

**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**INFLUÊNCIA DA UTILIZAÇÃO DE PROCESSO  
UNIFICADO, TESTES E MÉTRICAS NA QUALIDADE DE  
PRODUTOS DE SOFTWARE**

**VERA LÚCIA XAVIER DE SALES CALÇADO**

**ORIENTADORA: Dra. JULIANA FERNANDES CAMAPUM**

**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA**

**PUBLICAÇÃO: PPGENE.DM – 311 A/2007**

**BRASÍLIA/DF: SETEMBRO 2007**

**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENERGIA ELÉTRICA**

**INFLUÊNCIA DA UTILIZAÇÃO DE PROCESSO  
UNIFICADO, TESTES E MÉTRICAS NA QUALIDADE DE  
PRODUTO DE SOFTWARE**

**VERA LÚCIA XAVIER DE SALES CALÇADO**

**DISSERTAÇÃO DE MESTRADO ACADÊMICO SUBMETIDA AO  
DEPARTAMENTO DE ELÉTRICA DA FACULDADE DE TECNOLOGIA DA  
UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS  
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE.**

**APROVADA POR:**

---

**JULIANA FERNANDES CAMAPUM, PhD (ENE/UnB)  
(ORIENTADORA)**

---

**ADSON FERREIRA DA ROCHA, DOUTOR (ENE/UnB)  
(EXAMINADOR INTERNO)**

---

**KÁTHIA MARÇAL DE OLIVEIRA, DOUTORA (UCB)  
(EXAMINADOR EXTERNO)**

**BRASÍLIA/DF, 13 DE SETEMBRO DE 2007.**

## FICHA CATALOGRÁFICA

CALÇADO, VERA LÚCIA XAVIER DE SALES

Influência da Utilização de Processo Unificado, Testes e Métricas na Qualidade de Produtos de Software. [Distrito Federal] 2007.

xiii, 138 p., 210x297mm (EnE/FT/UnB, Mestre, Engenharia Elétrica, 2007).

Dissertação de Mestrado – Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica.

- |                          |                    |
|--------------------------|--------------------|
| 1. Qualidade de Software | 5. Caso de Uso     |
| 2. CMM/CMMI              | 6. RUP             |
| 3. ISO IEC 9126          | 7. Caso de Teste   |
| 4. Métricas              | 8. GQM             |
| I. EnE/FT/UnB            | II. Título (série) |

## REFERÊNCIA BIBLIOGRÁFICA

CALÇADO, V. L. X. S. (2007). Influência da Utilização de Processo Unificado, Testes e Métricas na Qualidade de Produtos de Software. Dissertação de Mestrado, publicação ENE.DM – 311 A/2007. Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 138 p.

## CESSÃO DE DIREITOS

NOME DO AUTOR: Vera Lúcia Xavier de Sales Calçado

TÍTULO DA DISSERTAÇÃO: Influência da Utilização de Processo Unificado, Testes e Métricas na Qualidade de Produtos de Software.

GRAU: Mestre

ANO: 2007

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente com propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta monografia de pós-graduação pode ser reproduzida sem a autorização por escrito do autor.

---

Vera Lúcia Xavier de Sales Calçado  
Quadra 201, Bloco C, Apto 203, Residencial Imprensa I  
CEP 71937 540, Brasília-DF

## **DEDICATÓRIA**

A Deus, pela vida, pela saúde em todos os sentidos, pela proteção, pela sabedoria, pela luz, pelo alimento do meu corpo, de minha mente e de minha alma, pelos ensinamentos, por minha família, pelos meus amigos, pelos meus dons, por minha vontade, por minha força, por eu estar aqui e por estar sempre ao meu lado... Não há como expressar em palavras tudo que devo... Sem ti, nossa vida seria nada, em ti, podemos tudo.

Ao meu grande amor, Ildomar, meu esposo, pela compreensão da minha ausência, devido a meu estudo. E pela sua dedicação em organizar tempo para me ajudar durante esse período, abrindo mão de seus finais de semana e feriados, contribuindo para que eu vencesse este e vários outros obstáculos. Sem a sua ajuda, não teria conseguido chegar onde estou.

Ao meu filhinho, Ian, que me inspira, me ajuda e me acompanhou em todo o tempo que dediquei a este trabalho. E, mesmo agora, ajuda-me a ver coisas que não tinha percebido, e está aqui comigo agora enquanto amamento.

À minha mãe, Oswalda, que sempre esteve ao meu lado com palavras de apoio e incentivo, “brincas” e conselhos. E por ter sido também meu pai desde minha infância...

Em memória do meu pai, Constâncio, que, com certeza, se estivesse aqui estaria me incentivando e apoiando. O senhor ainda está em meu pensamento em tudo que faço.

## AGRADECIMENTOS

A minha família: Cláudia e Manuel (Leo); Cláudio Roberto, Dênia e Ítalo, pelo apoio nesse processo.

Aos meus sogros, Ildo e Maria Abadia, que sempre me apoiaram no estudo e torceram pelo meu sucesso.

Em especial, à minha orientadora, Dr<sup>a</sup> Juliana Camapum, que acreditou no meu potencial e me apoiou no desenvolvimento do meu trabalho. Este não teria sido possível sem a sua orientação, seu apoio, sua amizade.

Para “Dinha” e “Zé” (*in memoriam*), por todo apoio, diálogo e carinho que recebi.

A minha amiga Liana, que me ajudou em toda esta etapa e por todos os conselhos e, principalmente, pela troca de idéias desde que nos conhecemos.

Ao Professor Dr Adson, que me estendeu a mão quando precisei.

Aos amigos Gilmar Correa, Maria Nirce, Bruno César, Marta Maria, Luhana, Letícia, José Antônio, Regina e Lindomar Júnior. Tem sido um privilégio a amizade de vocês.

Aos demais professores do Departamento de Engenharia Elétrica, pelo ensino que me foi dado.

Aos funcionários da Secretaria de Pós-Graduação do Departamento de Engenharia Elétrica, pela ajuda e atenção no que foi preciso.

## **RESUMO**

### **INFLUÊNCIA DA UTILIZAÇÃO DE PROCESSO UNIFICADO, TESTES E MÉTRICAS NA QUALIDADE DE PRODUTOS DE SOFTWARE**

**Autora: Vera Lúcia Xavier de Sales Calçado**

**Orientadora: Prof<sup>a</sup>. Dr<sup>a</sup>. Juliana Fernandes Camapum**

**Programa de Pós-graduação em Engenharia Elétrica**

**Brasília, mês de setembro (2007)**

No processo de produção de software espera-se que o seu desenvolvimento conduza a um produto confiável. Para conseguir melhores níveis de qualidade, as empresas buscam a certificação de seus processos e modelos.

Neste trabalho, buscou-se propor uma abordagem que utilize modelos de maturidade e processo unificado. Ademais, foi dedicada uma parcela significativa de tempo e esforço para a criação, emprego e gerenciamento dos testes nas diversas fases do ciclo de vida do software.

Finalmente, como é comum em todo processo de produção, foi necessário avaliar quesitos como: tempo entre especificação e produção, custo da produção e qualidade atingida.

Com o objetivo de avaliar a qualidade do produto, bem como os outros quesitos mencionados, realizou-se um estudo considerando o impacto da adoção de testes no processo de produção de software, tanto na fase de concepção do produto, como na fase de produção, em comparação com um processo no qual a aplicação de teste não segue as especificações de um processo unificado.

Essa avaliação foi feita mediante a aplicação de um conjunto de métricas criadas com o objetivo de avaliar a qualidade do produto na fase de produção, bem como avaliar a qualidade da própria manutenção do produto nessa fase.

Palavras-chave: Qualidade de Software, CMM, CMMi, Processo Unificado, RUP, ISO/IEC 9126-3, GQM, Testes, Processo de Software, Caso de Uso (CDU), Caso de Teste (CDT)

## **ABSTRACT**

### **THE INFLUENCE OF THE USE OF UNIFIED PROCESS, TESTS AND METRICS IN THE QUALITY OF SOFTWARE PRODUCTS**

**Author: Vera Lúcia Xavier de Sales Calçado**

**Supervisor: Prof<sup>ª</sup>. Dr<sup>ª</sup>. Juliana Fernandes Camapum**

**Programa de Pós-graduação em Engenharia Elétrica**

**Brasília, month of September (2007)**

During the software production, it is expected that the development can guide the process towards a reliable product. The companies that are envisaging good levels of quality are reaching such levels by the certification or their process.

In this work, it is proposed a new approach that deploys maturity models together with a unified process. Beside, this work spent much time and effort in the creation, deployment and management of tests in the several phases of the software life cycle.

Finally, as it is common in most process of production, it was necessary to evaluate some requirements like: spent time between specification and production, costs of production and the effective level of quality.

With the aim to evaluate the quality of the product, as well as some other issues, a study had to be carried out. This study took into consideration the impact of the adoption of tests in the process of production, not only during the implementation but also in the phase of production. Such results were compared with a product that has been made without the deployment of a unified process.

This evaluation was done by the application of a set of metrics created in order to measure the quality of the product during the production phase and also to evaluate the quality of maintenance of the product itself.

**Keywords: Quality of Software, CMM, CMMi, Unified Process, RUP, ISO/IEC 9126-3, GQM, Tests, Process of Software, Use Case (CDU), Test Case (CDT)**

# SUMÁRIO

<b>1 – INTRODUÇÃO.....</b>	<b>1</b>
<b>1.1 – Qualidade de Software no Brasil .....</b>	<b>1</b>
<b>1.2 – Motivação.....</b>	<b>6</b>
<b>1.3 – Objetivo.....</b>	<b>6</b>
<b>1.4 – Estrutura da Dissertação .....</b>	<b>6</b>
<b>2 – QUALIDADE DE SOFTWARE .....</b>	<b>9</b>
<b>2.1 – Aferição da Qualidade .....</b>	<b>9</b>
2.1.1 – Medição objetiva da qualidade.....	10
2.1.2 – Medição Subjetiva da Qualidade .....	10
2.1.3 – Comparação entre as formas de medição .....	10
<b>2.2 – Processo de Software.....</b>	<b>11</b>
2.2.1 – Processo Unificado .....	11
2.2.2 – Rational Unified Process (RUP) .....	12
<b>2.3 – Modelos de Gestão da Qualidade no Processo de software .....</b>	<b>14</b>
2.3.1 – Modelo de Capacidade e Maturidade (CMM) .....	14
2.3.2 – CMM Integrado (CMMi).....	16
<b>2.4 – Ciclo de Vida de Software .....</b>	<b>17</b>
<b>2.5 – Gerência de Qualidade de Software .....</b>	<b>18</b>
<b>2.6 – Testes .....</b>	<b>20</b>
<b>2.7 – Métricas.....</b>	<b>21</b>
<b>2.8 – Relação entre Gestão da Qualidade, Testes e Métricas .....</b>	<b>22</b>
<b>2.9 – Avaliação dos processos.....</b>	<b>25</b>
<b>3 – TESTES .....</b>	<b>26</b>
<b>3.1 – Desenvolvimento do Produto.....</b>	<b>26</b>
3.1.1 – Papéis .....	32
3.1.2 – Tipos de testes.....	32
3.1.3 – Abordagens de Teste.....	34
3.1.4 – Classificação quanto à forma de execução.....	36
<b>3.2 – Essenciabilidade dos Testes .....</b>	<b>36</b>
<b>3.3 – Processo de escolha do tipo de Teste.....</b>	<b>37</b>
<b>3.4 – PROPOSTA PARA Aplicação de Teste .....</b>	<b>38</b>
<b>3.5 – Solicitação de Manutenção .....</b>	<b>40</b>
<b>3.6 – Objetivo do Teste.....</b>	<b>42</b>
<b>4 – MÉTRICAS DE QUALIDADE.....</b>	<b>43</b>
<b>4.1 – Abordagem GQM.....</b>	<b>43</b>
<b>4.2 – Norma ISO /IEC 9126.....</b>	<b>45</b>
<b>4.3 – Manutenibilidade .....</b>	<b>46</b>
<b>4.4 – Métricas.....</b>	<b>48</b>
<b>4.5 – Consolidação.....</b>	<b>52</b>



<b>5 – APLICAÇÃO E RESULTADOS.....</b>	<b>53</b>
<b>5.1 – Descrição do Produto MGP .....</b>	<b>53</b>
<b>5.2 – Descrição sucinta do processo .....</b>	<b>55</b>
<b>5.3 – Descrição do Produto SGQ .....</b>	<b>58</b>
<b>5.4 – Comparação entre os Produtos MGP e SGQ .....</b>	<b>60</b>
5.4.1 – Fase de Desenvolvimento onde foram realizados os testes.....	62
5.4.2 – Análise Comparativa.....	67
5.4.3 – Fase de Produção - Coleta de Dados.....	68
5.4.4 – Análise Comparativa dos Resultados.....	78
<b>6 – CONCLUSÕES E TRABALHOS FUTUROS .....</b>	<b>85</b>
<b>7 – REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>89</b>
<b>A – CASO DE USO DO MGP .....</b>	<b>94</b>
<b>B – CASO DE USO DO SGQ .....</b>	<b>101</b>
<b>C – CASO DE TESTE DO MGP.....</b>	<b>104</b>
<b>D – MANUTENÇÃO CORRETIVA MGP.....</b>	<b>121</b>
<b>E – MANUTENÇÃO EVOLUTIVA MGP.....</b>	<b>123</b>
<b>F – MANUTENÇÃO CORRETIVA DO SGQ.....</b>	<b>124</b>
<b>G – ARTIGOS PUBLICADOS.....</b>	<b>125</b>

## LISTA DE TABELAS

Tabela 2.1 - Níveis do CMMI .....	16
Tabela 2.2 - Gerência de Qualidade de Software .....	19
Tabela 3.1 - Informações de um CDT .....	28
Tabela 3.2 - Solicitação de Manutenção.....	41
Tabela 4.1 - Descrição do Objetivo .....	46
Tabela 4.2 – Resumo do sentido das métricas.....	52
Tabela 5.1 – Tabela comparativa MGP x SGQ.....	60
Tabela 5.2 – Erros encontrados no Desenvolvimento do MGP .....	64
Tabela 5.3 – Solicitações de manutenção do MGP .....	69
Tabela 5.4 – Solicitação de manutenções do SGQ.....	69
Tabela 5.5 – Dados para cálculo de M1 no SGQ .....	72
Tabela 5.6 – Dados para cálculo de M1 no MGP.....	72
Tabela 5.7 – Dados para cálculo de M2 no SGQ .....	73
Tabela 5.8 – Dados para cálculo de M2 no MGP.....	73
Tabela 5.9 – Dados para cálculo de M3 .....	74
Tabela 5.10 – Dados para cálculo de M4 para o MGP.....	75
Tabela 5.11 – Dados para cálculo de M4 para o SGQ .....	76
Tabela 5.12 – Dados para cálculo de M5 para o MGP.....	76
Tabela 5.13 – Dados para cálculo de M5 para o SGQ .....	77
Tabela 5.14 – Dimensionamento das equipes .....	78
Tabela 5.15 – Comparação para M1.....	80
Tabela 5.16 – Comparação para M2.....	82
Tabela 5.17 – Comparação para M3.....	82
Tabela 5.18 – Comparação para M4.....	83
Tabela 5.19 – Comparação para M5.....	84

