necessário para execução da tarefa.
- Prazo para conclusão da tarefa.

A execução do plano de ação se dá através da implementação das suas tarefas. As tarefas podem ser distribuídas para a equipe do projeto.

4.2.4.3 Passo: Acompanhar Ações

Neste passo, as tarefas devem ser monitoradas para que se tenha uma visibilidade do avanço do MiniDMAIC. Estes resultados devem ser acompanhados pelo gerente do projeto de acordo com o processo de Monitoramento e Controle do Projeto (*Project Monitoring and Control – PMC*).

4.2.5 Fase: Controlar

A fase “Controlar” consiste na medição, avaliação dos resultados obtidos e divulgação dos resultados e lições aprendidas, conforme podemos observar na representação da Tabela 4.6.

Tabela 4.6: Representação da Atividade “Controlar” do MiniDMAIC.

Objetivos: Consiste na medição, avaliação dos resultados obtidos e divulgação dos resultados e lições aprendidas.	
Papéis Principais: Gerente do Projeto	Papéis Adicionais: Equipe do Projeto Alta direção
Entrada: Plano de medição	Saída: Resultados alcançados Lições aprendidas MiniDMAIC consolidado (informações a serem enviadas para a base histórica)
Passos: 1. Medir resultados 2. Avaliar resultados 3. Divulgar principais resultados e lições aprendidas	

4.2.5.1 Passo: Medir Resultados

Após a implementação das ações no projeto os resultados obtidos do período devem ser medidos pelo gerente de projeto utilizando os mesmos indicadores selecionados na fase “Medir”, de forma a verificar se o resultado quantitativo foi alcançado.

4.2.5.2 Passo: Avaliar Resultados

Na avaliação dos resultados obtidos deve ser feita uma análise pelo gerente do projeto juntamente com a equipe para verificar se os resultados esperados estabelecidos na fase “Definir” foram atingidos e se ocorreu melhoria em relação ao coletado na fase “Medir” antes do tratamento do problema. Esta comparação vai servir de base para comprovar se houve melhoria no projeto e verificar se o problema foi realmente tratado.

4.2.5.3 Passo: Divulgar os Principais Resultados e Lições Aprendidas

Ao final da execução do MiniDMAIC, os resultados obtidos devem ser compartilhados pelo projeto com toda a organização, registrando-os em um repositório organizacional acessível a todos os projetos. O compartilhamento destas informações pode

servir de insumo para o tratamento de problemas similares em outros projetos, bem como para melhorias dos processos no nível organizacional. A forma de divulgação deve seguir o processo definido para a área de processo de Foco no Processo Organizacional (*Organizational Processes Focus – OPF*), que define como serão compartilhadas as lições aprendidas pela organização. Caso sejam identificadas possíveis melhorias nos processos organizacionais, as mesmas devem ser encaminhadas para o Grupo de Engenharia de Processos (*Engineering Process Group – EPG*) para que possam ser analisadas e devidamente tratadas.

4.2.6 Disponibilizar as Oportunidades de Melhoria para os Ativos Organizacionais

A base histórica organizacional deve conter todos os dados advindo da execução dos MiniDMAICs dos projetos. Tendo em vista existirem dados de mais de um projeto, o grupo de engenharia do processo poderá analisar mais dados objetivando identificar tendências de problemas de forma a definir melhorias a serem implementadas nos processos e seus ativos em nível organizacional. Caso o problema já tenha causa conhecida, ou as causas já tenham sido identificadas dentro dos projetos, deve ser gerada uma ação simples organizacional.

Além dos MiniDMAIC, de acordo com Albuquerque (2008), as seguintes fontes de dados também poderão auxiliar na identificação de problemas recorrentes nos ativos de processos da organização: (i) avaliação da adequação do processo ; (ii) avaliação da aderência ao processo; (iii) avaliação da aderência dos produtos gerados pelas atividades do processo aos padrões estabelecidos na organização; (iv) avaliação *post-mortem*; (v) indicadores de monitoração dos processos; (vi) lições aprendidas; (vii) solicitação de dispensa de execução de atividade; (viii) diretrizes; (ix) justificativas de alteração do processo e (x) solicitações de alteração para o processo.

Vale ressaltar que algumas dessas fontes podem ser úteis, também, na contextualização dos defeitos e problemas identificados.

Algumas informações devem ser registradas na base histórica da organização como: tipo de problema, causas do problema, ações tomadas para tratamento das causas e melhorias obtidas. Estas informações são importantes para que os problemas identificados nos projetos a partir da abordagem MiniDMAIC sejam relacionados e organizados de forma a possibilitar a identificação de problemas no nível organizacional.

4.3 DMAIC X MINIDMAIC

O MiniDMAIC é baseado nos passos do método DMAIC definidos por Tayntor (2003). Alguns passos foram suprimidos pela complexidade das técnicas estatísticas relacionadas, como por exemplo, no passo “Calcular o Nível Sigma Atual”, e também retirados os passos relacionados aos requisitos do cliente e mudanças no processo padrão, conforme ilustrado na Tabela 4.7. O MiniDMAIC tem como objetivo principal a análise e resolução de causas nos projetos de desenvolvimento de software e não foca em mudanças no processo padrão da organização, que é o objetivo principal do DMAIC.

Tabela 4.7: Comparação dos Passos do DMAIC Definidos por Tayntor (2003) com os Passos da Abordagem MiniDMAIC.

Fase	DMAIC (Passos)	MiniDMAIC (Passos)	Justificativa de Exclusão dos Passos
Definir	Definir o Problema	- Definir o Problema	
	Formar a Equipe	- Formar a Equipe e Estimar Prazo de Execução	
	Estabelecer um <i>Project Charter</i>	- Determinar a Fonte do Problema - Identificar os Riscos de não Tratamento do Problema - Identificar os Processos Afetados - Definir Resultados Esperados	
	Desenvolver o Plano do Projeto	-	- Não há necessidades de muitos planos para executar análise de causas em projetos
	Identificar os Clientes	-	- O cliente deve ser identificado no plano do projeto de software
	Identificar os Artefatos Resultantes	- Durante toda execução do MiniDMAIC	
	Identificar e Priorizar os Requisitos do Cliente	-	- Os requisitos do cliente não são identificados diretamente. Estes podem estar alinhados ao passo Identificar os Processos Afetados

Medir	Determinar o que Medir	- Planejar Medições	
	Conduzir as Medições	- Medir Situação Atual	
	Calcular o Nível Sigma Atual	-	- Necessário alto conhecimento em técnicas estatísticas, o que pode ser inviável no contexto de projetos
	Determinar a Capacidade do Processo	-	- Necessário alto conhecimento em técnicas estatísticas, o que pode ser inviável no contexto de projetos
	<i>Benchmark</i> dos líderes do Processo	-	- Este <i>Benchmark</i> deve ser realizado em nível organizacional, já que não há mudança no processo executado em nível de projeto
Analisar	Determinar as Causas de Variação	- Determinar as Causas do Problema - Priorizar as Causas do Problema	
	<i>Brainstorming</i> de Idéias para Melhoria de Processo	- Definir Ações Candidatas	
	Determinar as Melhorias que tem Maior Impacto nos Requisitos do Cliente	- Definir Ações Candidatas	
	Desenvolver o Mapa do Processo Proposto	-	- Não há necessidade de disponibilização de mudança no processo em nível de projeto
	Avaliar os Riscos Associados com o Processo Revisado	-	- Não há necessidade de disponibilização de mudança no processo em nível de projeto, pois estes riscos podem ser tratados no nível organizacional
Melhorar	Obter a Aprovação das Mudanças Propostas	- Priorizar Ações	
	Finalizar o Plano de Implementação	- Elaborar e Executar Plano de Ação	
	Implementar as Mudanças Aprovadas	- Elaborar e Executar Plano de Ação - Acompanhar Ações	

Controlar	Estabelecer as Métricas Chaves	- Medir Resultados - Avaliar Resultados	
	Desenvolver a Estratégia de Controle	-	- Não há necessidade de disponibilização de mudança no processo em nível de projeto
	Celebrar e Comunicar o Sucesso	- Divulgar os Principais Resultados e Lições Aprendidas	
	Implementar o Plano de Controle	-	- Não há necessidade de disponibilização de mudança no processo em nível de projeto
	Medir e Comunicar as Melhorias	- Medir Resultados - Avaliar Resultados - Divulgar os Principais Resultados e Lições Aprendidas	

4.4 CAR X MINIDMAIC

Para um melhor entendimento do relacionamento do MiniDMAIC com a área de processo Análise e Resolução de Causas (CAR), foi realizado um mapeamento entre os passos do MiniDMAIC com as práticas específicas de CAR, como podemos ver na Tabela 4.8. Vale ressaltar que tal relacionamento não torna necessário que o MiniDMAIC cubra todas as práticas exigidas no CMMI, uma vez que algumas das práticas da PA de Análise e Resolução de Causas devem ser executadas em nível organizacional.

Tabela 4.8: Relacionamento dos Passos do MiniDMAIC com as Práticas Específicas de CAR

Fase	MiniDMAIC (Passos)	CAR (Práticas Específicas)	Observações
Definir	Passo 1 - Definir o Problema	-	Relacionado à PA <i>Quantitative Project Management</i> – QPM
	Passo 2 - Determinar a Fonte do Problema	-	Relacionado à PA <i>Quantitative Project Management</i> – QPM
	Passo 3 - Identificar os Processos Impactados	-	Relacionado à PA <i>Quantitative Project Management</i> – QPM
	Passo 4 - Identificar os Riscos de não Tratamento do	-	Relacionado à PA <i>Risk Management</i> – RSKM

	Problema		
	Passo 5 - Definir Resultados Esperados	-	Relacionado à PA <i>Quantitative Project Management – QPM</i>
	Passo 6 - Formar a Equipe e Estimar Prazo de Execução	-	Relacionado à PA <i>Project Monitoring and Control – PMC</i> e GP 2.7 - Identificar e Envolver os <i>Stakeholders</i> Relevantes
Medir	Passo 1 – Planejar Medições	SP 1.1 - Selecionar os dados de defeitos para análise	-
	Passo 2 – Medir Situação Atual	SP 1.1 - Selecionar os dados de defeitos para análise	-
Analisar	Passo 1 - Determinar as Causas do Problema	SP 1.2 - Analisar as Causas	-
	Passo 2 – Priorizar as Causas do Problema	SP 1.2 - Analisar as Causas	-
Melhorar	Passo 1 - Definir Ações Candidatas	SP 1.2 - Analisar as Causas	-
	Passo 2 – Priorizar Ações	SP 2.1 - Implementar as Ações Propostas -	Relacionado à GP 2.10 - Revisar o Status com o Gerente de Alto Nível
	Passo 3 - Elaborar e Executar Plano de Ação	SP 2.1 - Implementar as Ações Propostas	-
	Passo 4 - Acompanhar Ações	-	Relacionado à PA <i>Project Monitoring and Control – PMC</i>
Controlar	Passo 1 - Medir Resultados	SP 2.2 - Avaliar os Efeitos das Mudanças	-
	Passo 2 - Avaliar Resultados	SP 2.2 - Avaliar os Efeitos das Mudanças	-
	Passo 3 - Divulgar os Principais Resultados e Lições Aprendidas	SP 2.3 - Registrar os Dados	-

Como pode-se observar, para o atendimento da área de processo de análise e resolução de causas, não foram necessários vários passos definidos no DMAIC. Essa análise das fases do método DMAIC serviu de base para a definição da abordagem proposta.

4.5 CONCLUSÃO DO CAPÍTULO

Este capítulo apresentou a proposta desta dissertação, que consiste numa abordagem para análise e resolução de causas de problemas em projetos de desenvolvimento de software baseada nas fases do método DMAIC. A abordagem é detalhada através do fluxo das fases e passos definidos para cada fase. Além disso, o capítulo apresentou como as lições aprendidas das ações MiniDMAICs nos projetos podem auxiliar na implantação de melhorias organizacionais e também como as práticas da área de Processo CAR estão relacionados aos passos da abordagem proposta. O capítulo seguinte apresenta uma experiência de uso desta abordagem.

5 EXPERIÊNCIA DE USO

Esse capítulo apresenta a experiência de uso da abordagem MiniDMAIC em projetos de uma organização de maturidade nível 5 do CMMI e a customização da abordagem na ferramenta Jira.

5.1 INTRODUÇÃO

A abordagem MiniDMAIC foi aplicada no contexto de projetos do Instituto Atlântico¹³ para utilização de análise de causas e defeitos dentro dos projetos de desenvolvimento de software. O Instituto Atlântico é uma instituição de pesquisa e desenvolvimento com sede em Fortaleza, Ceará, fundada em novembro de 2001 por iniciativa do Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPqD), e filiais em Sobral e São Paulo. Desde a sua fundação, o Instituto Atlântico iniciou um amplo Programa de Qualidade, sendo aderente à norma ISO 9001:2000 e obtendo recentemente o nível 5 de maturidade no modelo CMMI-DEV 1.2. A empresa também possui um forte incentivo para o gerenciamento de projetos disciplinado, tendo sido estabelecido em 2008 um escritório de projetos para gerir o portfólio de projetos da empresa e manter o processo de gestão de projetos baseado no PMBOK e CMMI, mas adaptado à cultura da empresa. Visando a melhoria contínua dos seus processos, o Atlântico implantou os níveis 4 e 5 de maturidade do modelo CMMI com o apoio da metodologia *Six Sigma* (TAYNTOR, 2003).

O Instituto Atlântico tem no seu quadro de funcionários cerca de 270 colaboradores que participam do desenvolvimento de projetos de pesquisa e desenvolvimento para vários tipos de negócio (automação comercial, automação bancária, automação de processos, portais,

¹³ Site do Instituto Atlântico: www.atlantico.com.br.

telecomunicações, setor financeiro, indústria, consultoria, gestão e governo), abrangendo as mais diversas modalidades, entre elas: fábrica de sistema, fábrica de software, fábrica de teste ou fábrica de solução.

Em agosto de 2009 o Instituto Atlântico foi avaliado no nível 5 de maturidade do CMMI e obteve sucesso, alcançando o nível máximo de maturidade do modelo. A abordagem utilizada para cobrir a área de processo Análise e Resolução de Causas no contexto dos projetos de desenvolvimento de software da organização foi a abordagem do MiniDMAIC, não sendo encontrado nenhum ponto fraco em relação ao uso dessa abordagem nos projetos e tendo sido verificado o atendimento das práticas para todas as práticas específicas e genéricas em relação a essa área de processo. Para complementar o atendimento da área de processo Análise e Resolução de Causas no contexto organizacional, foram implementados projetos de melhoria com auxílio do método DMAIC.

5.2 IMPLEMENTAÇÃO DO MINIDMAIC NO JIRA

Todos os passos do MiniDMAIC foram implementados na ferramenta Jira¹⁴, uma ferramenta comercial de gerenciamento de *workflow* que pode ser facilmente customizada. A ferramenta já é utilizada na organização para acompanhamento de problemas e outras ações, e possibilitou implantar uma ação de análise de causas de forma mais simples nos projetos. No Apêndice A são apresentadas as telas de implementação no Jira de cada fase da ação MiniDMAIC e descrições de preenchimentos dos campos.

5.3 EXPERIÊNCIA DE USO DO MINIDMAIC

5.3.1 Caracterização do MiniDMAIC para organização

Seguindo as práticas da área de processo Análise e Resolução de Causa do CMMI, alguns critérios e condições foram definidos pela organização para disparar uma ação MiniDMAIC nos projetos. Os MiniDMAICs poderiam ser disparados para análise de causas de defeitos/problemas ou desvio nos indicadores do projeto. Nas reuniões gerenciais, o

14 Site do Jira: www.atlassian.com/software/jira/

coordenador do projeto deveria analisar juntamente com o gerente a necessidade de realizar MiniDMAIC para a situação apresentada. O responsável pelo planejamento e acompanhamento de uma ação MiniDMAIC deveria ser o coordenador do projeto.

Para a organização foram consideradas as seguintes fontes típicas de defeitos/problemas:

- Indicadores do projeto.
- Relatório de testes sistêmicos.
- Resultado dos testes de integração.
- Relatório de testes do cliente.
- Problemas em revisão técnica que afetam os requisitos ou o funcionamento correto da aplicação.
- Reclamações do cliente.

Além disso, as situações listadas abaixo poderiam requerer a análise de causa e defeito utilizando MiniDMAIC:

- Número elevado no indicador de testes sistêmicos. Por exemplo: indicador acima da meta do projeto ou fora dos limites de especificação.
- Projeto fora de controle, onde os resultados dos indicadores com processos controlados estatisticamente não atendam os limites de especificação definidos pelo projeto ou limites de *baseline* organizacionais (Ex.: produtividade, desvio na entrega, densidade de defeitos, etc.).
- Número elevado de defeitos classificados como *critical* e *blocker* em testes sistêmicos (de acordo com análise do coordenador).
- Elevado número de defeitos encontrados pelo cliente (de acordo com análise do coordenador).
- Defeitos encontrados na primeira bateria de teste do projeto.
- Necessidade de analisar os tipos de defeitos mais comuns.
- Erros ocorrendo de forma frequente nas diversas baterias de teste.

Quando a análise de causa e defeito fosse realizada, a seleção de defeitos para análise deveria levar em consideração os seguintes fatores:

- Tipos de defeitos mais comuns.
- Frequência da ocorrência.
- Similaridade entre defeitos.

A organização possui um processo de testes bem definido e para classificar os defeitos deveriam ser considerados: (i) a criticidade; (ii) os tipos de defeitos e (iii) a origem dos defeitos em relação às fases do ciclo de vida do desenvolvimento de software.

A criticidade dos defeitos baseou-se no padrão IEEE 1044 (1994) e possui a seguinte classificação:

- **Blocker:** falha que causa o bloqueio da funcionalidade principal testada ou da aplicação, impedindo que o teste seja executado. Também são considerados os casos que impossibilitam a execução de outros requisitos.
- **Critical:** falha em que os passos do caso de teste puderam ser executados, porém, tiveram um resultado desastroso. Também são considerados os casos onde uma funcionalidade secundária não pôde ser executada com sucesso.
- **Major:** falha que tem como retorno resultados incorretos, mas não trazem um grande impacto ao cliente.
- **Minor:** falha em pontos não essenciais de requisitos.
- **Trivial:** problemas considerados cosméticos / acessórios, que não afetam a funcionalidade do sistema.

Os tipos de defeitos da organização basearam-se na classificação ortogonal dos defeitos (CHILLAREGE et al., 1992) possuindo os seguintes tipos de defeitos:

- Interface.
- Função (funcionalidade).
- Montagem / empacotamento / integração.
- Atribuição.
- Documentação.

- Verificação (validação de campos).
- Algoritmo (lógica interna).
- Tempo / serialização / performance.

As origens dos defeitos também basearam-se na classificação ortogonal dos defeitos (CHILLAREGE et al., 1992), possuindo as seguintes origens:

- Requisitos.
- A&P – Arquitetura.
- A&P – Design.
- Implementação.
- Testes.

A abordagem MiniDMAIC foi aplicada em vários projetos na organização desde a sua definição. No entanto, ao longo da execução dos MiniDMAICs nos projetos de desenvolvimento de software do Atlântico a abordagem foi sendo refinada e melhor adequada para uma análise de causas mais eficaz e eficiente no contexto de projetos. A seguir, será apresentado um dos projetos da organização em que o MiniDMAIC foi executado. Vale ressaltar, que a aplicação do MiniDMAIC neste projeto, foi executada a abordagem já refinada com as melhorias sugeridas por outros projetos que utilizaram a abordagem anteriormente.

5.3.2 Caracterização do Projeto Piloto

O projeto de desenvolvimento de software da organização selecionado como piloto foi um projeto considerado de grande porte e que tem como característica iterações curtas, utilizando a metodologia Scrum (SCHWABER, 2004). Este projeto corresponde ao desenvolvimento de vários sub-projetos de aplicativos experimentais para dispositivos móveis (celulares). O projeto teve início em novembro de 2008 com término previsto para março de 2010. O projeto tem escopo totalmente flexível, com sub-projetos e seus requisitos sendo definidos em conjunto com o cliente ao longo do mesmo, restrito ao consumo de um banco de aproximadamente 24.000 horas de trabalho. Participam do projeto cerca de 20 pessoas, que são alocadas em times pequenos de no máximo 6 pessoas para a realização dos sub-projetos,

que duram em média 2 ou 3 meses, realizados em *sprints* de 4 semanas. A caracterização geral dos sub-projetos inseridos no contexto do projeto pode ser vista na Tabela 5.1.

Tabela 5.1: Caracterização dos Sub-Projetos do Projeto Piloto.

Tipo	Software Embarcado
Restrições	Preço fixo + Prazo limitado + Escopo flexível
Duração	2 ou 3 meses. <i>Sprints</i> com duração 4 semanas
Estimativa	<i>Story Points</i> + <i>Use Case Points</i>
Tamanho da Equipe	Pequeno (até 6 pessoas)
Linha de Produto	Telefonia Móvel
Estabilidade dos Requisitos	Pequena (Requisitos muito voláteis)
Envolvimento do cliente	Médio
Complexidade do projeto	Grande

É importante destacar, que autora da abordagem não participou do projeto piloto, apenas participou da elaboração e acompanhamento da implantação da abordagem desempenhando o papel de integrante do grupo de processo da organização.

Nas próximas seções será mostrada a execução de cada fase da abordagem MiniDMAIC no projeto piloto selecionado.