## LISTA DE FIGURAS

Figura 1.1- Evolução das certificações de CMM no Brasil.....	4
Figura 1.2- Evolução das certificações de CMMI no Brasil .....	5
Figura 2.1 - Descrição do RUP.....	14
Figura 2.2 – Importância de Processos, Testes e Métricas para a Qualidade de Software..	22
Figura 2.3 - Relação entre Gestão da Qualidade, Testes e Métricas .....	24
Figura 2.4 - Avaliação dos processos. ....	25
Figura 3.1 - Desenvolvimento do Produto. ....	27
Figura 3.2 - Disciplina de Teste.....	28
Figura 3.3 - Atividade de Especificação de Testes nas fases do Processo. ....	30
Figura 3.4 - Diagrama de Atividade de Execução de Teste. ....	31
Figura 3.5 - Estratégias dos testes caixa branca. ....	35
Figura 3.6 – Estratégias dos testes caixa preta. ....	35
Figura 3.7 - Processo de Testes. ....	39
Figura 5.1 – Papéis das equipes do MGP. ....	54
Figura 5.2 – Papéis das equipes do Produto SGQ.....	59
Figura 5.3 – Tempo gasto para Testes no MGP. ....	61
Figura 5.4 – Criticidade.....	66
Figura 5.5 – Origem dos erros.....	66
Figura 5.6 – Evolução mensal da métrica M1.....	73
Figura 5.7 – Evolução mensal da métrica M2.....	74
Figura 5.8 – Resultado da métrica M3 para o MGP e o SGQ.....	75
Figura 5.9 – Evolução mensal da métrica M4.....	76
Figura 5.10 – Evolução mensal da métrica M5.....	77
Figura 5.11 – Análise conjunta das métricas M1 e M4.....	79
Figura 5.12 – Análise conjunta das métricas M2 e M5.....	81

## LISTA DE SÍMBOLOS, NOMENCLATURAS E ABREVIACÕES

ABNT	Associação Brasileira de Normas Técnicas
AIE	Arquivos de Interface Externa
ALI	Arquivos Lógicos Internos
APF	Análise de Ponto de Função
CDU	Caso de Uso
CE	Consulta Externa
CMM	Capability Maturity Model for Software
CMMI	Capability Maturity Model Integrated
CMU	Carnegie Mellon University
CDT	Caso de Teste
EE	Entrada Externa
GQM	Goal Question Metric
IBM	International Business Machines
IBQP	Instituto Brasileiro de Qualidade e Produtividade
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
	Instituto Nacional de Metrologia, Normalização e Qualidade
INMETRO	Industrial
ISO	International Organization for Standardization
JAVA	Linguagem de Programação
KPA	Key Process Areas
KSLOC	Kilo Source Line of Code – Mil Linhas de Código
MGP	Módulo de Gestão de Planejamento
PA's	Process Areas
PBQP	Programa Brasileiro de Qualidade e Produtividade
PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
RUP	Rational Unified Process
SE	Saída Externa
SEI	Software Engineering Institute
SGBD	Sistema Gerenciador de Banco de Dados
SGQ	Sistema de Gestão de Qualidade
SPICE	Software Process Improvement and Capability dEtermination
TI	Tecnologia da Informação
UML	Unified Modeling Language

## GLOSSÁRIO

**Artefatos** – Podem ser um modelo, um elemento de modelo ou um documento.

**Análise de Pontos por Função** – Técnica de avaliação de um sistema, conhecida como FPA – *Function Point Analysis*, baseada na medição do valor das funções executadas pelos programas, ao invés de utilizar como base o volume ou a complexidade do código dos programas.

**Caso de Teste** – É a definição (geralmente formal) de um conjunto específico de entrada de teste, condições de execução e resultados esperados.

**Caso de Uso** – Define um conjunto de instâncias de CDU, no qual cada instância é uma seqüência de ações realizada por um sistema que produz um resultado de valor observável para determinado ator.

**Disciplina** – Cada fase pode comportar várias iterações, e cada iteração, por sua vez, está organizada em disciplinas (*workflows*), que descrevem o que deve ser feito em termos de atividades, responsabilidades e artefatos.

**Iterações** – São pequenas etapas do ciclo de vida, definidas por seqüências distintas de atividades com planejamento detalhado e critério de validação, resultando sempre em algum produto, interno ou externo.

**Modelo de Teste** – Descreve como os componentes executáveis do modelo de implementação são testados.

**Papéis** – Definem o comportamento e responsabilidades dos indivíduos ou conjunto de indivíduos trabalhando em equipe durante o desenvolvimento de software.

**RUP** – É um processo de engenharia de software que aborda todo o ciclo de vida de um projeto e guia as equipes de desenvolvimento no gerenciamento e nas atividades práticas de engenharia de software.

**UML** – É uma linguagem gráfica para visualização, especialização, construção e documentação dos artefatos de um sistema fundamentado em software.

# 1 – INTRODUÇÃO

## 1.1 – QUALIDADE DE SOFTWARE NO BRASIL

Desde há muito tempo, o Brasil tem adotado uma política para aumentar a qualidade e a produtividade da indústria brasileira.

A criação do Programa Brasileiro de Qualidade e Produtividade – PBQP, na década de 1990, e que teve como grande encorajador o então Presidente da República Fernando Collor de Melo, apresentava como ponto focal a necessidade de ampliar e desenvolver a produção automobilística. Desse modo, seria possível dar um salto das chamadas “carroças” para os modelos de automóveis que circulavam pelos países desenvolvidos. Contudo, cabe ressaltar que o programa não visava somente à melhoria da produção de carros, mas, em vez disso, propunha-se a alavancar todo o processo de produção do Brasil.

A qualidade perseguida pela indústria de Tecnologia da Informação (TI) é sustentada e garantida por diversos critérios e normas, conforme será descrito a seguir. Entretanto, considerando apenas o caráter preliminar, as dimensões da qualidade passam por critérios como:

- **Efetividade** – que diz respeito ao fato de que o sistema atinge os objetivos para os quais foi especificado a um baixo custo.
- **Pessoal** – que reflete a comparação entre a qualidade do produto e a expectativa previamente estabelecida.
- **Custo** – diz respeito aos recursos dispendidos para atingir um objetivo.
- **Rastreabilidade** – para que seja possível traçar o histórico das ações que envolveram cada parte do projeto.
- **Manutenibilidade** – que proporciona o grau de confiança para que possíveis falhas ou inconsistências possam ser corrigidas após a entrega de um produto etc.

Nesse sentido, com vistas a assegurar níveis de qualidade adequados às particularidades de cada projeto ou mercado, os profissionais da área de qualidade desenvolvem métodos e

ferramentas para a aplicação de testes que visam a aferir a profundidade dos critérios que compõem a qualidade.

Como resultado, a postura Qualidade passou a fazer parte do cotidiano de vários setores, quais sejam: energia, telecomunicações, têxtil, entre outros. Houve, em decorrência dessa nova atividade, a demanda por profissionais que conseguissem implementar, garantir e certificar uma política de qualidade. Foram criados cursos para ensinar como criar produtos de melhor qualidade, de acordo com a política de qualidade da certificação ISO 9000.

Nesse sentido, o setor de software, que andava ao largo de toda essa atividade, demorou algum tempo para incorporar seu modo de produção às novas exigências do mercado de tecnologia de informação. A exemplo de outros países, o governo federal lançou o Programa Brasileiro de Qualidade e Produtividade (PBQP) para a indústria de software.

A primeira fase do PBQP [1], que aconteceu entre os anos de 1990 e 1995, tinha o objetivo de estimular, articular, orientar e apoiar os esforços da sociedade brasileira na busca de competitividade internacional, por meio da promoção de ações de melhoria da qualidade e aumento da produtividade dos bens e serviços produzidos e oferecidos no país. Esse plano resultou da junção de projetos e comissões, com orientação unificada e execução descentralizada pelos agentes envolvidos.

O período compreendido entre os anos de 1996 e 1998 foi marcado pela concepção dos projetos estratégicos. O aumento de fatores relacionados à competitividade do setor de software no âmbito internacional ensejou o aperfeiçoamento do PBQP [2]. A necessidade de aumentar a eficiência do aparato estatal, aliado com o fato de ser necessário o fortalecimento das microempresas, também contribuiu para que o PBQP sofresse alterações.

As alterações redundaram na mudança estratégica do PBQP. Essa fase focalizou suas atividades em 4 (quatro) macroprioridades, quais sejam:

- Qualidade de Vida.
- Qualidade e Emprego
- Qualidade no Setor Produtivo

- Qualidade e Participação na Administração Pública

Entre os anos de 1998 e 2001, o governo adotou medidas com vistas a incorporar no programa outras parcelas da sociedade. Para tanto, realizou um realinhamento nas estratégias do PBQP [3]. Esse realinhamento estratégico foi consubstanciado com:

- o estabelecimento de metas mobilizadoras nacionais;
- a criação de expressões mobilizantes;
- a implantação de novo modelo de gerenciamento integrado;
- a coordenação executiva conjunta por órgão governamental (INMETRO) e entidade da sociedade civil (IBQP).

Os ajustes foram efetuados em 1999, e consistiam em ações tais como: ratificação da organização por metas, criação de subprogramas estruturantes, incorporação de novos temas, alterações na composição do Comitê Nacional da Qualidade e Produtividade e na Coordenação Executiva do PBQP e acompanhamento e avaliação do PBQP.

O PBQP e a mudança das regras jurídicas desenhadas pelo governo visavam a proporcionar um panorama favorável ao desenvolvimento do setor de informática no Brasil. Essa mudança de postura acabou proporcionando o surgimento de setores especializados na gestão de uma política de qualidade na produção de softwares.

O principal objetivo de uma política de qualidade é o de criar, construir e entregar produtos com qualidade. Em termos gerais, a adoção de uma política eficiente ataca o problema de se produzir melhor sob diversas óticas. Dentre as formas de se alcançar melhores níveis de qualidade estão ações como: investimentos em capacitação de seus colaboradores, obtenção de insumos, verificação de preenchimento dos requisitos, entre outras. Essa última abrange uma vasta aplicação de atividades a projetos; dentre esses, encontra-se a aplicação de testes.

A qualidade, num conceito amplo, refere-se à estratégia de gestão em que se procura otimizar a produção e reduzir os custos. Mais especificamente, a qualidade de software foca seu objetivo em produzir softwares de modo a prover produtos que atendam a requisitos dos clientes, tais como segurança e desempenho, e dos desenvolvedores de software, tais como custo no desenvolvimento e dinâmica do trabalho.



Até tempos bem recentes, a aplicação de testes consistia apenas em uma mera verificação do nível de atendimento de um requisito em particular. O encarregado do teste aplicava um procedimento que julgava conveniente. O resultado podia ou não ser satisfatório. Se fosse insatisfatório, o problema era levantado para ser corrigido. Se não, o quesito era considerado atendido.

Como resultado da própria evolução do setor de informática, impulsionador dessa mesma evolução, o quesito qualidade passou a ser visto como um fator de diferencial competitivo.

A reboque das necessidades do setor de tecnologia, como ocorre com outros setores de produção, surgiram propostas de normas cujo objetivo principal era o de prover direcionamento para a implementação de uma política de qualidade. Como exemplo, podem-se citar as normas SPICE, CMM e CMMI.

Outra resposta dada à demanda por padrões de qualidade foi o projeto SPICE (*Software Process Improvement and Capability Determination*), criado em 1993 por uma equipe de especialistas internacionais para apoiar o desenvolvimento das versões iniciais desta Norma e coordenar a utilização destas versões pela comunidade. O projeto SPICE foi aprovado como Relatório Técnico em 1998 e, em outubro de 2003, publicada a Norma Internacional ISO/IEC 15504.

No Brasil, o aumento na demanda por certificações pode ser percebido pelo aumento de certificações, conforme Figura 1.1e Figura 1.2.

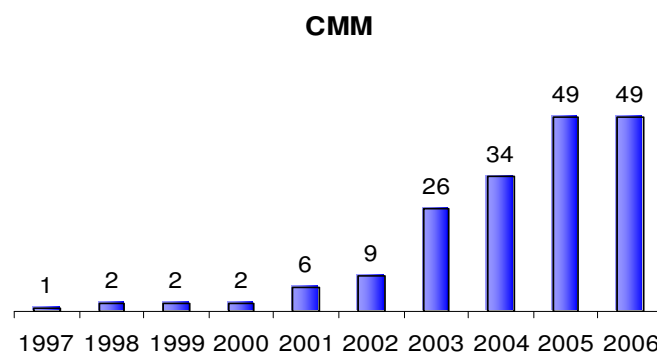


Figura 1.1- Evolução das certificações de CMM no Brasil<sup>1</sup>

<sup>1</sup> Fontes: ISD Brasil, Procesix, empresas qualificadas e imprensa especializada, compilado por MCT/SEPIN/DIA. Situação em agosto/2006

## CMMI

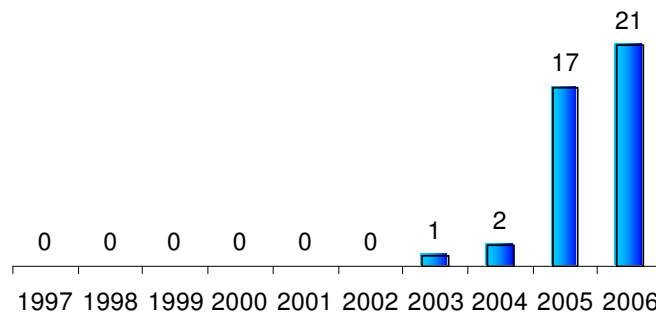


Figura 1.2- Evolução das certificações de CMMI no Brasil<sup>2</sup>

O fornecimento de um software comercial permeia várias expectativas com diversos níveis de exigência. Por essa razão, o emprego de testes atingiu a importância que hoje pode ser vista nas companhias desenvolvedoras de sistemas de informação.