5.3.3 Executando as Fases “Definir” e “Medir” do MiniDMAIC

Todos os projetos de desenvolvimento de software da organização coletam e analisam mensalmente os indicadores de projeto no Relatório de Desempenho do Projeto. Um dos indicadores da organização que possui processos e subprocessos controlados estatisticamente é o indicador de densidade de defeitos.

No projeto em que foi executado o MiniDMAIC, para relato de experiência deste trabalho, identificou-se uma grande quantidade de defeitos em testes sistêmicos e verificou-se

que o indicador densidade de defeitos em testes sistêmicos estava acima dos limites de *baseline* da organização, como mostrado no gráfico de controle¹⁵ na Figura 5.1.

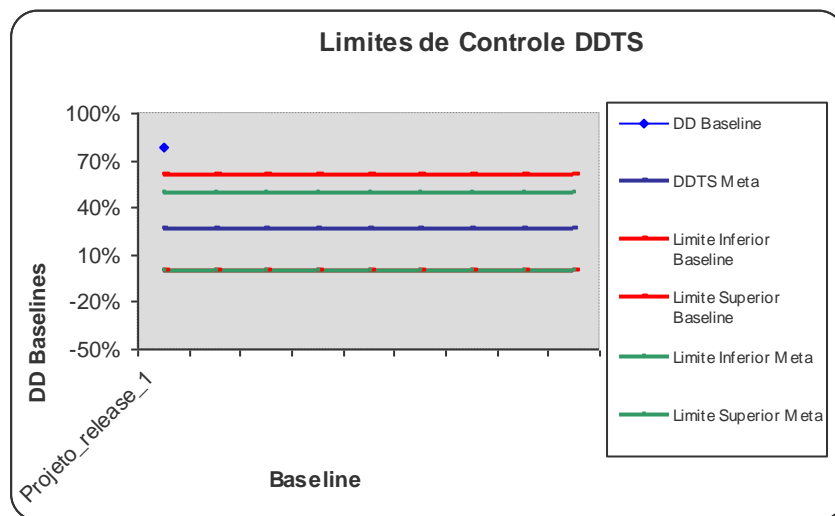


Figura 5.1: Gráfico de Controle do Projeto para *Baseline* Densidade de Defeitos em Testes Sistêmicos.

Desta forma, foi identificada a necessidade de abrir uma ação MiniDMAIC para o projeto a fim de analisar a causa raiz dos defeitos do projeto.

A organização possui uma base histórica de projetos passados localizada em uma ferramenta de gestão de conhecimento, com acesso para todas as pessoas da organização. Essa base histórica contém informações como: informações gerais dos projetos, indicadores dos projetos, lições aprendidas, riscos e MiniDMAICs abertos pelos projetos.

Inicialmente, foi verificado a base histórica da organização para encontrar MiniDMAICs relacionados a densidade de defeitos que foram executados em outros projetos. Existiam dois MiniDMAICs relacionados a este problema que foram tomados como base para melhor execução e análise de causas no projeto.

Analisando a *baseline* de desempenho da densidade de defeitos em testes sistêmicos da organização, foi tomado como meta do projeto ficar dentro dos limites de especificação do projeto (limite superior e inferior meta), reduzindo a densidade de defeitos em 81% para atingir a meta de densidade de defeitos em testes sistêmicos que havia sido estabelecida. Não houve necessidade de identificar uma nova medição para medir o problema, pois o problema

¹⁵ Gráficos de Controle são utilizados para monitorar a variabilidade existente nos processos, permitindo distinguir causas comuns de causas especiais (SOFTEX, 2009b).

já estava caracterizado no indicador densidade de defeitos em testes sistêmicos, que já era considerado nos projetos da organização e que é controlado estatisticamente.

Em uma planilha foram coletados todos os defeitos relacionados ao escopo da *release* e estes defeitos foram classificados por criticidade, origem e tipo de defeito como mostrado na Figura 5.2. Esta classificação ajuda a conhecer a origem dos defeitos de acordo com a sua classificação e saber quais são as mais recorrentes. No caso do projeto, o maior número de defeitos tinha como criticidade *major*, origem na implementação e tipos de defeitos de funcionalidade e algoritmo.

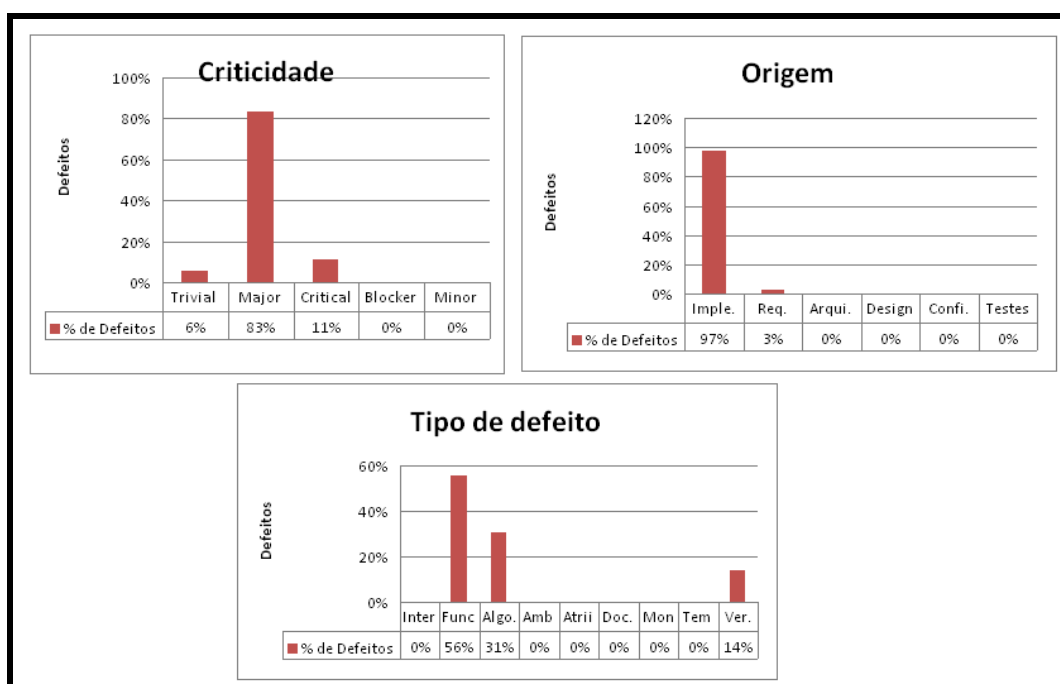


Figura 5.2: Classificação dos Defeitos Encontrados em Testes Sistêmicos do Projeto.

Nesta fase estabeleceu-se o seguinte:

- Meta: reduzir a densidade de defeitos em testes sistêmicos em 81%, ficando dentro dos limites de especificação do projeto.
- Processo(s) afetado(s): implementação.
- Riscos: Não foram identificados riscos relacionados ao problema.
- *Baseline* de desempenho organizacional: densidade de defeitos em testes sistêmicos
- Responsáveis pela fase: coordenador do projeto, líder técnico e *Quality Assurance*.

- Duração: 1 dia.

Durante a execução destas duas fases em paralelo, houve apenas dificuldade no processo de classificação dos defeitos, que demandou um esforço muito grande da equipe para análise destes defeitos.

5.3.4 Executando Fase “Analisar” do MiniDMAIC

Nesta fase foram alocados especialistas para que os defeitos encontrados fossem analisados. No caso da ação MiniDMAIC no projeto piloto, foram alocados os seguintes especialistas: coordenador do projeto, líder técnico, *Quality Assurance*, desenvolvedores, analista de requisitos e analista de teste.

Com base na classificação dos defeitos da fase medir e no agrupamento dos defeitos recorrentes, foi feito um *brainstorming* com a equipe do projeto, visando encontrar a causa raiz dos defeitos. O *brainstorming* foi composto por duas reuniões com a equipe para identificar e priorizar as causas do problema. Na primeira reunião a equipe teve como insumo a coleta de defeitos da fase medir e sua classificação, e foram coletadas idéias de possíveis causas sem se preocupar se aquelas causas realmente eram causas raiz do problema.

Após a identificação das causas, foi analisado cada defeito para conhecer com qual causa estava ligado. Assim, foram identificadas quais as causas se repetiam mais quando consolidadas por defeitos. Com base nessa consolidação, foi realizada uma segunda reunião com a equipe e mostradas as causas consolidadas para priorizar as causas raízes do problema. Todas as causas identificadas foram priorizadas pela equipe. Foram identificadas e priorizadas, com auxílio do gráfico de pareto, as seguintes causas no *brainstorming*:

- **Causa 1:** componentes arquiteturais desenvolvidos em paralelo aos casos de uso.
- **Causa 2:** *baseline* gerada sem realização de testes em ambiente idêntico ao de produção.
- **Causa 3:** falta de entendimento dos requisitos pelos desenvolvedores.
- **Causa 4:** escopo da *sprint* mal dimensionado (estimativa e sequência de desenvolvimento dos casos de uso).
- **Causa 5:** arquitetura não adequada ao desenvolvimento concorrente do time.

Analisando as causas levantadas e priorizadas relacionadas com os problemas encontrados na iteração, observa-se que:

- O planejamento foi mal dimensionado. Muitos casos de uso foram planejados para um curto período de tempo (tempo fixo de 4 semanas). Visando cumprir o escopo definido para a iteração, algumas atividades essenciais para a qualidade final do produto não foram executadas de acordo com a estimativa planejada. Entre elas pode-se citar a realização de testes de integração e testes no dispositivo móvel.
- O time não tinha total conhecimento dos requisitos do projeto. Era a primeira *sprint* do projeto, e não foi realizada nenhuma reunião ou *workshop* com os desenvolvedores para nivelamento e discussão dos requisitos. Os artefatos de definição de requisitos foram definidos, porém não foram seguidos.
- A arquitetura inicialmente definida não estava madura, acarretando vários problemas e esforços adicionais para o desenvolvimento.

Em seguida, foi feito um *brainstorming* em uma reunião para identificar possíveis ações para tratamento das causas. Foram identificadas as seguintes ações:

- **Ação 1:** realizar testes de integração antes dos testes sistêmicos.
- **Ação 2:** realizar *workshop* de requisitos para nivelamento e entendimento dos casos de uso pelo time do projeto.
- **Ação 3:** realizar testes do caso de uso em ambiente idêntico ao de produção.
- **Ação 4:** definir e divulgar o conceito de "*done*" para a conclusão da implementação do caso de uso.
- **Ação 5:** melhorar planejamento das próximas iterações com participação do time (o planejamento deveria prever o desenvolvimento e integração dos componentes arquiteturais antes do desenvolvimento dos casos de uso).
- **Ação 6:** realizar *refactoring* do componente da arquitetura.

Na Tabela 5.2 podemos verificar a relação entre as causas identificadas e as ações priorizadas para o seu tratamento.

Tabela 5.2: Relacionamento das Causas e Ações Identificadas para Tratamento das Causas dos Defeitos.

Causas	Ações
Causa 1	Ação 1, Ação 3, Ação 4
Causa 2	Ação 1, Ação 3, Ação 4
Causa 3	Ação 2
Causa 4	Ação 5
Causa 5	Ação 6

A fase “Analisar” do MiniDMAIC no projeto foi bem detalhada, tendo sido analisados todos os defeitos encontrados para maior eficácia da ação. Além disso, focou-se bem nas causas raízes dos defeitos para não atacar causas erradas. A fase teve duração de dois dias. Apesar disso, houve dificuldade da equipe do projeto em entender o que realmente era causa raiz dos defeitos, tendo sido necessário o apoio do *Quality Assurance* para direcionar a equipe e focar nas causas do problema.

5.3.5 Executando a Fase “Melhorar” do MiniDMAIC

Todas as ações identificadas no *brainstorming* foram consideradas importantes de serem implementadas e eram ações de fácil implementação. Foi definido no Jira um plano de ação para que as ações identificadas para tratamentos das causas fossem implementadas. Cada ação foi incluída na ação MiniDMAIC no Jira como uma *sub-task* do MiniDMAIC. Para cada ação foram atribuídos responsáveis para resolução da ação e determinado um prazo para término da ação dentro do projeto. Nesta fase, todos os especialistas alocados na fase analisar tiveram participação. Abaixo segue uma descrição da execução das ações:

- **Ação 1:** A equipe executou os testes de integração nas *sprints* 2 e 3, antes dos testes sistêmicos. Verificou-se que a equipe de desenvolvimento conseguiu identificar praticamente a mesma quantidade de problemas que a equipe de testes sistêmicos, comprovando a eficácia da ação.
- **Ação 2:** foi realizado o *workshop* de requisitos nas *sprints* 2 e 3 com a participação da equipe de requisitos, IHC, testes e desenvolvimento. Durante a execução da ação houve um repasse de entendimento dos requisitos da equipe de requisitos para

o restante da equipe. A prática contribuiu bastante para o nivelamento da equipe em relação ao entendimento dos requisitos e foram apontadas alterações necessárias nos requisitos que não tinham sido pensadas anteriormente.

- **Ação 3:** Na primeira execução desta ação verificou-se um impedimento. Como os testes do caso de uso não tinham sido feitos em ambiente idêntico ao de produção, foi encontrado um erro que impedia o teste. Além disso, nem toda equipe possuía os celulares para rodar o teste, o que limitou a execução da ação. O erro que impedia o teste foi corrigido e os testes do caso de uso passaram a ser executados nas *sprints* 2 e 3.
- **Ação 4:** na reunião de planejamento da *sprint* 2 do projeto, o conceito de “*done*” foi definido em conjunto com o time e divulgado com todos, através de ata e cartazes afixados na sala do projeto. Essa prática foi utilizado durante as *sprints* 2 e 3. Os conceitos de “*done*” definidos foram:
 - Requisitos: casos de uso concluídos e revisados com reajustes.
 - Análise e Projeto: diagrama de classes concluído e revisado com reajustes.
 - Codificação: código gerado e revisado com reajustes e testes unitários codificados e documentos com 75% de cobertura.
- **Ação 5:** melhorar o planejamento das próximas iterações com a participação do time (o planejamento deveria prever o desenvolvimento e integração dos componentes arquiteturais antes do desenvolvimento dos casos de uso). A melhoria dos planejamentos teve início na *sprint* 2 do projeto. Para esta *sprint* foi feita reunião de planejamento junto com a equipe do projeto e foi detalhada em ata. No planejamento, o desenvolvimento e a integração dos componentes arquiteturais foram planejados para iniciarem antes do desenvolvimento dos casos de uso. Além disso, tanto as atividades de *refactoring* de casos de uso quanto as atividades para entendimento dos requisitos executadas de acordo com a Ação 3 foram planejadas para serem realizadas inicialmente. Durante a *sprint* 3 a mesma ação foi realizada novamente.

- **Ação 6:** esta ação foi planejada na execução da Ação 5 e o *refactoring* do componente da arquitetura foi realizado pela equipe do projeto, melhorando a manutenibilidade da aplicação.

A equipe teve dificuldade em implantar a ação 3 devido à indisponibilidade de dispositivo com o ambiente idêntico ao de produção para toda a equipe. As outras ações foram implementadas com mais facilidade pelo time do projeto. Em média, a implementação das ações teve duração de duas semanas.

5.3.6 Executando a Fase “Controlar” do MiniDMAIC

Após a implementação das ações para tratamento das causas dos defeitos, foram medidos os resultados para analisar o grau de eficácia conseguido. Na segunda *sprint* do projeto foi medido o resultado e identificado que obteve-se uma melhoria de 38% no indicador de densidade de defeitos em testes sistêmicos e que o resultado atendia aos limites de especificação do projeto. Apesar disso, a meta estabelecida de 81% não foi atendida. Então, optou-se executar a fase melhorar, implementando as mesmas ações no *sprint* 3, e medir o resultado novamente para verificar se as ações realmente eliminaram as causas raízes dos defeitos.

Na *sprint* 3 foi medido novamente o indicador de densidade de defeitos em testes sistêmicos e foi verificado uma melhoria ainda maior, ficando bem próximo da meta estabelecida para o projeto. Apesar da meta não ser atendida no resultado da *sprint* 3, os resultados esperados foram considerados satisfatórios e foi verificado nos dois *sprints* posteriores dos projetos que as causas dos defeitos foram realmente tratadas. A melhoria na terceira *sprint* foi de 51%. Na Figura 5.3 é mostrado um gráfico de controle ilustrando a melhoria obtida pelo projeto ao longo das *sprints*.

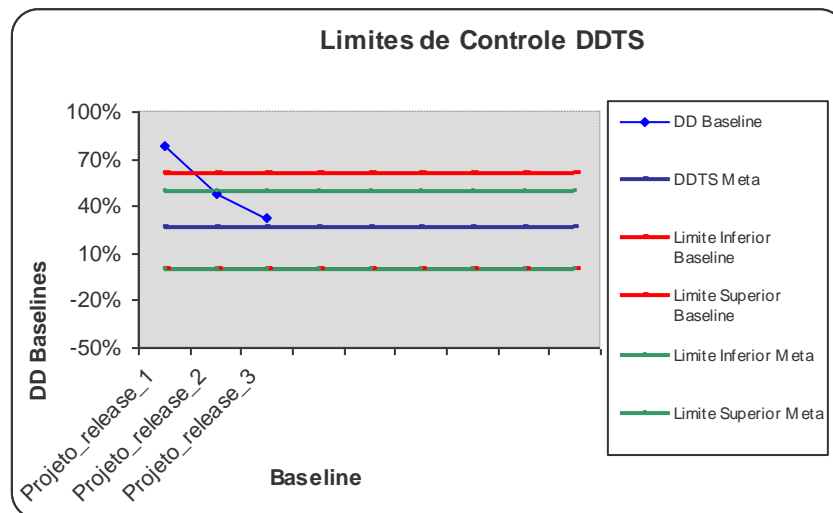


Figura 5.3: Gráfico de Controle do Projeto para *Baseline* Densidade de Defeitos em Testes Sistemáticos com os Resultados Finais após Execução da Ação MiniDMAIC.

Após a comprovação das melhorias implantadas, foi realizada uma reunião com a equipe para coletar lições aprendidas da equipe e fechar a ação com os resultados coletados. Como principal lição aprendida da execução da análise de causas no projeto, foi verificado que na primeira *sprint* era importante estabelecer um escopo mínimo que permitisse a construção da arquitetura e conhecimento do time sobre o domínio de negócio da aplicação que estava sendo desenvolvida.

Após o fechamento da ação, o coordenador do projeto enviou todo o insumo de execução da ação MiniDMAIC para base histórica da organização, através de uma ação no Jira.

Devido ao projeto ter retornado à fase “Melhorar” para executar as ações na *sprint* 3 do projeto, o MiniDMAIC no projeto teve uma duração mais longa, aproximadamente 6 semanas. Essa estratégia de executar novamente a fase “Melhorar” na próxima *sprint* do projeto foi escolhida pela equipe para verificar se realmente as ações estavam sendo eficazes e para eliminar as causas raízes do problema. Caso o projeto obtivesse uma melhoria realmente comprovada no primeiro momento, a duração da ação MiniDMAIC seria em média de duas a três semanas.

5.3.7 Disponibilizar as Oportunidades de Melhoria para os Ativos Organizacionais

Todas as ações MiniDMAICs da organização são analisadas e consolidadas pelo grupo de processos e grupo de medição e análise da organização. A ferramenta Jira gera um documento, em formato *Word*, de toda execução da ação MiniDMAIC. Este documento é enviado à base histórica pelo projeto e publicado em uma ferramenta de gestão de conhecimento para ser possível ser consultado por todos os projetos da organização.

Para facilitar o acompanhamento de todas as ações MiniDMAICs pelo grupo de processo, algumas informações consideradas mais importantes são consolidadas em uma planilha. A Tabela 5.3 mostra as informações consolidadas com o insumo do MiniDMAIC executado no projeto ilustrado neste trabalho.

Tabela 5.3: Informações Consolidadas dos MiniDMAICs.

Tipo de Problema	Causas do Problema	Ações Tomadas para Tratamento da Causa	Melhoria Obtida
Alta Densidade de Defeitos em Testes Sistêmicos	<ul style="list-style-type: none"> - Causa 1: componentes arquiteturais desenvolvidos em paralelo aos casos de uso. - Causa 2: <i>baseline</i> gerada sem realização de testes em ambiente idêntico ao de produção. - Causa 3: falta de entendimento dos requisitos pelos desenvolvedores. - Causa 4: escopo da <i>sprint</i> mal dimensionado (estimativa e seqüência de desenvolvimento dos casos de uso). - Causa 5: arquitetura não adequada ao 	<ul style="list-style-type: none"> - Ação 1: realizar testes de integração antes dos testes sistêmicos. - Ação 2: realizar <i>workshop</i> de requisitos para nivelamento e entendimento dos casos de uso pelo time do projeto. - Ação 3: realizar testes do caso de uso em ambiente idêntico ao de produção. - Ação 4: definir e divulgar o conceito de "done" para a conclusão da implementação do caso de uso. - Ação 5: melhorar planejamento das próximas iterações com participação do time (O planejamento deve prever o desenvolvimento e integração dos componentes 	Redução da Densidade de defeitos em 51%

	desenvolvimento concorrente do time.	arquiteturais antes do desenvolvimento dos casos de uso). - Ação 6: realizar <i>refactoring</i> do componente da arquitetura.	
--	--------------------------------------	--	--

Em um projeto DMAIC da organização relacionado à diminuição do retrabalho nos projetos, verificou-se que a maior parte do retrabalho de um projeto estava relacionada à correção de defeitos. Para descobrir as causas deste problema, optou-se por analisar os MiniDMAICs da organização relacionados com a alta densidade de defeitos em testes sistêmicos. Verificou-se que algumas causas dos defeitos se repetiam na organização como, por exemplo: arquitetura não adequada e falta de entendimento dos requisitos pelos desenvolvedores. Na fase “Medir” do DMAIC todas as causas dos defeitos identificadas nos MiniDMAICs dos projetos foram consolidadas. Após a consolidação, foi feita uma análise e tomada de decisão para priorização das principais causas dos defeitos. As causas mencionadas acima fizeram parte da lista de causas priorizadas no DMAIC organizacional.