Nesse sentido, a adoção de procedimentos de testes nas diversas fases de produção de um software fornece subsídios importantes para os usuários e clientes, na medida em que os requisitos estabelecidos na concepção do projeto sejam atendidos, e para as empresas desenvolvedoras. Porquanto, a realização de testes garante que as funcionalidades originalmente contratadas foram efetivamente implementadas no projeto.

Este trabalho apresenta um procedimento de testes baseado numa abordagem manual de aplicação dos mesmos. Isto é, não serão empregadas ferramentas específicas ou especialmente desenvolvidas para a aplicação dos testes. Em vez disso, será descrito um procedimento passo a passo, que se objetiva a cobrir a maior parte das situações, além de atribuir ao profissional a cargo da elaboração e aplicação dos mesmos, a responsabilidade pelo controle e gestão da qualidade do software desde as primeiras versões produzidas até a efetiva entrega do produto aos usuários e clientes.

---

<sup>2</sup> Fontes: ISD Brasil, ProceSix, empresas qualificadas e imprensa especializada, compilado por MCT/SEPIN/DIA. Situação em agosto/2006

## **1.2 – MOTIVAÇÃO**

A gerência de projetos envolve todo o processo de desenvolvimento de software. Como motivador, e às vezes como reflexo dessa situação, a sociedade demanda cada vez mais produtos e soluções com qualidade que atendam às expectativas dos usuários e clientes. Assim, é preciso definir o escopo e o planejamento do projeto a ser desenvolvido. Isso envolve o acompanhamento do projeto, coleta de dados, avaliação das métricas, visando à entrega do produto dentro do prazo e custo esperados e com a qualidade adequada. Estas etapas são de fundamental importância para o sucesso do projeto.

Nota-se que a grande preocupação é devida ao custo decorrente da implementação de um produto de software sem qualidade, que apresenta desde custos com retrabalho no desenvolvimento de sistemas até a disponibilização de recursos para manutenções.

## **1.3 – OBJETIVO**

O objetivo deste trabalho é propor um modelo de gestão de qualidade no desenvolvimento de um produto de software com base na aplicação de testes nas fases de desenvolvimento e produção e no emprego de métricas voltadas à avaliação da manutenção do produto. O emprego de testes visa a identificar e corrigir o maior número de erros e inconsistências na fase de desenvolvimento de modo que a fase de produção se torne o mais imune possível à ocorrência de falhas. Ademais, a utilização de métricas durante a fase de evolução visa a verificar a estabilidade do sistema e possibilitar um dimensionamento e avaliação da equipe de manutenção, além de fornecer subsídios importantes para novos projetos.

## **1.4 – ESTRUTURA DA DISSERTAÇÃO**

O conteúdo desse trabalho foi dividido em 6(seis) capítulos, sobre os quais apresenta-se um breve resumo a seguir:

## **CAPÍTULO 1 – INTRODUÇÃO**

Este capítulo apresenta o panorama sobre qualidade de software no Brasil, bem como descreve os aspectos motivadores para o desenvolvimento deste trabalho e, por fim, introduz quais os objetivos que se pretende alcançar.

## **CAPÍTULO 2 – QUALIDADE DE SOFTWARE**

Esse capítulo apresenta uma abordagem geral sobre qualidade, incluindo suas características e formas de avaliação. Além disso, são apresentadas algumas considerações focadas no desenvolvimento de produtos de software.

## **CAPÍTULO 3 – TESTES**

Esse capítulo apresenta uma abordagem geral sobre testes, incluindo os tipos de testes aplicados à melhoria de processos. Essa aplicação consiste na verificação de existência de não conformidades nas fases de desenvolvimento do sistema.

## **CAPÍTULO 4 – METRICAS DE QUALIDADE**

Esse capítulo apresenta uma proposta referente a um conjunto de métricas que foram criadas com base na abordagem GQM. As métricas foram desenvolvidas visando à verificação do quesito manutenibilidade, conforme definição da Norma ISO/IEC 9126. Esse conjunto tem como principal enfoque a avaliação dos aspectos relativos à qualidade na manutenção de produtos de software quando o mesmo encontra-se na fase de produção.

## **CAPÍTULO 5 – APLICAÇÃO PRÁTICA**

Esse capítulo mostra a aplicação prática dos conceitos trabalhados nos capítulos anteriores. Esta consiste no emprego de Processo Unificado para o desenvolvimento de um produto de software. Durante o desenvolvimento foi focado, principalmente, a definição, especificação e emprego de testes, conforme estabelecido pelo RUP. Para complementar o trabalho, o sistema foi acompanhado durante a fase de produção onde foi empregado o conjunto de métricas para avaliar a qualidade da manutenção do sistema e o impacto da adoção da disciplina de testes realizados na fase de desenvolvimento. Finalizando, foi feita uma análise dos resultados alcançados, mediante a comparação entre outro produto de software produzido por processo sem a adoção clara de uma disciplina de teste.

## **CAPÍTULO 6 – CONCLUSÕES E TRABALHOS FUTUROS**

Esse capítulo destina-se à apresentação das conclusões do trabalho. Além disso, foram feitas algumas sugestões de trabalhos futuros que possam vir a ser desenvolvidos de forma a complementar o tema abordado por esta dissertação e, por conseguinte, contribuir para a Qualidade de Software.

## **2 – QUALIDADE DE SOFTWARE**

Primeiramente, cabe destacar o que significa o termo qualidade. De acordo com o Dicionário Houaiss [4], qualidade é “característica superior ou atributo distintivo positivo que faz alguém ou algo sobressair em relação a outros; virtude”.

Especificamente no que se refere à indústria de software, sob o ponto de vista da Norma ISO 8402:2000 [5], pode-se definir sistema da qualidade como um conjunto composto pela estrutura organizacional, pelos procedimentos, pelos processos e pelos recursos necessários para implementar a gestão da qualidade numa organização. Por sua vez, a gestão da qualidade pode ser descrita como o conjunto de atividades que determina a política da qualidade, os objetivos e as responsabilidades, implementando-os por meios como o planejamento, o controle, a garantia e a melhoria da qualidade [6].

De forma simplificada, pode-se dizer que um sistema da qualidade baseado na série de normas ISO 9000 [7] é um conjunto de recursos e regras estabelecidas e implementadas de forma adequada, com o objetivo de orientar cada parte da organização para que execute um conjunto de atividades de forma harmônica, de maneira correta e no tempo apropriado, com o objetivo de tornar a organização competitiva [8].

No contexto do desenvolvimento de software, qualidade pode ser entendida como um conjunto de características a serem satisfeitas em um determinado grau, de modo que o produto de software atenda às necessidades explícitas e implícitas de seus usuários [9].

### **2.1 – AFERIÇÃO DA QUALIDADE**

A qualidade, portanto, é um objetivo a ser perseguido por pessoas e organizações, sejam essas públicas ou privadas, voltadas a produtos ou serviços. Entretanto, é necessário verificar-se como a qualidade pode ser medida.

A partir da necessidade de se estabelecer critérios para classificar processos e produtos, criaram-se medidas de qualidade que podem ser classificadas em objetivas e subjetivas.

### **2.1.1 – Medição objetiva da qualidade**

A qualidade objetiva refere-se à medida ou classificação de algum processo ou resultado em função da coleta de dados. A princípio, esta classificação deve-se ao fato de que em uma mesma situação o resultado é obtido independentemente do meio, ferramenta ou pessoal que coletou os dados. Ou seja, a partir de dados coletados, a classificação de qualidade vai ser uniforme [10].

Especificamente no que se refere a software, as métricas objetivas são indicadores calculados a partir de dados e eventos ocorridos no sistema que permitem a obtenção de valores que designarão o nível de qualidade.

Importa ressaltar que no processo de gestão de qualidade de um software, deve-se ter em mente a relação entre custo e benefício em se utilizar uma ou outra métrica, bem como no estabelecimento da frequência de coleta da mesma [10].

### **2.1.2 – Medição Subjetiva da Qualidade**

A qualidade subjetiva, como o próprio nome diz, refere-se tão-somente à designação ou classificação da qualidade a partir do ponto de vista de um sujeito. Essa forma de se aferir a qualidade é importante quando não se tem um delineamento claro para se estabelecer parâmetros objetivos para medir a qualidade. Nesses casos, lança-se mão do caráter de subjetividade ou juízo de avaliação.

### **2.1.3 – Comparação entre as formas de medição**

A associação dos resultados obtida a partir das análises objetiva e subjetiva proporciona informações importantes acerca da efetiva qualidade de um dado produto. A relação a seguir ilustra algumas das características das medições objetivas e subjetivas.

#### **Objetiva:**

- Pode ser utilizada de forma mais freqüente.
- Independe daquele que realiza a obtenção dos dados.
- Facilita a comparação entre produtos similares.

- É difícil construir um conjunto de métricas que abranjam todas as características de qualidade esperadas pelos usuários.

**Subjetiva:**

- Difícil de ser utilizada de forma freqüente.
- Avaliação depende daquele que a executa.
- Permite um posicionamento sobre o software em termos gerais.

## **2.2 – PROCESSO DE SOFTWARE**

Uma etapa inicial na busca da qualidade de software consiste na atuação em todas as etapas do seu processo e na integração das diversas áreas do seu projeto.

Com a evolução dos setores produtivos nas mais diversas áreas, o mercado brasileiro tem necessitado cada vez mais de softwares seguros e confiáveis. Com isso, foi possível chegar-se à conclusão de que o trabalho não deve ser realizado isoladamente por gerentes de projetos, desenvolvedores treinados, analista de requisitos, analista de testes, entre outros. Para que o produto de software seja produzido com qualidade é necessário trabalho em equipe, agregando competências de todas as áreas.

Para se alcançar a qualidade no desenvolvimento, metodologias apropriadas às atividades que são desenvolvidas devem ser adotadas, ou seja, processos que sejam capazes de integrar as diversas etapas do desenvolvimento de software. Uma metodologia bem desenvolvida proporciona o desenvolvimento adequado de um projeto seguro.

### **2.2.1 – Processo Unificado**

De acordo com Jacobson, I [11], o processo unificado (Unified Process-UP) de desenvolvimento de software é o conjunto de atividades necessárias para transformar requisitos do usuário em um sistema de software. Ele é baseado em componentes, e significa que o produto de software deve ser construído a partir de componentes interconectados via interfaces muito bem definidas. O processo unificado utiliza a



Linguagem de Modelagem Unificada (Unified Modeling Language – UML) [13] no preparo de todos os artefatos do sistema [14].

Com a evolução dos processos de engenharia de software, os conceitos de UP foram encampados pela Rational. Assim, o processo unificado passou a ser chamado de Processo Unificado Rational, que será descrito a seguir.

## **2.2.2 – Rational Unified Process (RUP)**

O RUP envolve todo o ciclo de desenvolvimento de software e segue uma abordagem disciplinada para atribuir tarefas e responsabilidades à equipe. Sua principal meta é garantir a produção de software de alta qualidade com cronograma e custo previsíveis, e que esteja de acordo com as necessidades dos usuários finais.

O RUP pode ser adaptado para melhor atender às exigências da organização e possui uma estrutura de processo que proporciona uma maior flexibilidade para as adaptações, tais como possibilidade de acréscimo de novos requisitos e solicitações de mudanças corretivas, adaptativas e evolutivas.

### **2.2.2.1 – Estrutura de Processo**

Para entender a estrutura do processo do RUP é necessário ter em mente que ele é composto de fases e disciplinas. As fases dizem respeito ao aspecto dinâmico do processo, ou seja, como ele é representado em procedimentos de ciclos, fases, iterações e marcos. As disciplinas, por sua vez, representam o aspecto estático, ou seja, consiste em atividades executadas em maior ou menor proporção nas fases do RUP.

O RUP trabalha com o conceito de fases de desenvolvimento. Essas fases têm um objetivo bem definido e podem possuir uma ou mais iterações. As quatro fases do RUP são: Concepção, Elaboração, Implementação e Transição.

Na Concepção os CDU de negócio são estabelecidos e o escopo do projeto delimitado [12]. Nessa fase todas as entidades externas com as quais o sistema interage (atores) são identificadas e a natureza dessa interação é definida em alto nível. Aqui também se inclui a

identificação dos critérios de sucesso, avaliação de risco, estimativa dos recursos necessitados.

Na Elaboração são feitos os levantamentos dos requisitos e dos planos que estejam estáveis para prever o restante do projeto, são produzidos os protótipos, são concebidas as arquiteturas que suportarão os requisitos e é estabelecido um ambiente de apoio para o projeto.

A Construção é a fase na qual o software é desenvolvido de uma maneira iterativa e incremental, até que o produto seja finalizado. As interações entre as diversas partes que compõem o software, uma vez implementadas nesta fase, já viabilizam testes mais integrados, não sendo necessário aguardar até a fase da transição. Desta forma, as chances de se incorrer em erros futuros e complexos diminuem.

Na Transição o sistema será apresentado ao usuário final. Nesse contato, os usuários farão suas próprias verificações e testes, podendo solicitar alterações e correções que serão encaminhadas aos responsáveis para que sejam efetuadas gerando melhorias. O usuário ou cliente somente dará o aceite formal do produto de software quando este desempenhar suas atividades de forma satisfatória, atendendo aos requisitos estabelecidos.

Cada fase é composta por disciplinas bem definidas, que são agrupamentos de atividades referentes a um mesmo assunto técnico ou gerencial. A seguir são listadas essas disciplinas:

1. Modelagem de Negócio
2. Requisitos
3. Análise e Projeto
4. Implementação
5. Testes
6. Implantação
7. Configuração e Gerência de Mudanças
8. Gerenciamento de Projetos
9. Ambiente

O relacionamento entre disciplinas e fases pode ser mais bem visualizado na Tabela 2.1 Essa figura é conhecida como gráfico de baleias. Nela pode ser visto que as disciplinas

estão presentes em quase todas as fases e apresentam um nível de atividade em maior ou menor grau de acordo com as fases.