Como já existiam ações implementadas em projetos e que foram eficazes, as seguintes ações foram priorizadas pelo projeto de melhoria: definir e divulgar o conceito de "done" para a conclusão da implementação do caso de uso e realizar *workshop* de requisitos para nivelamento e entendimento dos casos de uso pelo time do projeto. Estas ações estão sendo avaliadas em outros projetos pilotos da organização e se comprovadas, tais melhorias serão disponibilizadas para todos os projetos da organização, fazendo parte do processo padrão organizacional. Podemos concluir, então, que as mesmas causas e ações implantadas em um projeto podem contribuir na implantação de melhorias no nível da organização.

5.3.8 Lições Aprendidas e Oportunidades de Melhoria

Apesar de, a abordagem MiniDMAIC ter sido executada em outros projetos de software, durante a execução da ação do MiniDMAIC pelo projeto destacado neste trabalho, foram identificadas algumas das principais dificuldades e lições aprendidas:

- Não houve dificuldades na fase “Definir”, já que o processo de medição e análise é bem consolidado nos projetos da organização, e possui um controle estatístico de

processos e subprocessos relacionado aos defeitos em testes sistêmicos. Além disso, há uma *baseline* de desempenho da organização para o indicador densidade de defeitos em testes sistêmicos, bem como metas que dão subsídios para o projeto definir suas metas.

- Na fase “Medir” houve uma parada de toda equipe na coleta dos defeitos para classificá-los, demandando um esforço grande. Em um momento crítico do projeto isso não poderia ocorrer.
- A classificação dos defeitos foi essencial no entendimento dos defeitos pela equipe, auxiliando na identificação de causas raízes.
- Apesar da equipe do projeto ter recebido treinamento de análises de causas, de ferramentas estatísticas básicas (gráfico de pareto, diagrama de causa e efeito, gráfico de controle, etc.) e da execução do MiniDMAIC no Jira, a equipe teve dificuldade em identificar o que é realmente causa raiz do problema.
- O fato de a equipe ser experiente e já trabalharem juntos há muito tempo no mesmo projeto foi um facilitador na identificação e priorização de causas e ações, já que todos eles conheciam bastante o projeto. Este fator também facilitou na implementação das ações no projeto.
- A orientação do *Quality Assurance* em toda a execução do MiniDMAIC do projeto foi essencial para o sucesso da ação, visto que era a primeira vez que a análise de causas estava sendo executada no projeto.
- O fato de o projeto ter iterações curtas em *sprints* facilitou a medição dos resultados no projeto na fase “Controlar”, uma vez que toda *sprint* passa por todas as fases do ciclo de desenvolvimento e puderam ser medidos todos os indicadores do projeto.
- A análise dos vários MiniDMAIC é fundamental para a identificação de oportunidades de melhoria para os processos no nível organizacional. Com isso, percebeu-se que, de acordo com o nível de maturidade da organização, novas fontes de dados poderão agregar sobremaneira a melhoria dos processos. Estas novas fontes podem ser adicionadas à lista de dados que podem ser analisados, definida em Albuquerque (2008).

- A abordagem implementada na ferramenta Jira facilitou o uso do MiniDMAIC pelo projeto, pois esta já contém todos os campos obrigatórios para executar cada fase.
- Para a análise dos defeitos encontrados, a abordagem é bem estruturada, no entanto, poderia também ter orientações para análise dos outros indicadores que estão fora do controle estatístico do projeto (ponto fora dos limites de especificação ou limites de *baseline*) e problemas recorrentes que não fossem de origem dos defeitos em testes sistêmicos.
- Uma oportunidade de melhoria apresentada na avaliação CMMI nível 5 diz respeito à necessidade de na primeira bateria de testes dos projetos ser feita uma análise de causas dos defeitos. Caso o projeto não tenha ações a serem tomadas em decorrência dos defeitos, o MiniDMAIC poderia ser finalizado na fase “Analisar”.
- No *template* de análise de causas para defeitos em testes sistêmicos, não existia uma identificação de cada causa relacionada ao defeito encontrado e priorização das causas e ações. No projeto utilizado para experiência de uso foram implantadas as seguintes melhorias no *template*: foram relacionadas as causas para cada defeito, foram incluídos a tabela e gráfico para priorização das causas utilizando o gráfico de pareto, e foram relacionadas as causas priorizadas e ações, assim como a priorização das ações. As melhorias foram enviadas para o grupo de engenharia de processos disponibilizá-las para a organização.

5.4 CONCLUSÃO DO CAPÍTULO

Nesse capítulo foi apresentado a experiência de uso da abordagem MiniDMAIC em um projeto piloto de uma organização de maturidade nível 5 do CMMI, bem como, foram detalhados os critérios e condições que a organização escolheu para disparar uma ação MiniDMAIC. Finalmente, foram apresentadas as principais lições aprendidas advindas da execução da referida abordagem. O capítulo seguinte apresenta a conclusão deste trabalho bem como perspectivas de trabalhos futuros.

6 CONCLUSÃO E TRABALHOS FUTUROS

Neste capítulo são apresentadas as conclusões finais do trabalho, focando nas contribuições, limitações e trabalhos futuros.

6.1 CONSIDERAÇÕES FINAIS

O tratamento dos problemas e defeitos encontrados nos projetos de software ainda é deficiente na maioria das organizações. As análises em geral não focam suficientemente no problema e suas possíveis origens, levando a tomadas de decisões erradas, que acabam por não resolver o problema. Também é difícil a implementação de um processo de Análise e Resolução de Causas (CAR) em projetos, conforme prescrito pelo nível 5 do CMMI, devido aos recursos limitados com os quais os mesmos têm que trabalhar.

A abordagem proposta no trabalho visa minimizar estas dificuldades, propondo uma abordagem consistente de análise e resolução de causas baseada no método DMAIC que já está consolidado no mercado e é aderente a área de processo Análise e Resolução de Causas do CMMI.

O MiniDMAIC foi e está sendo aplicado em projetos de desenvolvimento de software do Instituto Atlântico, que foi avaliado CMMI nível 5 em agosto de 2009 e obteve o nível máximo de maturidade do modelo. Um dos fatores que auxiliaram ao cumprimento do nível 5 do CMMI em relação à área de processo Análise e Resolução de Causas, foi a implantação do MiniDMAIC no contexto de projetos de software da organização. Quatro projetos foram avaliados, todos estes projetos executaram a abordagem MiniDMAIC para análise de causas e não foram encontrados pontos fracos em nenhuma área de processo dos níveis 4 e 5 do CMMI durante a avaliação oficial.

Com base nos resultados alcançados e lições aprendidas deste trabalho, podemos concluir que a abordagem MiniDMAIC é útil para análise e resolução de causas no contexto de projetos de desenvolvimento de software, sendo ao mesmo tempo compatível com o DMAIC e a área de processos de Análise e Resolução de Causas do CMMI, considerando a aplicação apenas em projetos.

6.2 PRINCIPAIS CONTRIBUIÇÕES

Como principais contribuições do trabalho realizado durante esta pesquisa podemos destacar:

- A investigação da aderência da área de processo Análise e Resolução de Causas do CMMI com do método DMAIC do *Six Sigma*, baseando-se nos passos do método descrito por Tayntor (2003).
- Definição de uma abordagem para análise de causas aderente à área de processo Análise e Resolução de Causas do CMMI para o contexto de projetos de desenvolvimento de software.
- Implementação da abordagem em uma ferramenta *workflow* (Jira) para facilitar o uso da ação MiniDMAIC no projeto, possibilitando que todos os passos da abordagem fossem seguidos pelos executores e os campos obrigatórios fossem preenchidos.
- Execução e análise da abordagem em projetos de desenvolvimento de software de uma organização que obteve o CMMI nível 5.
- Integração da abordagem MiniDMAIC com os processos que lidam com a identificação e implementação de melhorias de processos no nível organizacional.

6.3 LIMITAÇÕES

O MiniDMAIC foi elaborado no contexto de organizações que possuem pelo menos o nível 4 de maturidade do CMMI, visto que as ações MiniDMAIC terão resultados ainda melhores, uma vez que diversos parâmetros para medir os resultados dos projetos já estarão

definidos, bem como a utilização de ferramentas de análise estatística já será prática comum na organização.

Isso não impede que ele seja aplicado em organizações que não tenham esse nível de maturidade. Porém, alguns passos desta abordagem que tratam de problemas relacionados ao desempenho do processo podem não ter o resultado esperado, devendo ser adaptados para a realidade da organização.

6.4 TRABALHOS FUTUROS

Durante o desenvolvimento e a aplicação da abordagem proposta, algumas possibilidades de trabalhos futuros foram identificadas:

- Implementação da abordagem no contexto da organização para análise de causas no nível organizacional, como: qualidade, gestão de processo, medição e análise e treinamento.
- Adequar a abordagem aos níveis 2 e 3 do CMMI, desassociando a abordagem do enfoque estatístico para que organizações com menor maturidade possam também utilizar a abordagem para análise das causas dos defeitos e problemas.
- Identificar outras técnicas de análise de causas para a fase “Analisar” da abordagem, de forma que possa facilitar a equipe de desenvolvimento a escolher uma técnica adequada para o projeto.
- Melhorar a disponibilização das melhorias do MiniDMAIC para a organização.
- Desenvolvimento da abordagem MiniDMAIC em uma ferramenta de apoio dentro do modelo de software livre, para que o uso da abordagem possa ser utilizado por outras organizações ou pelo meio acadêmico.

BIBLIOGRAFIA

- (AHERN et al., 2003) AHERN, D. M.; CLOUSE, A.; TURNER, R. **CMMI Distilled: A Practical Introduction to Integrated Process Improvement**. 2nd edition, Addison Wesley, 2003.
- (ALBUQUERQUE, 2008) ALBUQUERQUE, A. B. **Avaliação e Melhoria de Ativos de Processos Organizacionais em Ambientes de Desenvolvimento de Software**. Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2008.
- (ALLEN, 2006) ALLEN, T. T. **Introduction to Engineering Statistics and Six Sigma: Statistical Quality Control and Design of Experiments and Systems**. Springer, 2006.
- (ALTSHULLER, 1999) ALTSHULLER, G.. **Innovation Algorithm: TRIZ, systematic innovation and technical creativity**. Technical Innovation Ctr, 1999.
- (AMMERMAN, 1998) AMMERMAN, M. **The Root Cause Analysis Handbook: A Simplified Approach to Identifying, Correcting, and Reporting Workplace Errors**. Productivity Press, 1998.
- (ANDRIETTA, 2002) ANDRIETTA, J. M., AUCHICK, P. A. **A importância do Método Seis Sigma na Gestão da Qualidade Analisada sob**

uma Abordagem Teórica. Revista de Ciência & Tecnologia, v.11, n. 20. p 91-98. Jul/Dez 2002.

- (BAKER et al., 1990) BAKER, A. L.; BIEMAN, J. M.; GUSTAFSON, S. A.; MELTON, A.; WHITTY, R. A **philosophy for software measurement.** Journal of Systems and Software, vol. 12, 1990.
- (BANAS QUALIDADE, 2009) BANAS QUALIDADE. **Melhoria Contínua - Soluções de Problemas.** Quality News. São Paulo. Disponível em < <http://www.vceconsult.com.br/Empresanoticias.asp?codigo=178> >. Acesso em: 06 out, 2009.
- (BEZERRA et al., 2009) BEZERRA, C. I. M.; COELHO, C.; GONCALVES, F. M.; GIOVANO, C.; ALBUQUERQUE, A. B. **MiniDMAIC: Uma Abordagem para Análise e Resolução de Causas em Projetos de Desenvolvimento de Software.** VIII Simpósio Brasileiro de Qualidade de Software, Ouro Preto. Anais do VIII Simpósio Brasileiro de Qualidade de Software, 2009.
- (BLAUTH, 2003) BLAUTH, R. **Seis Sigma: uma estratégia para melhorar resultados.** Revista FAE Business, nº 5, Abril 2003.
- (BOHN,2000) BOHN, R. **Stop fighting fires.** V.78, p. 82-92. Boston: Havard Business Review, 2000.
- (BREYFOGLE, 2003) BREYFOGLE, F. W. **Six Sigma: smarter solutions using statistical methods.** 2nd edition. Austin: Wiley, 2003.
- (CABRERA, 2006) CABRERA, A.. **Dificuldades de Implementação de Programas Seis Sigma: Estudos de casos em empresas com diferentes níveis de maturidade.** Dissertação de Mestrado,

USP-SP, São Carlos, 2006.

- (CAMPOS, 1992) CAMPOS, V. F. **TQC: Controle da Qualidade Total (no estilo japonês)**. Vicente Falconi Campos. Belo Horizonte, MG: Fundação Christiano Ottoni. Escola de Engenharia da UFMG. Rio de Janeiro: Bloch Ed,1992.
- (CARD, 2005) CARD, D. N. **Defect Analysis: Basic Techniques for Management and Learning**. Advances in Computers 65, 2005.
- (CHILLAREGE et al., 1992) CHILLAREGE, R. et al. **Orthogonal Defect Classification: a Concept for in-Process Measurements**. IEEE Transactions on SE, v.18, n. 11, pp 943-956, Nov, 1992.
- (CHILLAREGE, 2006) CHILLAREGE, R. **ODC - a 10x for Root Cause Analysis**. Proceedings RAM *Workshop*, Berkeley CA, 2006
- (CHRISISSIS et al., 2006) CHRISISSIS, M. B.; KONRAD, M.; SHRUM, S. **CMMI: Guidelines for Process Integration and Product Improvement**. 2nd edition, Boston, Addison Wesley, 2006.
- (CHRISTMANSSON e CHILLAREGE, 1996) CHRISTMANSSON, J., CHILLAREGE, R. **Generation of an Error Set that Emulates Software Faults**. In: Proc. Of the 26st IEEE Fault Tolerant Computing Symposium, FCTS' 26, Sendai, Japan, 1996.
- (CMMI, 2006) CMMI-DEV, **CMMI for Development**, V1.2 model, CMU/SEI-2006-TR-008. Software Engineering Institute, 2006.
- (DEMING, 1986) DEMING, W. E. **Out of Crises**, Massachusetts Institute of Technology, Center of Advanced Engineering, Cambridge,

1986.

- (DEMING, 1990) DEMING, W. E. **Qualidade: A revolução da administração.** Rio de Janeiro, Marques-Saraiva, 1990.
- (DENNIS, 1994) DENNIS, M. **The Chaos Study**, The Standish Group International, 1994.
- (DENOVE e POWER, 2006) DENOVE, C.; POWER, J. **Satisfaction: How Every Great Company Listens to the Voice of the Customer.** Portfolio, 2006.
- (ECKES, 2001) ECKES, G. **A Revolução Seis Sigma: o método que levou a GE e outras empresas a transformar processos em lucros.** Rio de Janeiro, Campos, 2001.
- (ENDRES, 1975) ENDRES, A., **An Analysis of Errors and Their Causes in Systems Programs**, IEEE Transactions on Software Engineering, SE-1, 2, June 1975, pp. 140-149, 1975.
- (FENTON e PFLEEGER, 1997) FENTON, N.; PFLEEGER, S.L. **Software metrics: a rigorous & practical approach.** PWS Publishing Company, 2nd edition, 1997.
- (FLORAC e CARLETON, 1999) FLORAC, W. A.; CARLETON, A. D. **Measuring the software process: statistical process control for software process improvement.** Addison-Wesley, 1999.
- (GEORGE et al., 2005) GEORGE, M. L; ROWLANDS, D.; PRICE, M.; MAXEY, J.; JAMINET, P.; WATSON-HEMPHILL, K.; COX, C.. **The**

Lean Six Sigma Pocket Toolbook: A Quick Reference Guide to Nearly 100 Tools for Improving Process Quality, Speed, and Complexity. New York, McGraw-Hill, 2005.

(GONÇALVES et al., 2008a) GONÇALVES, F., BEZERRA, C., BELCHIOR, A., COELHO, C., PIRES, C., **Implementing Causal Analysis and Resolution in Software Development Projects: The MiniDMAIC Approach**, 19th Australian Conference on Software Engineering, pp. 112-119, 2008.

(GONÇALVES, 2008b) GONÇALVES, F. **Abordagem para Análise e Resolução de Causas de Problemas Aplicando Multicritério**. Dissertação de Mestrado da UNIFOR, Fortaleza, 2008.

(GUPTA, 2006) GUPTA, P. **Six Sigma Business Scorecard – Creating a comprehensive corporate performance measurement system**. 2nd edition, McGraw-Hill, 2006.

(HALL e FELTON, 1997) HALL, T.; FENTON, N. **Implementing effective software metrics programs**. IEEE Software, v. 14, n. 2, Mar/Apr 1997.

(HARRY e SCHROEDER, 2000) HARRY, M.; SCHROEDER, R. **Six Sigma: the breakthrough management strategy revolutionizing the world's top corporations**. New York, Currency Publishers, 2000.

(IEEE, 1994) **IEEE standard classification for software anomalies**. IEEE Std 1044-1993. 2 Jun 1994.

(ISHIKAWA, 1998) ISHIKAWA, K. **Controle de Qualidade Total – À Maneira Japonesa**. Rio de Janeiro, Campus, 6^a edição, 1998.

(ISO/IEC 12207, 2008) ISO/IEC 12207:1995/Amd 2:2008, **Information Technology - Software Life Cycle Process**, Amendment 2. Genebra: ISO,

2008.

- (JIRA, 2009) Atlassian Software Systems Pty Ltd. All rights reserved. Copyright © 2009. <http://www.atlassian.com/software/jira/>. Acessado em 02/11/09.
- (JOGLEKAR, 2003) JOGLEKAR, A. M. **Statistical Methods for Six Sigma: In R&D and Manufacturing**. Wiley-Interscience, 2003.
- (KALINOVSKI et al., 2008) KALINOWSKI, M., TRAVASSOS, G. H., CARD, D. N., **Guidance for Efficiently Implementing Defect Causal Analysis**, VII Simpósio Brasileiro de Qualidade de Software (SBQS), Florianópolis, 2008.
- (KALINOVSKI, 2009) KALINOWSKI, M. **DBPI: Abordagem de Prevenção de Defeitos de Software para Apoiar Melhoria de Processos e Aprendizado Organizacional**. Exame de Qualificação UFRJ, Rio de Janeiro, 2009.
- (KASSE, 2004) KASSE, T. **Practical Insight into CMMI**. Cambridge, Massachusetts, Artech House Publishers, 2004.
- (KULPA et al., 2003) KULPA, MARGARET K.; JOHNSON, KENT A. **Interpreting the CMMI: a process improvement approach**. Florida, Auerbach, 2003.
- (LARSON, 2003) LARSON, A. **Demystifying Six Sigma: A Company-Wide Approach to Continuous Improvement**. New York, Amacon, 2003.

- (LIKER, 2004) LIKER, J. **The Toyota Way: 14 Management ciles from the World's Greatest Manufacturer**. Mc Graw-Hill, USA, 2004.
- (MONTGOMERY, 2004) MONTGOMERY, D. C. **Introdução ao Controle Estatístico da Qualidade**. 4ª edição, Rio de Janeiro, LTC, 2004.
- (MUTAFELIJA et al., 2003) MUTAFELIJA, B.; STROMBERG, H. A. **Systematic Process Improvement Using Iso 9001: 2000 and CMMI**. Artech House, 2003.
- (PALADY, 1997) PALADY, P. **FMEA: Análise dos Modos de Falha e Efeito – Prevendo e prevenindo problemas antes que ocorram**. Instituto IMAM, 1997.
- (PANDE, 2001) PANDE, S. **Estratégia Six Sigma: como a GE, a Motorola e outras grandes empresas estão aguçando seu desempenho**. Rio de Janeiro, Qualitymark, 2001.
- (PANG, 2002) PANG, A. **What is bug ?**. Software Development Canada, Canada. Disponível em : <http://www.softwaredevelopment.ca/bugs.shtml>. Acesso em 15/10/2009.
- (PMI, 2008) PMI - Project Management Institute. **A Guide to the Project Management Body of Knowledge**, 4a. edição, EUA, 2008.
- (PYZDEK, 2003) PYZDEK, T. **Uma ferramenta em busca do defeito zero**. HSM Management, São Paulo, HSM do Brasil, n. 38, 2003.
- (ROBITAILLE, 2004) ROBITAILLE, D. **Root Cause Analysis: Basic Tools and Techniques**. Chico, CA: Paton Press, 2004.