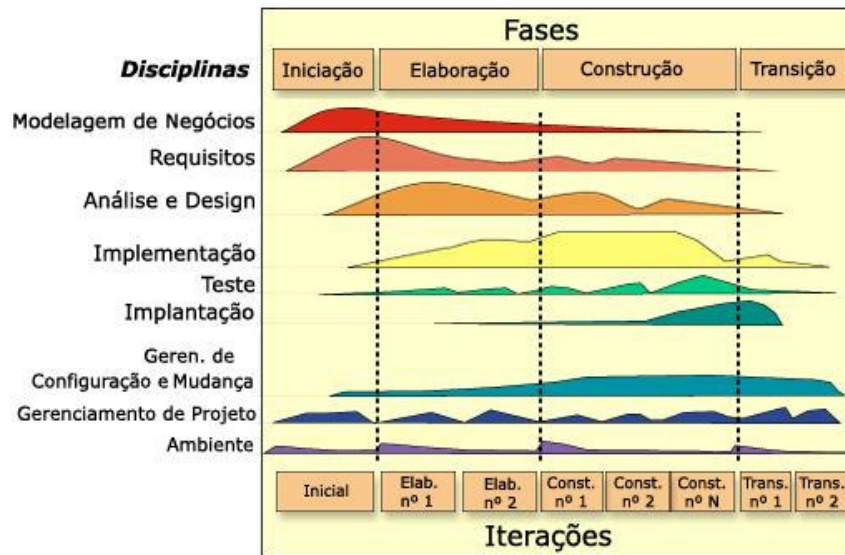


Figura 2.1 - Descrição do RUP

## 2.3 – MODELOS DE GESTÃO DA QUALIDADE NO PROCESSO DE SOFTWARE

Existem vários modelos de gestão da qualidade no processo. Este trabalho utiliza o modelo CMMi, que consiste na evolução do modelo CMM. Ambos os modelos são descritos nos itens seguintes.

### 2.3.1 – Modelo de Capacidade e Maturidade (CMM)

O CMM foi criado pelo Instituto de Engenharia de Software (SEI – *Software Engineering Institute*). O SEI é um centro de pesquisa e desenvolvimento da *Carnegie Mellon University*. O primeiro CMM desenvolvido pelo SEI foi o SW-CMM (CMM for Software), focado apenas na disciplina da engenharia de software. Depois desse, outros CMM's foram desenvolvidos, focados em outras disciplinas, como engenharia de sistemas, subcontratação, pessoas e desenvolvimento integrado de produtos.

O CMM originou-se de pesquisas e trabalhos do SEI, baseando-se na visão de Watts Humphrey [15], pesquisador vindo da IBM. Seus trabalhos [9] objetivaram adaptar uma metodologia de qualidade de processos para a área de software, para que esta pudesse ser adotada para a contratação de produção externa de software, no sentido de estabelecer um nível de maturidade que o prestador de serviços deveria ter para atender estas demandas, de acordo com CMU/SEI (1997) [14].

O SW-CMM descreve os elementos-chave da evolução de um processo de software imaturo para um processo maduro e disciplinado. Abrange práticas para planejamento, engenharia e gestão do desenvolvimento de software que, quando seguidas, melhoram a habilidade da organização em atender metas para custos, cronograma, funcionalidade e qualidade do produto.

A classificação de maturidade pelo CMM possui os seguintes níveis:

- **Nível 1** - Inicial: o processo de software é considerado ad hoc e até mesmo caótico. Poucos processos são definidos e o sucesso depende de esforços individuais [18].
- **Nível 2** - Repetível: processos básicos de gerenciamento de projetos são estabelecidos para controle de custos, cronograma e escopo. A disciplina garante a repetição dos processos implementados com sucesso em projetos de características similares.
- **Nível 3** - Definido: os processos de software para gerenciamento e engenharia são documentados, padronizados e integrados para toda organização. Todos os projetos utilizam uma versão aprovada e customizada dos processos padrões da organização para desenvolver e manter softwares.
- **Nível 4** - Gerenciado: medições detalhadas da qualidade dos processos e produtos são coletadas. Tanto os processos quanto os produtos de software são gerenciados quantitativamente.
- **Nível 5** - Otimizado: a melhoria contínua do processo de software é permitida pelo *feedback* quantitativo do processo e por idéias e tecnologias inovadoras testadas na organização.

### 2.3.2 – CMM Integrado (CMMi)

Para solucionar os problemas gerados pelo surgimento de outros CMM's e de normas não contempladas, o SEI criou uma nova versão do CMM, chamada CMMi, para resolver esses problemas (vide SEI, 2004) [16].

O objetivo do CMMI é integrar os diversos CMM's numa estrutura única, todos com a mesma terminologia, processos de avaliação e estrutura. Além disso, o projeto também se preocupou em tornar o CMM compatível com a norma ISO/IEC 15504 – ISO (2004) [18], de modo que avaliações em um modelo sejam reconhecidas como equivalentes aos do outro e, naturalmente, incorporar ao CMM as melhorias sugeridas por CMU/SEI (1997) [16] que aparecem ao longo dos anos.

O CMMI é um guia desenvolvido pela comunidade de software sendo um modelo único que integra os CMM's também. Também, incorpora as necessidades de melhorias identificadas pelo uso do CMM no mundo, é compatível com a norma ISO/IEC 15504 [18] e é alinhado com o Corpo de Conhecimento de Gerência de Projetos (PMBOK – *Project Management Body of Knowledge*) [21]. Além disso, ele também possui um direcionamento claro e objetivo para a interpretação das práticas, apresentando sub-práticas e produtos típicos de trabalho para cada prática, sendo mostrado em duas representações, vistas em SEI (2004): a contínua e a em estágios, que possuem níveis de capacidade e maturidade, respectivamente. Na Tabela 2.1 a seguir são ilustrados os níveis para cada representação.

Tabela 2.1 - Níveis do CMMI

Nível	Contínua (Capacidade)	Em Estágios (Maturidade)
Nível 0	Incompleto	
Nível 1	Realizado	Inicial
Nível 2	Gerenciado	Gerenciado
Nível 3	Definido	Definido
Nível 4	Gerenciado Quatitativamente	Gerenciado Quatitativamente
Nível 5	Otimizado	Otimizado

A representação estagiada pode ser utilizada para verificar o nível de maturidade da organização em geral [19]. Cada área de processo pertence exclusivamente a um dos cinco

níveis de maturidade. Para obter certificação em determinado nível, a organização precisa atender às metas de todas as áreas de processo definidas para o nível e às metas das áreas de processo definidas para os níveis anteriores.

A estrutura desta representação do modelo é composta por cinco níveis de maturidade. Cada nível de maturidade é composto de Áreas de Processo (*PA's – Process Areas*), compostas de metas específicas e genéricas. As metas específicas possuem práticas específicas e as metas genéricas possuem características comuns compostas de práticas genéricas.

A representação contínua pode ser utilizada para verificar o nível de capacidade dos processos. Nesta, cada área de processo possui características relativas a mais de um nível. Assim, uma área de processo que, na representação estagiada, pertence exclusivamente ao nível dois, no modelo contínuo pode ter características que a coloquem em outros níveis, ou seja, cada área de processo é classificada separadamente.

De acordo com o guia publicado em SEI (2004), a representação contínua do CMMI está organizada em seis níveis de capacidade, numerados de zero a cinco, que fornecem uma ordem recomendada aproximando a melhoria dos processos dentro de cada área. A medida que as metas específicas e genéricas são atingidas para uma área de processo em um determinado nível de capacidade, os benefícios da melhoria de processos são obtidos.

Os níveis de capacidade são focados no aumento da capacidade da organização em executar, controlar e melhorar o seu desempenho em uma área de processo. Ademais, os níveis de capacidade permitem monitorar, avaliar e demonstrar a evolução de uma organização na melhoria dos processos associados a uma área.

## **2.4 – CICLO DE VIDA DE SOFTWARE**

O ciclo de vida do software vai desde a concepção de idéias até a descontinuidade do produto de software [6]. Durante este ciclo são executados vários processos, sendo que cada um contribui para atingir os objetivos do ciclo [22].

Uma taxonomia para o ciclo de vida de software é definida pela Norma NBR ISO/IEC 12207 [24] – Processos de ciclo de vida do software. Essa norma tem como objetivo definir todos os processos necessários para que a aquisição, o fornecimento, a operação, o desenvolvimento e a manutenção do software possam ocorrer com qualidade. A norma agrupa as atividades que podem ser executadas durante o ciclo de vida do software em: processos fundamentais (aquisição, fornecimento, desenvolvimento, manutenção e operação), processos de apoio (documentação, gestão de configuração, garantia da qualidade, verificação, validação, revisão conjunta, auditoria, resolução de problemas, usabilidade e avaliação de qualidade de produto) e processos organizacionais (gerência organizacional, melhoria, infra-estrutura e recursos humanos).

Os modelos de ciclo de vida são categorizados pela definição de uma seqüência de atividades pré-definidas. Alguns modelos são do tipo:

- **Cascata** – esse foi o primeiro modelo a ser conhecido em engenharia de software e consiste basicamente num modelo linear em que cada passo deve ser completado antes que o próximo passo possa ser iniciado.
- **Incremental** – utilizado no processo unificado, tem como premissa a resolução dos riscos por que é centrado na arquitetura, incremental e orientado a CDU [25].
- **Espiral** – Esse modelo é dirigido a risco [24], pois em cada iteração temos uma análise de risco do projeto em si. O risco é visto como uma parte integrante do processo de desenvolvimento de software.

## 2.5 – GERÊNCIA DE QUALIDADE DE SOFTWARE

Propõe-se neste item um modelo para a Gerência de Qualidade de Software. Essa atividade tem o objetivo de elaborar um planejamento – Plano de Qualidade – para a condução das atividades de verificação da conformidade do processo e dos produtos do projeto quanto aos padrões e procedimentos estabelecidos para a garantia da qualidade de software.

A Gestão da Qualidade de Software (*Quality Assurance*) é a responsável por exercer as seguintes atividades no Processo de Software:

- a. assessorar a equipe de projeto quanto aos padrões e procedimentos do Processo para a garantia da qualidade de software;
- b. checar a conformidade das atividades e dos produtos do projeto quanto aos padrões e Processo estabelecidos;
- c. realizar as revisões de qualidade estabelecidas no Processo;
- d. auditar as atividades realizadas pela equipe de projeto e pelo cliente para verificar se eles estão procedendo de acordo com os padrões adotados e com os procedimentos estabelecidos no processo;
- e. comunicar ao gerente do projeto os problemas encontrados e verificar as providências tomadas para a solução desses problemas.

O “Plano da Qualidade” deve estar alinhado ao Projeto e deve ser validado pelo Gestor do Projeto.

Também, deve haver acompanhamento para verificação da conformidade do processo e dos produtos do projeto ao final de cada fase – Concepção, Elaboração, Construção e Transição. A Tabela 2.2, a seguir, propõe-se um resumo acerca da forma de controle na qualidade do software.

Tabela 2.2 - Gerência de Qualidade de Software

Objetivos:	Verificação da conformidade das atividades e produtos estabelecidos para a garantia da qualidade de software.
Responsáveis:	Gestor da Qualidade de Software.
Envolvidos:	Gerente do Projeto.
Entradas:	Atividades realizadas e produtos gerados no Processo. Plano da Qualidade do Software.
Saídas:	Atividades e produtos em conformidade com os padrões adotados. Plano da Qualidade do Software atualizado. Controle da Qualidade.
Quando deve ser aplicado:	Em todas as fases, inclusive durante a fase de Produção.

Essa atividade tem o objetivo de verificar a conformidade das atividades e dos produtos do projeto quanto aos padrões e procedimentos estabelecidos para a garantia da qualidade de software, e acontece durante todo o ciclo de vida do projeto.



O Gestor da Qualidade de Software é o responsável pela execução desta atividade conforme estabelecido no “Plano da Qualidade do Software”.

Ao final da fase de Concepção, ocorre uma revisão da documentação dos requisitos para assegurar que os padrões aplicáveis ao projeto estejam sendo aplicados e que os requisitos levantados e documentados forneçam informações suficientes para a fase de Elaboração. Os itens levantados durante essa atividade são registrados no documento de “Controle da Qualidade”.

Ao final da fase de Elaboração, é recomendável que se valide, com a área responsável pela Análise de Requisitos, as especificações propostas (CDU, interfaces, modelos etc.), considerando, sempre que possível, os seguintes aspectos:

- a. Discussão das funcionalidades descritas;
- b. Validação das informações e padrão das interfaces;
- c. Discussão dos relacionamentos definidos nos diagramas.

Ao final da fase de Construção, é recomendável que se valide com os implementadores os códigos gerados, avaliando sua padronização, desempenho, etc. Também são validados os resultados dos testes, verificando-os desde o seu planejamento até os eventos de erros e correções encontrados, certificando, portanto, a qualidade final do produto.

Ao final da fase de Transição, é recomendável que se avalie a satisfação do cliente quanto ao produto implantado.

## **2.6 – TESTES**

O teste consiste em submeter um produto de software ou uma de suas partes integrantes a uma condição controlada que tem o objetivo de simular um evento real de modo a permitir que erros, falhas ou inconsistências possam ser identificados. Durante a realização dos testes, os resultados são observados e registrados.

A validação do projeto é realizada por meio de uma série de testes sobre o produto de software totalmente implementado e integrado, e objetiva assegurar que o mesmo esteja em conformidade com as necessidades e requisitos definidos pelo usuário.

Os requisitos são as condições nas quais uma exigência deve ser satisfeita e podem ser classificados em:

- **Requisito funcional** – representa uma função que o produto deve realizar para atender a uma expectativa dos usuários.
- **Requisitos não-funcionais** – são condições que permeiam o funcionamento de um software, mas não estão diretamente associadas à realização de uma função. Por exemplo, o sistema pode produzir uma determinada saída para uma determinada entrada e, assim, atender a um requisito funcional, contudo, o tempo para realizar essa função pode não atender a uma demanda do usuário. Portanto, os requisitos não-funcionais incluem condições de desempenho, ergonomia, capacidade, etc.

## 2.7 – MÉTRICAS

Uma métrica é um dado numérico ao qual pode ser atribuída uma meta com o intuito de permitir a tomada de ação por parte dos gerentes do projeto. Ela deve ser calculada periodicamente, de modo a permitir a construção de uma análise histórica, bem como a construção de estimativas de pequeno, médio e longo prazo. Um dos seus principais objetivos é permitir a avaliação objetiva da situação [26]. As decisões devem ser baseadas no resultado dos indicadores, considerando as tendências e os referenciais de comparação. Uma análise consistente leva em consideração não somente uma métrica isoladamente, mas um conjunto delas, de modo a permitir inferir sobre o projeto a partir das várias óticas ou situações geradas por elas.