- (ROTONDARO et al., 2002) ROTONDARO, G. R.; RAMOS, A. W.; RIBEIRO, C. O.; MIYAKE, D. I.; NAKANO, D.; LAURINDO, F. J. B.; HO, L. L.; CARVALHO, M. M.; BRAZ, A. A.; BALESTRASSI, P. P. **Seis Sigma: Estratégia Gerencial para Melhoria de Processos, Produtos e Serviços**, São Paulo, Atlas, 2002.
- (SCHWABER, 2004) SCHWABER, K. **Agile Project Management with Scrum**, Microsoft, 2004.
- (SEI, 2006) SEI - Software Engineering Institute. **CMMI-DEV: CMMI for Development. V1.2 model**, CMU/SEI-2006-TR-008, <http://www.sei.cmu.edu/cmmi/general/>, December 2006.
- (SIVIY et al., 2005) SIVIY, J. M.; PENN, L. M.; HAPPER, E. **Relationship Between CMMI and Six Sigma**. Technical Note, CMU / SEI - 2005-TN-005, 2005.
- (SIVIY et al., 2007) SIVIY, J. M.; PENN, L. M.; STODDARD, R. W. **CMMI and Six Sigma; Partners in Process Improvement**. Addison-Wesley, 2007.
- (SMITH e ADAMS, 2000) SMITH, B.; ADAMS, E. **LeanSigma: advanced quality**. Proc. 54th Annual Quality Congress of the American Society for Quality, Indianapolis, Indiana, 2000.
- (SNEE, 1999) SNEE, R. D. **Why Should Statistician Pay Attention To Six Sigma?** Quality Progress, Vol. 32 N° 9, September 1999.
- (SOFTEX, 2009a) SOFTEX. **MPS.BR – Guia Geral: 2009**, setembro 2009. Disponível em www.softex.br. Acesso em: 15 nov. 2009.

- (SOFTEX, 2009b) SOFTEX. **Guia de Implementação – Parte 6: Nível B**, 2009. Disponível em: <http://www.softex.br/mpsbr>. Acesso em: 15 nov. 2009.
- (SOFTEX, 2009c) SOFTEX. **Guia de Implementação – Parte 7: Nível A**, 2009. Disponível em: <http://www.softex.br/mpsbr>. Acesso em: 15 nov. 2009.
- (STAMATIS, 2004) STAMATIS, D. H. **Six Sigma Fundamentals - A Complete Guide to the System, Methods and Tools**. New York, 2004.
- (SULLIVAN e CHILLAREGE, 1991) SULLIVAN, M.; CHILLAREGE, R. **Software Defects and their Impact on Systems Availability – A Study of field failures on Operating Systems**. In: Proc. Of the 21st IEEE Fault Tolerant Computing Symposium, FCTS' 91, pp. 2-9, 1991.
- (SULLIVAN e CHILLAREGE, 1992) SULLIVAN, M.; CHILLAREGE, R. **A comparison of Software Defects in Database Management Systems and Operating Systems**. In: Proc. Of the 22st IEEE Fault Tolerant Computing Symposium, FCTS' 92, Sendai, Japan, pp. 475-484, 1992.
- (TAYNTOR, 2003) TAYNTOR, C. B. **Six Sigma Software Development**. Flórida, Auerbach, 2003.
- (VIEIRA, 2006) VIEIRA, S. **Análise de Variância: (Anova)**. Atlas, 2006.
- (WATSON, 2001) WATSON, G. H. **Cycles of learning: observations of Jack Welch**. ASQ Publication, 2001.

(WERKEMA, 2006)

WERKEMA, M. C. C. **Ferramentas estatísticas básicas para a gerenciamto de processos**. Vol. 2. Belo Horizonte, MG. Fundação Christiano Ottoni, Escola de Engenharia da UFMG, 1995.

(WERKEMA, 1996)

WERKEMA, M. C; Aguiar, S. **Planejamento e Análise de Experimentos: Como identificar as principais variáveis influentes em um processo**. Belo Horizonte, Fundação Christiano Ottoni, vol. 8, 1996.

APÊNDICE A – PASSOS DO MINIDMAIC NA FERRAMENTA JIRA

Os passos da abordagem MiniDMAIC foram implementados no Jira, ferramenta já utilizada na organização que foi realizada a experiência de uso. Nas seções a seguir são detalhadas as telas de cada fase da abordagem e os campos a serem preenchidos. Alguns campos iniciados com (*) nas telas das fases da ação MiniDMAIC no Jira possuem preenchimento obrigatório.

B.1. FASE: DEFINIR

A fase “Definir” é uma fase de planejamento da ação e consiste na definição do problema, fontes, processos e subprocessos impactados, e resultados esperados, além da formação da equipe.

Step 2 of 2: Enter the details of the issue...

Project: Teste

Issue Type: MiniDMAIC

* Summary:

Description:

Organizational Performance Baseline:

Goals:

MiniDMAIC goals

Affected Process:

- None
- Análise e Projeto
- Análise e Tomada de Decisão
- Comercial
- Compras

* Assignee: [Assign to me](#)

Participants:

Start typing to get a list of possible matches.

Due Date:

Security Level:

Figura B.1: Tela Inicial de Criação da Ação MiniDMAIC no Jira.

Na Figura B.1 podemos observar as informações necessárias para que seja dado início a um MiniDMAIC:

- **Project:** este campo apresenta o nome do projeto e é preenchido automaticamente, de acordo com o projeto em questão;
- **Issue Type:** para iniciar uma ação MiniDMAIC, deve ser selecionada a opção “MiniDMAIC”;
- ***Summary:** este campo deve conter o problema a ser tratado.
- **Description:** este campo deve conter uma descrição detalhada do problema, citando desde quando o mesmo existe e qual o seu impacto ou consequências sobre o projeto. Esta descrição deve estar focando somente nos sintomas e não em causas ou soluções;
- **Organizational Performance Baseline:** este campo deve mostrar qual é a *baseline* de desempenho organizacional relacionada ao problema que está sendo tratado. A *baseline* de desempenho organizacional servirá de parâmetro para as metas de melhoria que serão estabelecidas posteriormente;

- ***Goals:** este campo deverá conter as metas e objetivos a serem atingidos com a execução da ação MiniDMAIC. Estas metas devem ser quantificáveis e viáveis;
- **Affected Process:** neste campo devem ser selecionados os processos que serão afetados com a execução da ação MiniDMAIC. Podem ser selecionados um ou mais processos;
- ***Assign To:** este campo deve conter o nome do responsável que estará conduzindo a ação MiniDMAIC. Esta função geralmente será conduzida pelo Coordenador do Projeto. Caso seja julgado necessário, pode ser designado um colaborador com formação de *Green Belt* (o QA do projeto, por exemplo) para apoiar o Coordenador;
- **Participants:** este campo deve listar as pessoas que participarão da ação MiniDMAIC, além do responsável pela sua condução. Podem ser selecionadas pessoas da equipe do projeto que estejam relacionadas direta ou indiretamente no problema a ser tratado, além de pessoas de fora do projeto que detenham conhecimento e possam colaborar na condução da ação MiniDMAIC;
- **Due Date:** Este campo deve conter o prazo máximo para término da ação. Esse período geralmente varia de 1 a 6 semanas, de acordo com a complexidade do problema e o tempo necessário para se obter os resultados esperados.
- **Security Level:** Todas as ações da organização possuem um nível de segurança, para ser visualizado apenas internamente ou também pode ser visível para o cliente.

B.2. FASE: MEDIR

A Figura B.2 apresenta a tela com o segundo passo da ação MiniDMAIC. A forma de preenchimento de cada campo será detalhada a seguir.

Figura B.2: Tela do Final da Fase “Medir” da Ação MiniDMAIC no Jira.

- ***Attachment (*Measurement Artifacts*):** este campo deve conter os documentos utilizados para medir o problema (classificação de problemas, Diagrama de Pareto).
- **Comment:** este campo pode ser utilizado caso haja necessidade de algum comentário adicional.
- **Viewable by:** neste campo são definidos os grupos que possuem permissão para visualizar os comentários. Como padrão, todos os participantes do projeto possuem acesso.

B.3. FASE: ANALISAR

A Figura B.3 apresenta a tela com o terceiro passo da ação MiniDMAIC. A forma de preenchimento de cada campo será detalhada a seguir.

Figura B.3: Tela do Final da Fase “Analisar” da Ação MiniDMAIC no Jira.

- ****Attachment (Analysis Tool):*** este campo deve conter os documentos resultantes da análise da causa raiz do problema (Ex: diagramas de causa-efeito).
- ****Analysis Participants:*** este campo deve conter as pessoas que participaram do processo de análise das causas do problema.
- ****Analysis Results (Root Cause):*** neste campo deve estar descrita qual a causa raiz do problema.
- ****Candidate Actions:*** neste campo podem ser incluídas algumas possíveis ações para tratamento da causa raiz.
- ****Comment:*** este campo pode ser utilizado caso haja necessidade de algum comentário adicional.
- ****Viewable by:*** neste campo são definidos os grupos que possuem permissão para visualizar os comentários. Como padrão, todos os participantes do projeto possuem acesso.

B.4. FASE: MELHORAR

A Figura B.4 apresenta a tela com o quarto passo da ação MiniDMAIC. A forma de preenchimento de cada campo será detalhada a seguir.

end improve

Prioritized Actions (attach ATD)

Attachment: Procurar...

The maximum file upload size is 10,00 Mb. Please zip files larger than this.

*** YOU MUST CREATE MINIDMAIC SUBTASKS FOR THE PRIORITIZED ACTIONS ***

Comment: (an optional comment describing this update)

Comment:

Viewable By: All Users

end improve Cancel

Figura B.4: Tela do Final da Fase “Melhorar” da Ação MiniDMAIC no Jira.

- ***Attachment (Prioritized Actions):** Caso seja necessário priorizar as ações de melhoria que serão implementadas, adicionar a análise e tomada de decisão da priorização.
- **Comment:** este campo pode ser utilizado caso haja necessidade de algum comentário adicional.
- **Viewable by:** neste campo são definidos os grupos que possuem permissão para visualizar os comentários. Como padrão, todos os participantes do projeto possuem acesso.
- **Subtasks (Action Plan):** deve-se criar uma subtarefa para cada ação do plano de ação.

B.5. FASE: CONTROLAR

A Figura B.5 apresenta a tela com o quinto passo da ação MiniDMAIC. A forma de preenchimento de cada campo será detalhada a seguir.

end control

Results Documents

Attachment:
The maximum file upload size is 10,00 Mb. Please zip files larger than this.

Results:

Comment: (an optional comment describing this update)

Comment:

Viewable By:

Figura B.5: Tela do Final da Fase “Controlar” da Ação MiniDMAIC no Jira.

- ***Attachment (Results Documents):** este campo deve conter documentos que mostrem resultados obtidos com a execução das ações MiniDMAIC (Ex.: Relatórios).
- ***Results:** neste campo, devem estar descritos os principais resultados obtidos na execução da ação MiniDMAIC, respondendo alguns questionamentos, tais como:
 - As metas foram atingidas?
 - Qual o custo benefício obtido?
- **Comment:** este campo pode ser utilizado caso haja necessidade de algum comentário adicional.
- **Viewable by:** neste campo são definidos os grupos que possuem permissão para visualizar os comentários. Como padrão, todos os participantes do projeto possuem acesso.