## 2.8 – RELAÇÃO ENTRE GESTÃO DA QUALIDADE, TESTES E MÉTRICAS

Como sugestão para uma boa política de qualidade a ser adotada para uma empresa de desenvolvimento de software, propõe-se a adoção de instrumentos que garantam um modelo de qualidade para a própria empresa, a utilização de testes para eliminar falhas ou inconsistências já na fase de desenvolvimento, e a utilização de métricas para avaliar todo o processo, como sustentação da política de qualidade de software, conforme visto na Figura 2.2

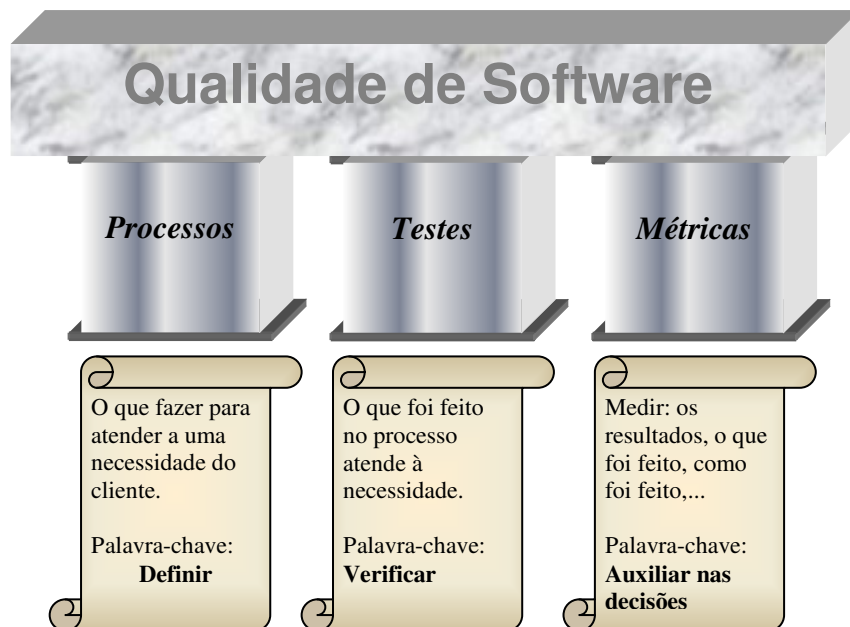


Figura 2.2 – Importância de Processos, Testes e Métricas para a Qualidade de Software

A adoção de qualquer modelo de gestão da qualidade de software deve fazer uso de ferramentas que em si avaliem se esse modelo está sendo bem implantado ou não. Os modelos de gestão de qualidade apresentam algumas características importantes, tais como:

- Documentação.
- Rastreabilidade.
- Padronização de processos.

Entretanto, a adoção dessas características, mesmo em níveis de atendimento de expectativa muito grandes não afastam a necessidade da adoção de testes a serem aplicados às várias fases do software. Este é um produto desenvolvido para a utilização em processos que pode apresentar falhas, defeitos e inconsistências como quaisquer outros produtos industriais. Por essa razão, faz-se necessário o emprego de testes.

O emprego de métricas justifica-se pela necessidade constante de avaliar qualquer processo com o objetivo de otimizar a produção e diminuir custos. Isso porque, mesmo numa situação onde todos os testes foram considerados satisfatórios, a empresa deve se preocupar com os custos por essa busca e, sobretudo, com o tempo gasto para atingir esse nível de qualidade.

Assim, sugere-se que a qualidade de software seja obtida a partir do relacionamento harmônico entre os seguintes tópicos:

**Modelo:** que diz respeito à definição de etapas, fases e processos. Esse tópico tem uma abrangência macro na medida em que estabelece as diretrizes do projeto, bem como seus aspectos operacionais.

**Testes:** que têm o objetivo de verificar a conformidade do produto aos requisitos, além de primar pela busca por falhas.

**Métricas:** cuja razão refere-se à medição de fatores específicos do projeto para subsidiar decisões gerenciais.

Um exemplo dessa ilustração pode ser visto na Figura 2.3

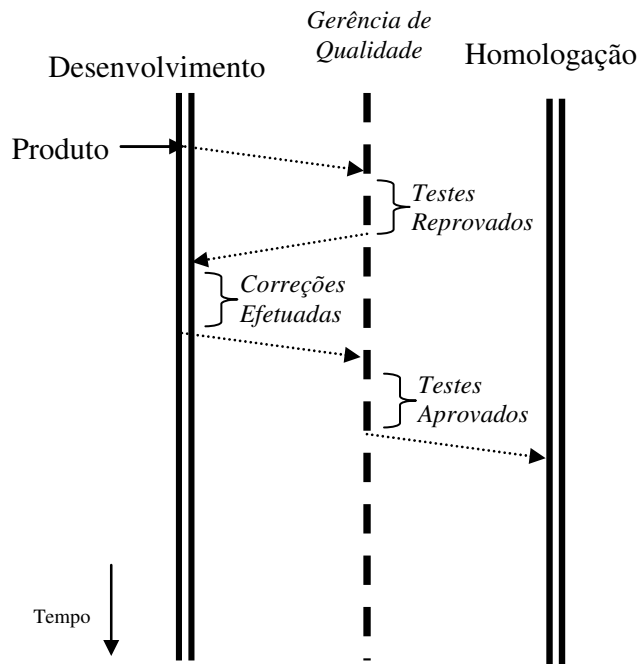


Figura 2.3 - Relação entre Gestão da Qualidade, Testes e Métricas

Os elementos da Figura 2.3 podem ser descritos da seguinte forma: as barras duplas representam as fases do projeto; a linha tracejada representa a atividade da Gerência de Qualidade; as setas representam o encaminhamento de um produto entre as fases ou entre uma fase e a gerência; e a seta pra baixo indica o sentido de progressão do tempo.

A ilustração mostra a passagem de um produto de software da fase de Desenvolvimento para a fase de Homologação. Essa passagem deve ser feita mediante a aplicação de testes pela Gerência de Qualidade. Se os testes forem satisfatórios, estes seguem para a Fase de Homologação. Caso contrário, o produto retorna para os desenvolvedores. Possíveis métricas a serem utilizadas nesse cenário simplificado consistem na medição do tempo gasto nos testes, tempo médio para um produto passar do desenvolvimento para homologação etc.

Assim, o modelo delimita todas as fases do processo, bem como o papel que cada um dos atores vai desempenhar. Os testes são aplicados em cada uma das fases com o objetivo de validar o requisito funcionalidade efetivamente implementado. E as métricas são coletadas de modo a ser possível avaliar todo o processo e contribuir para as adequações necessárias.

## 2.9 – AVALIAÇÃO DOS PROCESSOS

Na Figura 2.4 sugere-se um processo onde a análise do resultado das métricas e a realimentação dos usuários fornecem subsídios para as gerências de qualidade e de projeto efetuarem as alterações necessárias.

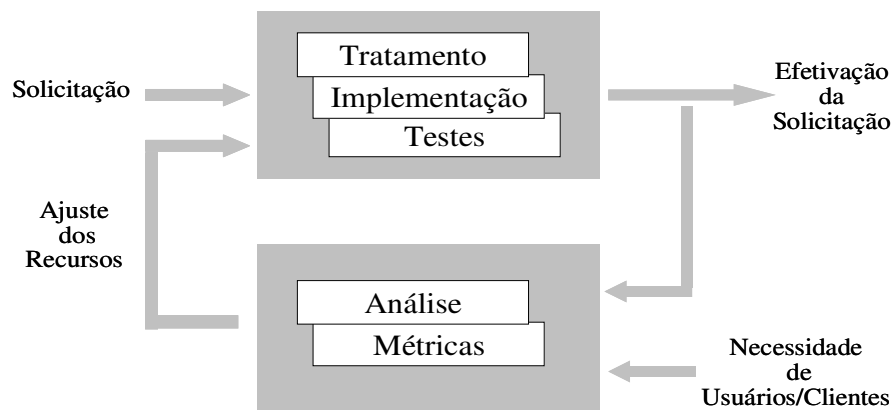


Figura 2.4 - Avaliação dos processos.

A partir de uma solicitação, as equipes responsáveis efetuam o devido tratamento e implementam as mudanças em observância ao processo de software definido, inclusive no que se refere à aplicação de testes. Só então o resultado da solicitação é efetivado.

Se o processo corresse em uma via, ou seja, um produto passasse da solicitação para a efetivação, seria necessária a confiança total no processo de software estabelecido. Contudo, na prática, todo processo deve prever a possibilidade de ser alterado de modo a se adaptar às circunstâncias.

Nesse sentido, o resultado da efetivação da solicitação, o tempo gasto para a efetivação e a percepção do cliente são parâmetros que fornecem subsídios para a análise de todo o processo. Esses subsídios permitem a realização de ajustes no processo e dimensionamento das equipes e recursos pelas gerências de qualidade e de projeto.

## **3 – TESTES**

### **3.1 – DESENVOLVIMENTO DO PRODUTO**

No RUP [11], a disciplina de testes tem como propósito verificar a interação entre os objetos e se todos os requisitos foram implementados corretamente, para garantir a conformidade do produto. Nessa disciplina é possível encontrar defeitos e inconsistências, e determinar quando o sistema estará pronto para ir para as fases de produção e operação, além de ser possível assegurar que as correções foram tratadas antes da entrega do produto de software.

Ao se aplicar testes em sistemas, vê-se que um grande número de abordagens e tipos de testes foram desenvolvidos. Entretanto, é imperativo que se saiba que a garantia absoluta da qualidade de um sistema é uma meta difícil de ser atingida. Acima de um determinado nível de confiabilidade, os custos empregados para tal tornam-se proibitivos para a consecução do projeto. Assim, na prática, admitem-se níveis de segurança inferiores a 100%.

Partindo-se da premissa de que a qualidade absoluta não pode ser atingida, pode-se inferir que a partir de um certo patamar, o aumento de qualidade não acompanha de igual maneira os recursos gastos, conforme é ilustrado, a título de exemplo, na Figura 3.1. Nesta figura, os custos podem ser entendidos como financeiros, tempo, pessoal, complexidade, entre outros. Quando o nível de qualidade não é estabelecido na fase inicial do projeto, a equipe de desenvolvimento deve decidir em qual nível de qualidade o sistema será colocado na fase de produção.

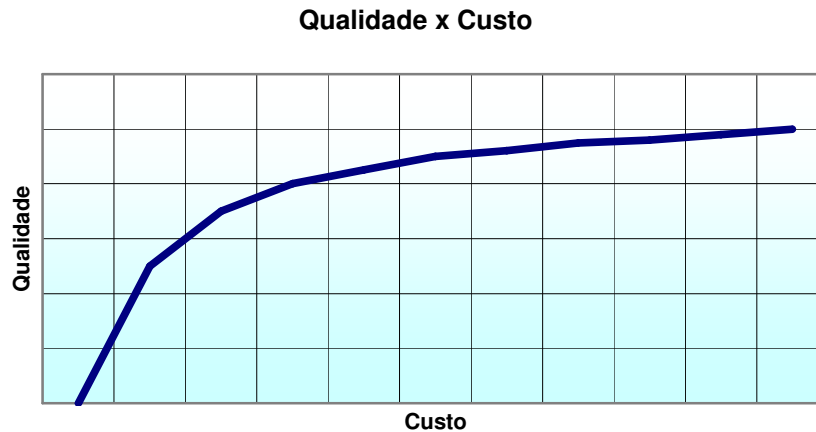


Figura 3.1 - Desenvolvimento do Produto.

Na disciplina de testes, o resultado da implementação é verificado por meio do teste de cada configuração e da versão final do sistema. As finalidades da disciplina de testes são:

- Planejar os testes necessários a cada iteração, incluindo os testes integração e os testes do sistema. Os testes de integração são requeridos para cada configuração de iteração. Já os testes do sistema são requeridos somente no fim da iteração.
- Projetar e implementar os testes, criando CDT com especificações e procedimentos detalhados da execução dos mesmos, e, se possível, componentes executáveis para automatizá-los.
- Executar, sistematicamente, vários testes e assegurar que os resultados estão corretos.

Algum planejamento inicial do teste pode ocorrer durante a fase de concepção, também conhecida como fase de iniciação. Entretanto, o mais comum é que o teste só seja realizado quando as configurações são integradas, conforme a Figura 3.2. Durante a fase de transição, o foco dos testes muda para os defeitos detectados durante o uso do sistema.



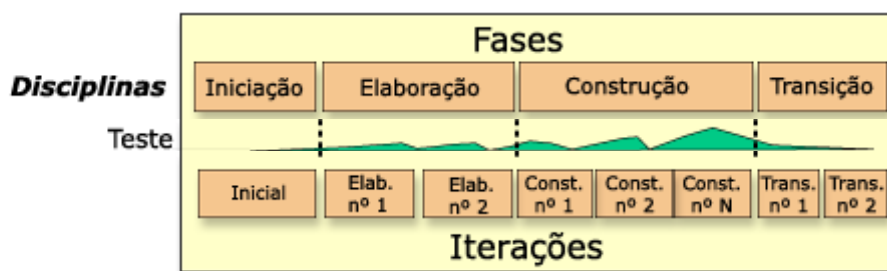


Figura 3.2 - Disciplina de Teste.

Os artefatos podem ser um modelo, um elemento de modelo ou um documento. Os artefatos de interesse neste trabalho são:

- a. **Modelo de Teste** – descreve como os componentes executáveis do modelo de implementação são testados. Além disso, ele pode também descrever como os aspectos específicos do sistema devem ser testados.
- b. **CDT** – é um conjunto de entradas de teste, condições de execução e resultados esperados desenvolvidos para um objetivo como, por exemplo, testar o caminho de determinado programa ou verificar o cumprimento de um requisito [12].

Na Tabela 3.1 é proposto um modelo de CDT contendo suas principais informações e definições.

Tabela 3.1 - Informações de um CDT

Usuário/Cliente	Nome da empresa que demandou os projetos.
Projeto	Nome do empreendimento a ser realizado que foi contratado pelo Usuário/cliente.
Versão do CDT	Cada uma das formas sucessivas que um CDT apresenta, à medida que se desenvolve o processo de sua elaboração e aperfeiçoamento.
Aceite	Manifestação de concordância do usuário/cliente referente ao CDU.
Identificação	Nome do CDT.
Atores	Nome do órgão a quem se destina o desenvolvimento do Projeto.
Pré-condições	Define os parâmetros iniciais.
Pós-condições	Define os parâmetros de saídas.

Caminho da tela	Seqüência de ações ou comandos que devem ser executados a fim de que se chegue ao ponto onde o teste é efetuado.
Cenários	Panoramas resultantes da combinação de situações referentes ao inter-relacionamento entre os fluxos básico, alternativos, de exceções e observações.
Fluxo do CDT	Identificação de quais fluxos estão sendo testados. Essa identificação deve obedecer ao disposto no CDU equivalente.
Descrição do teste	Identifica o que deve ser feito no CDT.
Resultado esperado	Identifica o que o sistema deve trazer de resposta.
Rotinas de testes	São passos onde são descritos como deve ser testado campo a campo da tela.
Controle de versão	
Versão	Número da versão.
Data	Data da versão.
Descrição da alteração	Descrever o que foi alterado no CDT.
Autor	Nome do responsável pela alteração.