APÊNDICE B – TEMPLATE ANÁLISE DE CAUSAS

Este apêndice contém o *template* para classificação dos defeitos, análise e priorização de causas que auxilia na execução da fase “Medir” e “Analisar” da abordagem proposta neste trabalho. O *template* original é em formato Excel.

Consolidação e Classificação dos Defeitos e Identificação das Causas

Projeto: <Nome do Projeto>

Responsável: <Nome do Responsável>

Chave	Resumo	Criticidade	Tipo do Defeito	Origem do Defeito	Causas Relacionadas
<ID>	<Descrição do resumo do defeito ou problema>	Trivial	Interface	Requisitos	<Descrição da causa do defeito ou problema>
<ID>	<Descrição do resumo do defeito ou problema>	Major	Função (funcionalidade)	A&P - Arquitetura	<Descrição da causa do defeito ou problema>
<ID>	<Descrição do resumo do defeito ou problema>	Minor	Algoritmo (lógica interna)	A&P - Arquitetura	<Descrição da causa do defeito ou problema>
<ID>	<Descrição do resumo do defeito ou problema>	Critical	Ambiente	Configuração	<Descrição da causa do defeito ou problema>
<ID>	<Descrição do resumo do defeito ou problema>	Blocker	Atribuição	Implementação	<Descrição da causa do defeito ou problema>
<ID>	<Descrição do resumo do defeito ou problema>	Blocker	Documentação	Requisitos	<Descrição da causa do defeito ou problema>
<ID>	<Descrição do resumo do defeito ou problema>	Critical	Função (funcionalidade)	Implementação	<Descrição da causa do defeito ou problema>
<ID>	<Descrição do resumo do defeito ou problema>	Critical	Algoritmo (lógica interna)	Implementação	<Descrição da causa do defeito ou problema>

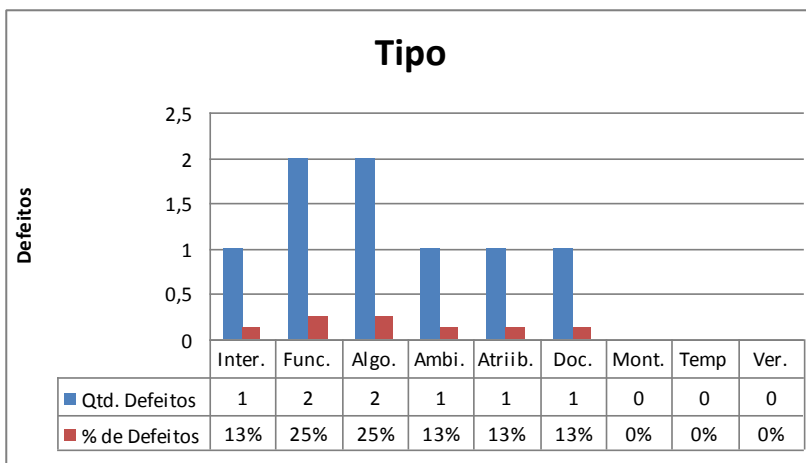
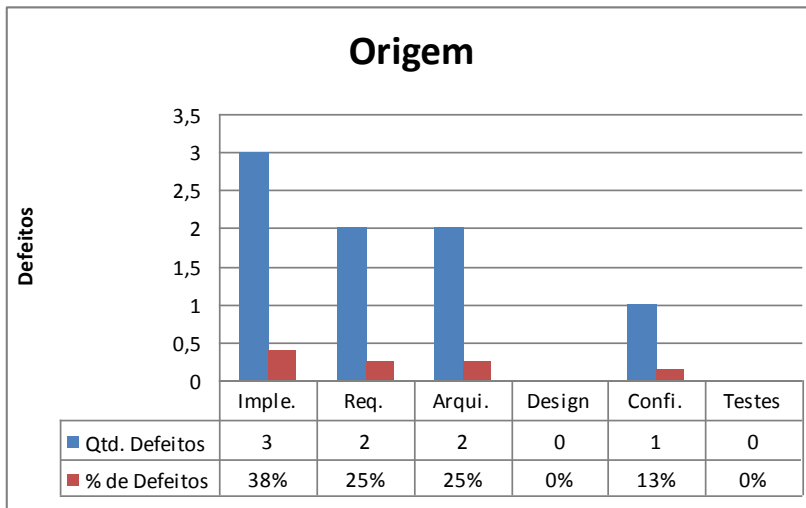
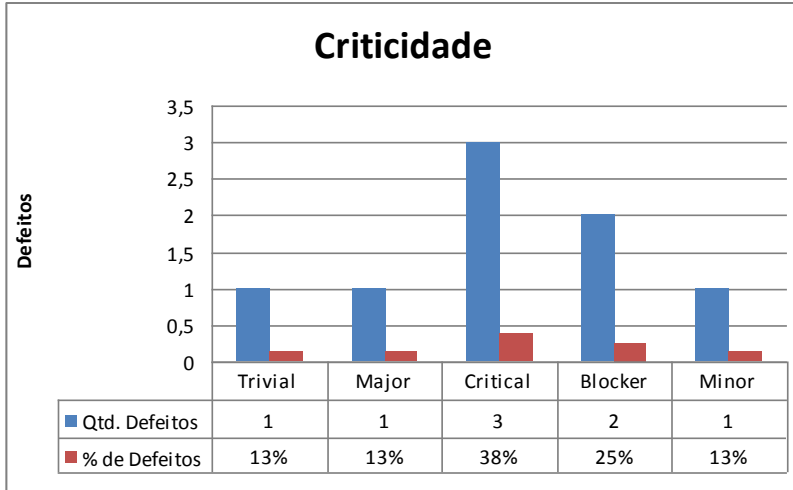
Priorização dos Defeitos por Criticidade, Origem e Tipo de Defeito

Criticidade	Qtd. Defeitos	% de Defeitos
Trivial	1	13%
Major	1	13%
Critical	3	38%
Blocker	2	25%
Minor	1	13%
Total	8	100%

Tipo do Defeito	Tipo do Defeito (Resumido)	Qtd. Defeitos	% de Defeitos
Interface	Inter.	1	13%
Função (funcionalidade)	Func.	2	25%
Algoritmo (lógica interna)	Algo.	2	25%
Ambiente	Ambi.	1	13%
Atribuição	Atrib.	1	13%
Documentação	Doc.	1	13%
Montagem/empacotamento/integração	Mont.	0	0%
Tempo/serialização/performance	Temp	0	0%
Verificação (validação de campos)	Ver.	0	0%
Total		8	100%

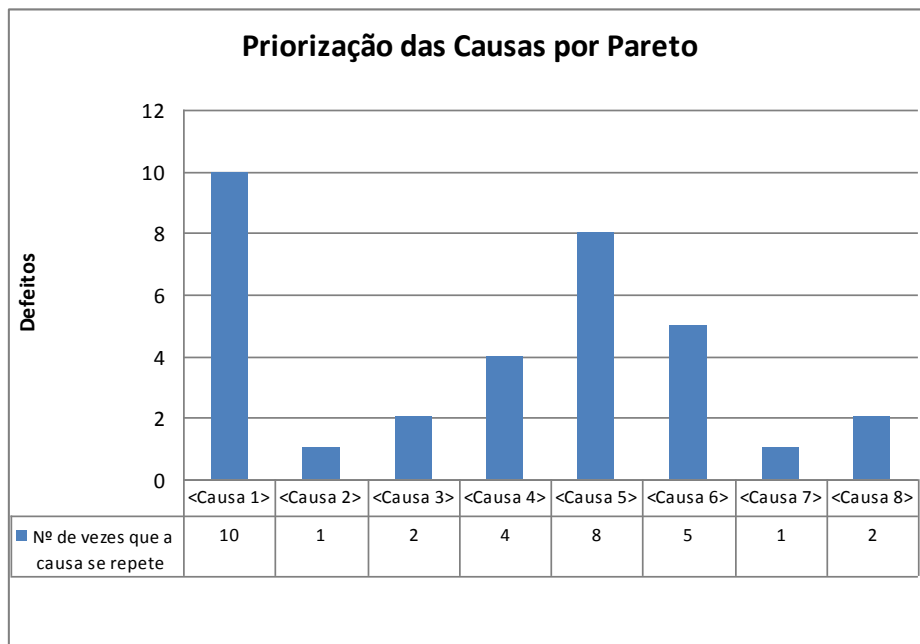
Origem do Defeito	Origem do Defeito (Resumido)	Qtd. Defeitos	% de Defeitos
Implementação	Imple.	3	38%
Requisitos	Req.	2	25%
A&P - Arquitetura	Arqui.	2	25%
A&P - Design	Design	0	0%
Configuração	Confi.	1	13%
Testes	Testes	0	0%
Total		8	100%

Gráfico de Pareto para Priorização dos Defeitos por Criticidade, Origem e Tipo de Defeito



Priorização das Causas

ID da Causa	Nº de vezes que a causa se repete	% de Defeitos
<Causa 1>	10	30%
<Causa 2>	1	3%
<Causa 3>	2	6%
<Causa 4>	4	12%
<Causa 5>	8	24%
<Causa 6>	5	15%
<Causa 7>	1	3%
<Causa 8>	2	6%
Total	33	100%



Identificação das Ações

ID da Causa Priorizada	ID das Ações	Descrição das Ações
<Causa 1>	<Ação 1>	<Descrição de cada ação>
<Causa 4>	<Ação 2>	<Descrição de cada ação>
<Causa 5>	<Ação 3>	<Descrição de cada ação>
<Causa 6>	<Ação 4>	<Descrição de cada ação>

Priorização das Ações

Tabela - Análise e Tomada de Decisão							
Alternativas	Critérios/Peso						Resultado
	Complexidade	Relacionado com a causa	Impacto	<Critério>	<Critério>	<Critério>	
	10	9	9	10	0	0	
<Ação 1>	0	0	0	0	0	0	0.0
<Ação 2>	0	0	0	0	0	0	0.0
<Ação 3>	0	0	0	0	0	0	0.0
<Ação 4>	0	0	0	0	0	0	0.0
<Alternativa>	0	0	0	0	0	0	0.0
<Alternativa>	0	0	0	0	0	0	0.0

Tabela - Critérios			
Critérios	Justificativa	Peso	Justificativa para o peso dos critérios
Complexidade	A ação é de fácil Implantação.	10	
Relacionado com a causa	Está realmente relacionada com as causas priorizadas	9	
Impacto	A ação realmente terá um grande impacto atacando as causas priorizadas do problema	9	
<Critério>			
<Critério>			
<Critério>			

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)