Para iniciar a especificação do CDT é preciso ter toda a parte de análise e projeto definida. Essas partes são constituídas das seguintes etapas:

- a. Levantamento dos Requisitos para elaboração dos CDU;
- b. Análise Arquitetural;
- c. Protótipo.

A Figura 3.3 a seguir ilustra o inter-relaciamento entre as fases, bem como o posicionamento da atividade de especificação de teste.

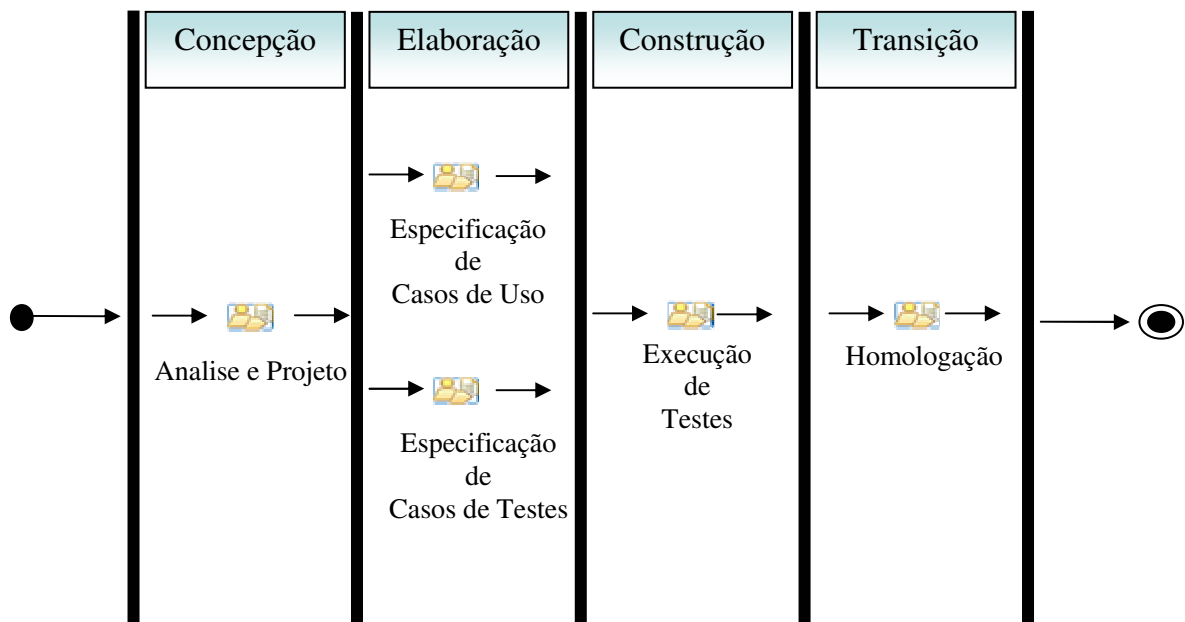


Figura 3.3 - Atividade de Especificação de Testes nas fases do Processo.

A partir do CDU, o analista de teste deve criar cenários para cada fluxo de dados: fluxo básico e fluxos alternativos. A especificação de CDT também utiliza o protótipo e modelo de dados. Durante a criação dos cenários, devem ser observados os seguintes aspectos:

- a. entrada;
- b. resultado esperado;
- c. dados em cada campo;
- d. regra de negócios;
- e. exceções; e
- f. observações.

A atividade de execução de testes, cujas etapas são mostradas na Figura 3.4, ocorrerá após sua especificação. Entretanto, esse é apenas um dos pré-requisitos, pois o CDU correspondente ao teste já deve ter sido implementado.

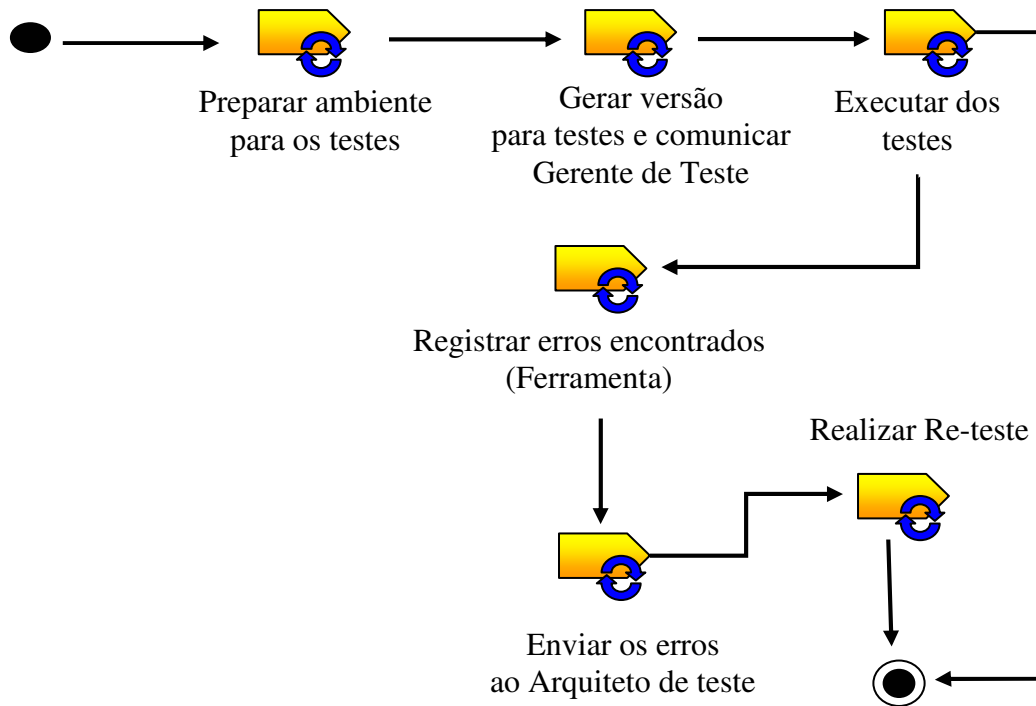


Figura 3.4 - Diagrama de Atividade de Execução de Teste.

### 3.1.1 – Papéis

Os papéis podem ser definidos como o comportamento e as responsabilidades de um indivíduo, ou conjunto de indivíduos trabalhando em equipe, dentro do contexto de uma organização de engenharia de software.

Os papéis, bem como as responsabilidades dos profissionais da área de testes são listados a seguir [29] [30]:

- **Gerente de Teste:** é responsável pela elaboração do plano de teste, distribuição das tarefas aos testadores, organização da documentação do teste, controle do registro dos erros e da realização dos testes, verificação das atividades pendentes após datas limites de término e homologa o término dos testes, baseado em informações fornecidas pelos testadores.
- **Analista de Teste:** suas atribuições consistem em inicialmente identificar e posteriormente definir os testes necessários, monitorar a abrangência dos testes e avaliar a qualidade geral obtida ao testar os Itens de Teste-alvo. É responsável pela criação dos CDT juntamente com as rotinas de testes.
- **Arquiteto de Teste:** é o profissional cuja responsabilidade consiste na montagem da infra-estrutura de testes. Também é responsável pela liderança técnica do trabalho de teste e pela comunicação entre a equipe de teste e a equipe de projeto (ou equipe de desenvolvimento).
- **Testador:** é o responsável por testar o produto de software. Ele realiza especificamente as tarefas que foram designadas a ele pelo Gerente de Teste. O Testador deve ter a responsabilidade de executar os CDT e criar *scripts* para verificar se os requisitos estão de acordo com o que o cliente solicitou.

### 3.1.2 – Tipos de testes

Há testes específicos para se verificar diversas dimensões de um produto de software. Abaixo são descritos alguns tipos de testes [28] [29] [30] [33]:

- **Teste de integração:** verifica se todos os componentes são executados de modo a assegurar o correto funcionamento das interfaces.
- **Teste de sistema:** nesta etapa todo o sistema é testado dentro de um ambiente operacional para assegurar o correto funcionamento da aplicação, de acordo com as solicitações do usuário.
- **Teste de aceitação:** é realizado pelos usuários para que o produto seja validado. São conduzidos pela equipe de testes, juntamente com a equipe do projeto. Esse teste assegura que seus requisitos estejam corretos antes de serem encaminhados para o ambiente de produção.
- **Teste de acessibilidade:** verifica se a interface do usuário fornece o acesso apropriado às funções do sistema e a navegação adequada. Além disso, esses testes garantem que os objetos dentro da interface do usuário funcionem de acordo com os padrões definidos pelo usuário.
- **Teste de integridade de banco de dados:** nestes, os bancos de dados e as regras de negócio devem ser testados de forma independente. Este tipo de teste não deve ser realizado por meio da interface de usuário definida para o acesso aos dados.
- **Testes de regressão:** devem ser executados na versão atual do sistema cujo enfoque se dá em partes do sistema que funcionavam corretamente em versões anteriores. Isso se deve ao fato de que toda vez que se insere uma característica nova no sistema deve-se testar toda a aplicação novamente, pois não se sabe o impacto dessa alteração nas funcionalidades que já haviam sido testadas com sucesso. Existem dois tipos de teste de regressão: total e parcial.
  - a. **total** – são realizados todos os testes sem qualquer exceção. Independente das mudanças que ocorreram, deve-se testar todo o sistema;
  - b. **parcial** – são realizados testes em um subconjunto de CDT. Esse subconjunto é definido a partir do inter-relacionamento existente entre os CDT e as modificações efetuadas.
- **Teste de carga (Stress):** seu objetivo é verificar o comportamento de um software em situações críticas. As situações críticas podem ser entendidas como aquelas que fogem ao previsto no planejamento de funcionamento do software para situações

normais. Assim, o aumento no número de transações, volume de dados, número de usuários podem levar a situações de “stress”.

- **Teste de segurança:** verifica se o sistema apresenta falhas de segurança. Estas podem ocorrer em verificação de senhas, códigos, criptografia de dados, permissões de acesso, entre outras. O maior objetivo desse tipo de teste é a garantia da integridade dos dados, o sigilo de informações e a execução de ações somente por pessoas autorizadas.
- **Teste de performance:** avalia se o desempenho do software corresponde aos requisitos definidos para situações específicas, tais como: pico máximo de acesso, pico máximo de operações, tempo de resposta a comandos.
- **Teste de usabilidade:** avalia o nível de facilidade na utilização do sistema. Compreende-se como facilidade o entendimento das funções do sistema pelo usuário por meio da utilização de manuais de ajuda, assistentes eletrônicos, etc.
- **Teste de volume:** é um tipo de verificação onde são submetidas grandes quantidades de dados ao sistema para determinar se limites que causam a falha do software são alcançados. Este tipo de teste também identifica a carga ou volume máximo persistente que o sistema pode suportar por um dado período.
- **Teste de instalação:** possui dois propósitos: garantir que o software possa ser instalado (completamente, de forma personalizada ou atualizado) sob condições apropriadas e verificar se o software funciona corretamente depois de instalado.
- **Teste de configuração:** verifica a operação do sistema em diferentes configurações de software e hardware, pois estas podem afetar o sistema a ser desenvolvido.

### 3.1.3 – Abordagens de Teste

Existem duas abordagens para conduzir um processo de validação: abordagem da “caixa branca” [28] [29] e abordagem da “caixa preta” [28] [29] [31]. A estratégia escolhida determina como se pode confrontar o problema e como serão conduzidos os procedimentos de testes visando a minimizar os esforços e ampliar a abrangência das verificações de não conformidades e das detecções de erros relacionados ao software.

### 3.1.3.1 – Abordagem do teste da caixa branca

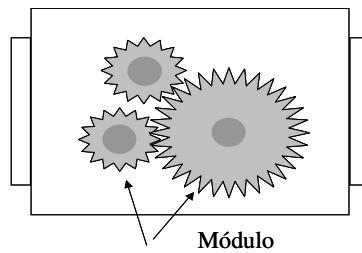


Figura 3.5 - Estratégias dos testes caixa branca.

O teste da caixa branca, apresentado na Figura 3.5, consiste em uma categoria de testes onde as etapas intrínsecas ao processo são consideradas, ou seja, o código, fonte, módulos, rotinas, entre outras. Esses testes empregam técnicas que estão embutidas na arquitetura interna do software.

### 3.1.3.2 – Abordagem do teste da caixa preta

O teste da caixa preta consiste em testes funcionais para garantir que os requisitos do sistema são plenamente atendidos pelo software que foi construído [31]. Essa estratégia tem o objetivo de verificar o sistema externamente. De um modo simplista, verificar se os “*outputs*” do sistema coincidem com o comportamento esperado para um “*input*” específico. A Figura 3.6 ilustra a representação de um teste caixa preta, onde apenas as saídas correspondentes a determinadas entradas são levadas em consideração, ou seja, as etapas intermediárias do processo não são observadas.

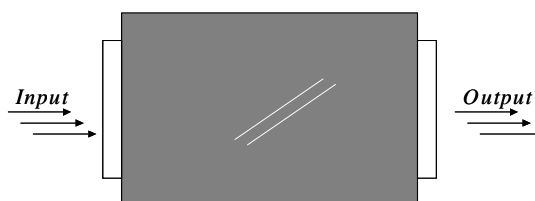


Figura 3.6 – Estratégias dos testes caixa preta.



Neste contexto, o teste funcional permite avaliar o comportamento do sistema. Mas, para isso, devem-se criar rotinas e CDT e seguir passo a passo os procedimentos e rotinas estabelecidas para esses testes.

#### **3.1.4 – Classificação quanto à forma de execução**

- **Testes Manuais:** essa é uma abordagem na qual os testes são realizados diretamente pelos profissionais envolvidos, e todas as etapas e itens são avaliados por esses profissionais. Assim, necessita-se de recursos humanos para execução de todos os CDT que foram especificados. A principal característica desse tipo de teste é o custo envolvido e a execução lenta. Dependendo do tamanho do sistema, dos recursos disponíveis pela empresa de Tecnologia da Informação (TI) e do cronograma do projeto, a adoção dos testes manuais não é uma estratégia viável.
- **Testes Automatizados:** diferentemente da abordagem dos testes manuais, na automatização os testes são executados por meio de ferramentas apropriadas para realizar um determinado objetivo. Essas ferramentas dispensam recursos humanos para as etapas intermediárias de uma tarefa. Como exemplo de aplicações de testes, pode-se citar: Teste de Funcionalidade, Teste de Carga, etc.

### **3.2 – ESSENCIABILIDADE DOS TESTES**

Um único sistema pode ser submetido a vários tipos de teste. A gerência responsável pela gestão de qualidade, juntamente com o gerente de teste, deve decidir quais os testes serão empregados. Sugere-se que alguns fatores devem ser levados em consideração, tais como: riscos envolvidos no projeto, relação custo-benefício, etc. Tendo-se em mente esses direcionadores, o estabelecimento dos testes priorizará os aspectos vitais, adentrando em outras características e tipos de testes em função dos recursos disponíveis.

### 3.3 – PROCESSO DE ESCOLHA DO TIPO DE TESTE

Na escolha dos testes, propõe-se que todos os cenários devem contemplar a realização de testes funcionais. Pois, a principal característica desses testes reside no fato de que os requisitos pré-determinados são atendidos, ou seja, para um determinado conjunto de dados inseridos em um sistema, o sistema apresentará uma saída condizente.

Quanto aos demais testes, entende-se que a decisão sobre quais serão escolhidos deve considerar as peculiaridades de cada sistema e as necessidades dos clientes.

Critérios consideráveis:

- a. Requisitos
- b. Segurança
- c. Tempo de Resposta
- d. Confiabilidade
- e. Privacidade
- f. Integridade
- g. Estabilidade
- h. Manutenibilidade

Dependendo da finalidade de cada sistema, alguns critérios sobressaem aos demais. Isso pode ser percebido quando se trata com tipos específicos, tais como:

- **Sistemas bancários:** o critério de Segurança, Confiabilidade e Integridade têm um peso bastante importante no processo de desenvolvimento.
- **Sistemas de comunicação:** Tempo de Resposta e Estabilidade são critérios importantes.
- **Sistemas comerciais:** maior importância ao atendimento de Requisitos.

Entretanto, essas são apenas características gerais para escolha dos critérios. Alguns dos sistemas podem fazer parte de ambientes complexos cuja demanda por critérios mais específicos deve ser levada em consideração. Nesses casos, as experiências da equipe de desenvolvimento juntamente com a percepção dos clientes devem ser consideradas.

### **3.4 – PROPOSTA PARA APLICAÇÃO DE TESTE**

A aplicação de teste consiste na execução de rotinas de teste, que por sua vez constituem-se de um ou mais CDT, a depender do modo de reproduzir os conjuntos de transações de CDU [32]. Para cada rotina de teste definido, devem ser previstos os acertos e os erros correspondentes. Ademais, em alguns CDT será necessário calcular os valores exatos dos resultados, sendo que nos demais casos, bastará ser obtida a classe de saída correspondente [33] [34].

Depois da criação dos cenários de teste, bem como de suas rotinas, os testes serão executados e os resultados obtidos. Caso ocorra erro, deve ser criado um documento de solicitação de manutenção onde os mesmos serão registrados. Em seguida, eles deverão ser repassados aos responsáveis para que as correções ou adaptações necessárias sejam efetuadas.

Esses erros podem ser tanto de implementação quanto de especificação, tendo tratamento distinto conforme o caso. Os erros de implementação são aqueles referentes a não conformidades entre o que foi definido nos CDU e a funcionalidade implementada na versão testada. Os erros de especificação dizem respeito a falhas e inadequações oriundas de fatores como:

- levantamento de requisitos;
- regras de negócio;
- prototipação;
- layout, dentre outros.

Uma vez encontrados os erros, após a realização dos testes na fase de desenvolvimento, a versão do software é enviada para as correções ou adaptações pertinentes. Caso sejam erros de especificação, o analista de requisito, com o consentimento do cliente fará as devidas alterações dos CDU. Então esses serão passados ao gerente e logo após para os desenvolvedores, juntamente com as solicitações de manutenção. Desta forma, os programadores implementarão a nova regra de negócios. Caso sejam erros de implementação, esses devem ser repassados diretamente aos desenvolvedores para as correções.

Concluída essa fase, será enviada uma nova versão onde os testes serão re-executados, chamados retestes. Os retestes devem considerar todos os CDU, e não somente aqueles que apresentaram erros. Uma pergunta que poderia surgir é: por que todos serão retestados? A razão para o trabalho fundamenta-se na necessidade de abranger todo o projeto e avaliar a possível existência de impactos negativos, inclusive nos casos que não apresentaram erros. Devido ao inter-relacionamento entre os módulos de um projeto, essa possibilidade não é rara de acontecer. Essa técnica é chamada de teste de regressão.

Se um CDU for retestado na nova versão e ainda persistirem erros, ou ainda, surgirem novos erros, novas solicitações devem ser enviadas aos desenvolvedores, até que não sejam encontrados mais erros. Só após não ser mais possível constatar erros durante a aplicação dos CDT, será possível encaminhá-lo à base de homologação dos usuários para que sejam feitos os testes do sistema. Na Figura 3.7 é proposto um fluxograma para o Processo de Testes descrito nesta seção [35].

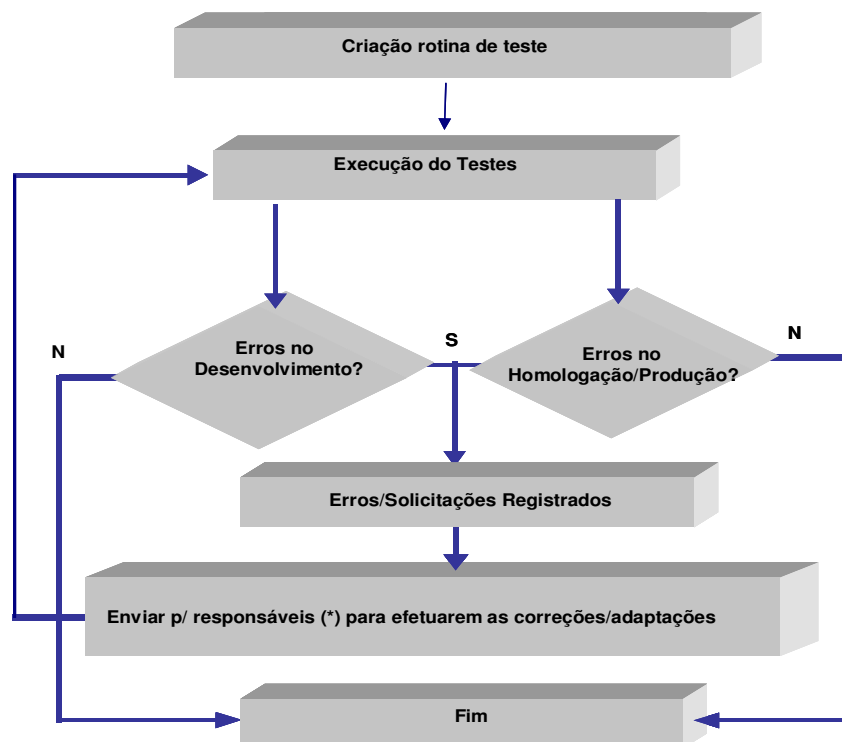


Figura 3.7 - Processo de Testes.

Na prática, os testes em fase de homologação e produção estão sujeitos a várias mudanças por parte dos usuários, tais como:

- regras de negócios;
- protótipo;
- mudança de layout;
- outras.

As solicitações de manutenção dos usuários nessa fase, quando relativas à implementação, serão entregues ao gerente de teste para posterior encaminhamento aos desenvolvedores, sendo que, quando relativas a solicitações evolutivas, serão encaminhadas ao analista de requisito e gerente de projeto, onde os pedidos serão avaliados. Se forem aceitos, esses seguem para os desenvolvedores.

Cabe lembrar que a fase de homologação também está sujeita a retestes por parte dos usuários, que os devem realizar enquanto houver erros, ou porque houve mudanças de requisitos.

Assim que os pedidos dos usuários forem atendidos e não mais apresentarem erros e mudanças, o software poderá ser encaminhado para a fase de produção.

### **3.5 – SOLICITAÇÃO DE MANUTENÇÃO**

No documento de solicitação de manutenção, os responsáveis pela realização dos testes relatam suas observações acerca da execução de funcionalidades do projeto. Nesse documento podem ser feitas mudanças dos seguintes tipos:

- **Corretiva** – quando o produto apresentado demonstra desconformidade em relação ao previsto no CDU;
- **Evolutiva/Adaptativa** – quando, após a avaliação dos usuários, estes solicitam que funcionalidades não previstas no CDU sejam adicionadas.

Na Tabela 3.2 são propostas as principais informações de um documento de solicitação, bem como suas definições:

Tabela 3.2 - Solicitação de Manutenção

Sistema	Nome do projeto / Módulo
Responsável	Nome do responsável pelo sistema (Analista de Requisito)
Solicitante	Nome de quem encontrou os erros
Manutenção	<input type="checkbox"/> Banco <input type="checkbox"/> Sistema <input type="checkbox"/> Servidor <input type="checkbox"/> Conexão
Versão do sistema	Em qual versão o sistema se encontra
CDT	Nome do CDT
Rotina do teste	Quais cenários foram testados
Contato	E-mail ou número telefônico da equipe responsável
Descrição do erro	Campo onde serão registrados todos os erros encontrados no sistema e as sugestões de melhoria para o sistema.
Manutenções	<input type="checkbox"/> Corretiva <input type="checkbox"/> Evolutiva <input type="checkbox"/> Adaptativa
Prioridade	<input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
<b>Dados da demanda</b>	
Data	Dia em que foi encontrado o erro pela equipe de testes
Hora	Hora em que foi encontrado o erro
<b>Atendimento</b>	
Data de entrega para a equipe de Desenvolvimento	O dia que em a solicitação foi repassada para os desenvolvedores fazerem as correções dos erros
Analista responsável ou Testador	Nome do responsável pela execução do teste (analista de teste ou Testador)
Data da Conclusão	O dia da conclusão da execução do teste
Hora da entrega para a equipe de Desenvolvimento	Hora em que foi entregue a solicitação para os desenvolvedores
Hora de conclusão	Hora do término do teste realizado pela equipe de teste

### **3.6 – OBJETIVO DO TESTE**

Um dos principais objetivos do teste consiste em verificar, por meio de uma execução controlada de ações, se um comportamento observado corre de acordo com o especificado. Isso quer dizer que aquele que testa busca verificar se os requisitos de um sistema foram implementados corretamente.

Entretanto, embora possa parecer um paradoxo, com a idéia apresentada acima, sob o ponto de vista de eficácia, um bom teste tem de:

1°. especialmente, encontrar erros, isto é, identificar o máximo de inconsistências e falhas nos produtos de software, para que esses possam ser corrigidos pela equipe de programadores, antes da entrega final para usuários.

2°. realizar o primeiro objetivo de forma rápida, uma vez que em aspectos de produção, o tempo é sempre um item de grande importância, assim sendo, o quanto antes o erro for encontrado, menor as chances de que os mesmos possam causar prejuízos. Assim, será mais fácil alcançar um nível alto de qualidade do produto.

## 4 – MÉTRICAS DE QUALIDADE

O objetivo deste capítulo é demonstrar a necessidade de se mensurar um produto de software. Como o principal objetivo de um sistema de gerenciamento é torná-lo o mais eficiente possível, devemos considerar a máxima estabelecida por alguns arautos da administração: “O que não pode ser medido, não pode ser gerenciado” (W. Edward Deming).

Com efeito, para se estabelecer uma abordagem que permita extrair benefícios de um sistema de gerenciamento da qualidade, primeiro é necessário conhecer o porquê medir, o que se está tentando medir, e qual é o seu tamanho.

É importante medir para que a empresa possa estabelecer melhor os relacionamentos entre seus processos, produtos e recursos. As métricas permitem que um processo ou produto do ciclo ou do desenvolvimento de um sistema informatizado seja analisado da mesma forma que um processo pode ser caracterizado pelo seu tamanho, peso ou desempenho.

O objetivo de se medir processos é obter informações para alcançar os resultados esperados. Para tanto, as medições devem ser integradas aos processos de software ao longo do seu ciclo de vida.

Assim sendo, este capítulo apresentará as bases necessárias para a criação de métricas a serem aplicadas à avaliação objetiva de algum quesito de qualidade de software. Para tanto, serão utilizados conceitos provenientes de abordagens e normas relativas à construção de métricas, para se criar um conjunto voltado para a avaliação do quesito manutenibilidade do sistema [48].

### 4.1 – ABORDAGEM GQM

O objetivo da abordagem GQM (*Goal-Question-Metric* – Objetivo-Questionamento-Métrica) [39] [40] [42] [43] é ajudar na criação de métricas a partir da definição de objetivos e questões inter-relacionadas. Ela especifica um sistema de medição, para um



conjunto de objetos, bem como de regras para a interpretação dos dados medidos. O modelo é composto dos seguintes níveis:

1. **Conceitual:** neste são definidos os objetivos para um determinado objeto, a partir de razões relacionadas ao ambiente e sob um determinado ponto de vista;
2. **Operacional:** neste, questões são utilizadas na definição de como o objetivo de medição será atingido. As questões caracterizam o objeto de medição com respeito aos aspectos de qualidade desejados e sob o ponto de vista selecionado;
3. **Quantitativo:** neste, um conjunto de métricas (Metric) é definido para cada questão proposta, visando a responder a questão de maneira quantitativa. Os dados resultantes das métricas podem ser objetivos, na medida em que dependem apenas do objeto medido, e subjetivos, quando existe a dependência do ponto de vista adotado.

O relacionamento entre os níveis conceitual, operacional e quantitativo constitui o pilar da aplicação da abordagem GQM.

O funcionamento do GQM [44] [45] sugere a realização de atividades específicas através das seguintes fases:

- a. **Planejamento** – aqui a aplicação a ser mensurada é escolhida, caracterizada e efetivada conforme o planejamento do projeto;
- b. **Definição** – onde os objetivos, questões, métricas e hipóteses devem ser definidos e documentados;
- c. **Coleta de dados** – quando os dados serão coletados para posteriormente serem utilizados para o cálculo das métricas.
- d. **Interpretação** – realiza a análise e avaliação dos resultados obtidos a partir do emprego das métricas. Com essas análises, é possível obter informações importantes acerca do produto, processo e até mesmo das métricas.

## 4.2 – NORMA ISO /IEC 9126

O padrão internacional ISO/IEC 9126 [46] descreve um modelo de qualidade para produtos de software, dividido em duas partes: a primeira parte especifica características para qualidade interna e externa; a segunda parte especifica características de qualidade para a utilização do software.

Este modelo foi proposto pelo *International Organization for Standardization (ISO)* e pelo *International Electrotechnical Commission (IEC)* e está dividido em 4 (quatro) relatórios técnicos, a saber:

1. ISO/IEC 9126-1: modelo de qualidade;
2. ISO/IEC 9126-2: métricas externas;
3. ISO/IEC 9126-3: métricas internas;
4. ISO/IEC 9126-4: métricas de qualidade no uso;

As métricas de qualidade no uso identificam se o produto atingiu as expectativas dos usuários e somente podem ser calculadas na utilização real do software.

As métricas externas podem ser obtidas apenas a partir dos estágios de teste e durante a utilização do produto e capturam seu comportamento no ambiente no qual ele irá funcionar.

Métricas internas são aplicáveis a produtos não-executáveis, desenvolvidos durante os estágios de produção do software. O objetivo destas é prever a qualidade final do produto através da quantidade de entregas executadas durante o ciclo de vida do desenvolvimento do produto.

A Norma ISO/IEC 9126-2 [47], que é voltada para a adoção de métricas externas, é estruturada com base em um conjunto definido de características. Neste trabalho, a característica escolhida foi relativa à Manutenibilidade.

### 4.3 – MANUTENIBILIDADE

Na aplicação da abordagem GQM para medição, é importante considerar que ela deve basear-se em objetivos. Então, a partir do nível conceitual, deve-se definir os objetivos respondendo a pontos específicos para esse fim. Tais pontos são:

- Propósito
- Assunto
- Objeto
- Ponto de Vista

Nesse sentido, a medição enfocou a manutenibilidade do produto [48]. Assim, o objetivo proposto para a aplicação da abordagem GQM está descrito na Tabela 4.1.

Tabela 4.1 - Descrição do Objetivo

<b>Objetivo</b>			
<b>Propósito</b>	<b>Assunto</b>	<b>Objeto</b>	<b>Ponto de Vista</b>
Avaliar	Efetivação de solicitações	Manutenção	Usuário

Assim, o objetivo é:

- Avaliar a efetivação de solicitações no processo de manutenção sob o ponto de vista do usuário.

O ponto de vista do usuário diz respeito a sua percepção, que em termos de tempo, quer em termos de tamanho do produto de software. Isso se deve ao fato de que não cabe ao usuário ponderar sobre a complexidade de uma solicitação, mas sim, como e quando ela será efetivada.

A seguir, a aplicação do modelo indica a necessidade de evoluir para o nível operacional do GQM. Nesse nível, as seguintes questões foram propostas:

- Q1 – Quão ágil é a efetivação das manutenções percebida pelo usuário?
- Q2 – Qual o nível de maturidade do produto?
- Q3 – Quanto o produto se modificou desde sua entrada em produção?

A questão Q1 procura medir a agilidade na efetivação de manutenção do produto de software, tanto para as solicitações corretivas quanto para as evolutivas. Nesse caso, o aspecto agilidade pode ser entendido como tempo para efetivação de uma solicitação de manutenção percebido pelo usuário.

A questão Q2 avalia o nível de maturidade do produto de software a partir das falhas identificadas.

A questão Q3 avalia o quanto o produto de software sofre modificações, ou seja, a frequência com que o mesmo está submetido a mudanças e adaptações desde que esse entrou na fase de produção.

Para a característica Manutenibilidade, a norma ISO/IEC 9126-2 sugere as seguintes subcaracterísticas:

- Analisabilidade
- Modificabilidade
- Estabilidade
- Testabilidade

Neste trabalho as métricas serão focadas nas características de Modificabilidade e Estabilidade.

Propõe-se a seguir uma estrutura de apresentação das métricas resultantes da análise das questões anteriormente listadas. Essa estrutura será organizada da seguinte forma:

- Questão – diz respeito ao questionamento que a métrica deve responder.
- Métrica – contém a sua definição propriamente dita é representada por uma equação.
- Método de coleta – estabelece as informações necessárias que devem ser obtidas para o cálculo da métrica.
- Sentido – avalia se a qualidade do quesito é diretamente (sentido para cima) ou inversamente (sentido para baixo) proporcional ao valor da métrica.

#### 4.4 – MÉTRICAS

Questão – Quão ágil é a efetivação das manutenções percebida pelo usuário?

Métrica 1 – Tempo médio para efetivar uma manutenção corretiva como mostra a Eq. (4.1)

$$M1 = \frac{\sum_{n=1}^C T_n}{C} \quad (4.1)$$

Onde:

$T_n$  = Tempo necessário para efetivar cada uma das  $n$  solicitações de manutenção corretiva.

$C$  = Número total de solicitações de manutenção corretiva.

Método de Coleta

1. Registrar data e horário no qual uma determinada solicitação de manutenção foi feita pelo usuário.
2. Registrar data e horário no qual uma determinada solicitação de manutenção foi efetivada e passada para o ambiente de produção.
3. Contar o número de solicitações de manutenção.

#### Sentido de M1

A métrica diz respeito a tempo médio para efetivação de uma solicitação de manutenção corretiva, ao mesmo tempo, sob o ponto de vista do usuário, tem-se que esse tempo corresponde ao tempo que o usuário espera para ter essa solicitação efetivada. Isso implica dizer que quanto menos tempo o usuário esperar, maior será sua percepção de qualidade. Assim sendo, o sentido dessa métrica evolui para baixo, ou seja, quanto menor o valor da métrica, melhor.

Questão – Quão ágil é a efetivação das manutenções percebida pelo usuário?

Métrica 2 – Tempo médio para efetivar uma solicitação de manutenção evolutiva como mostra a Eq. (4.2).

$$M2 = \frac{\sum_{n=1}^E T_n}{E} \quad (4.2)$$

Onde:

$T_n$  = Tempo necessário para efetivar cada uma das  $n$  manutenções evolutivas.

$E$  = Número total de manutenções evolutivas.

Método de Coleta

1. Registrar data e horário no qual uma determinada manutenção evolutiva feita pelo usuário.
2. Registrar data e horário no qual uma determinada manutenção evolutiva efetivada e passada para o ambiente de produção.
3. Contar o número de solicitações de manutenção evolutiva.

### Sentido de M2

A análise é similar à feita para a métrica M1. Nesse caso, a métrica diz respeito a tempo médio para efetivação de uma solicitação de manutenção evolutiva, ao mesmo tempo, sob o ponto de vista do usuário, tem-se que esse tempo corresponde ao tempo que o usuário espera para ter essa solicitação efetivada. Isso implica dizer que quanto menos tempo o usuário esperar, maior será sua percepção de qualidade. Assim sendo, o sentido dessa métrica evolui para baixo, ou seja, quanto menor o valor da métrica, melhor.

Questão – Qual o nível de maturidade do produto?

Métrica 3 – Tempo Médio entre ocorrência de Falhas como mostra a Eq. (4.3).

$$M3 = \frac{\sum_{n=1}^{F-1} TF_n}{F-1} \quad (4.3)$$

Onde:

$TF_n$  = Tempo entre duas falhas consecutivas

F = Número total de falhas.

Para o cálculo dessa métrica, as falhas serão consideradas como causadoras de manutenções corretivas.

Método de Coleta

1. Registrar data e horário no qual uma determinada falha ou erro ocorreu.
2. Numerar seqüencialmente as falhas ocorridas.
3. Contar o número de solicitações de correção.

### **Sentido de M3**

A métrica diz respeito a tempo médio entre ocorrência de falhas, o quê, sob o ponto de vista do usuário, corresponde ao período no qual o sistema não apresentou falhas, ou seja, funcionou de forma satisfatória, considerando apenas o quesito falhas, sem se adentrar em necessidades de manutenções evolutivas, aspectos de usabilidade, etc. Isso implica dizer que quanto mais tempo o usuário ficar sem identificar uma falha, maior será sua percepção de qualidade. Assim sendo, o sentido dessa métrica evolui para cima, ou seja, quanto maior o valor da métrica, melhor.

Questão – Qual o nível de maturidade do produto?

Métrica 4 – Taxa de falhas por Ponto de Função como mostra a Eq. (4.4)

$$M4 = \frac{F}{L} \quad (4.4)$$

Onde:

F= Número total de falhas

L= Tamanho do produto de software, em Pontos de Função, contados pelo método APF.

Para o cálculo dessa métrica, as falhas serão consideradas como causadoras de manutenções corretivas.

#### Método de Coleta

1. Registrar o número de falhas.
2. Medir o tamanho do produto de software em Pontos de Função.

#### Sentido de M4

A métrica avalia a maturidade do produto a partir da relação existente entre o número de falhas e o seu tamanho (medido em Pontos de Função). Nesse caso, a percepção do usuário diz respeito à noção de proporcionalidade dada pela métrica, ou seja, supondo dois sistemas hipotéticos, para um determinado número de falhas identificadas, o usuário avaliaria melhor aquele sistema de maior tamanho, pois, embora a quantidade de falhas fosse a mesma, o sistema menor deveria apresentar um comportamento melhor. Isso implica dizer que quanto menor a relação entre falhas e tamanho do sistema, maior será sua percepção de qualidade. Assim sendo, o sentido dessa métrica evolui para baixo, ou seja, quanto menor o valor da métrica, melhor.

Questão – Quanto o produto se modificou desde sua entrada em produção?

Métrica 5 – Taxa de manutenções evolutivas por Ponto de Função como mostra a Eq. (4.5)

$$M5 = \frac{N}{L} \quad (4.5)$$

Onde:

N= Número total de manutenções evolutivas.

L= Tamanho do produto de software, em Pontos de Função, contados pelo método APF.

#### Método de Coleta

1. Registrar o número de manutenções evolutivas.
2. Medir o tamanho do produto de software em Pontos de Função.

#### Sentido de M5

A métrica avalia a maturidade do produto a partir da relação existente entre o número de manutenções evolutivas e o seu tamanho (medido em Pontos de Função). Nesse caso, a percepção do usuário diz respeito à noção de proporcionalidade dada pela métrica, ou seja,



supondo dois sistemas hipotéticos, para um determinado número de solicitações evolutivas identificadas, o usuário avaliaria como mais estável aquele sistema de maior tamanho, pois, embora a quantidade solicitações evolutivas fosse a mesma, o sistema menor deveria apresentar um comportamento mais estável. Isso implica dizer que quanto menor a relação entre solicitações evolutivas e tamanho do sistema, maior será sua percepção de qualidade. Assim sendo, o sentido dessa métrica evolui para baixo, ou seja, quanto menor o valor da métrica, melhor.

#### 4.5 – CONSOLIDAÇÃO

A Tabela 4.2 a seguir apresenta uma consolidação sobre o sentido da evolução de cada uma das métricas abordadas neste capítulo.

Tabela 4.2 – Resumo do sentido das métricas

Métrica	Sentido
M1	↓
M2	↓
M3	↑
M4	↓
M5	↓

## **5 – APLICAÇÃO E RESULTADOS**

Este capítulo apresenta o resultado da aplicação do modelo de gestão de qualidade descrito neste trabalho, ou seja, o modelo baseado na adoção do RUP, associado à aplicação de testes manuais na fase de desenvolvimento buscando identificar o maior número de erros. E, dando continuidade à aplicação do modelo proposto, durante a fase de produção, avaliar o produto de software sob a ótica de manutenibilidade, por meio da aplicação de métricas.

Esse processo será aplicado durante o desenvolvimento de um produto de software e seus resultados serão comparados com outro produto que cujo modelo proposto neste trabalho não foi empregado.

Para efeito de comparação na fase de produção, será aplicado o conjunto de métricas de manutenibilidade proposto no capítulo 4 deste trabalho.

### **5.1 – DESCRIÇÃO DO PRODUTO *MGP***

O produto que será objeto da aplicação dos temas abordados neste trabalho é o Módulo de Gestão de Planejamento (MGP). O MGP foi desenvolvido para órgão governamental de alcance nacional e com necessidades específicas para subsidiar seu processo de planejamento de ações estratégicas. O seu objetivo é facilitar a criação de planos de ação das diversas unidades que compõem o órgão, bem como agrupá-los, verificar planejamento estratégico, acompanhar o desempenho de atividades, etc.

Durante o processo de desenvolvimento desse produto foram envolvidas várias equipes de profissionais cujos perfis variavam entre arquitetos, analistas, desenvolvedores e testadores. Esses profissionais foram agrupados nas seguintes equipes:

- Gerência do Projeto
- Equipe de Análise
- Equipe de Desenvolvimento
- Equipe de Teste

A distribuição dos papéis a cada uma das equipes, bem como o número de profissionais envolvidos são ilustrados na Figura 5.1

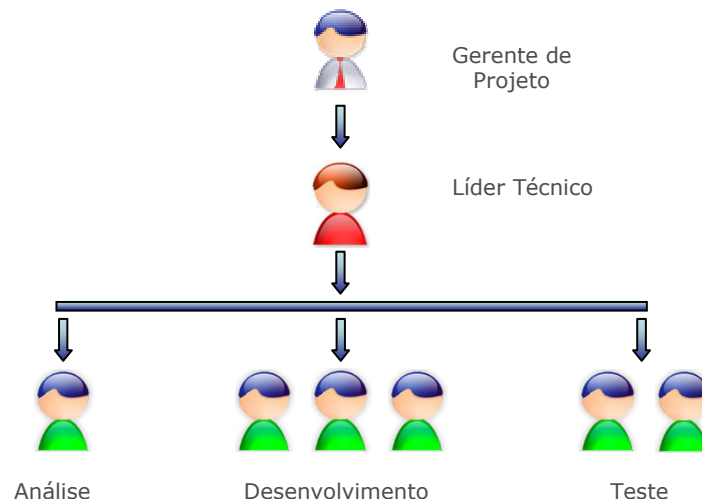


Figura 5.1 – Papéis das equipes do MGP.

Inicialmente, dois profissionais da equipe de desenvolvimento foram responsáveis pela manutenção do sistema de forma não exclusiva, ou seja, trabalhavam também na manutenção de outros sistemas. O desenvolvimento do MGP foi estruturado com base nas disciplinas do RUP [9]. A equipe de manutenção do órgão é responsável por uma série de outros produtos além do MGP.

Ao todo, o projeto consumiu aproximadamente 9 (nove) meses entre o levantamento de requisitos e a passagem do produto para produção.

O processo de desenvolvimento do produto pautou-se nas seguintes etapas:

1. Desenvolvimento do plano de projeto, modelagem de negócios, levantamentos de requisitos e a prototipação;
2. Definição dos CDU, especificação dos mesmos, geração de diagrama de classe e a modelagem de dados;
3. Implementação;
4. Planejamento dos CDT;

5. Execução dos testes;
6. Testes internos;
7. Testes de homologação pelos usuários;

Durante a fase de desenvolvimento e homologação, os CDU são testados por completo. Só então são feitas as solicitações de manutenção corretivas, evolutivas e adaptativas. A cada versão gerada, são realizados retestes para a verificação/constatação da implementação das solicitações feitas e identificação de possíveis novos erros ou inconsistências. Nesse processo, não foi definido um limiar para a identificação de uma quantidade razoável de erros, pois cada CDU é testado integralmente e os erros encontrados nessa fase são todos encaminhados para correção.

A utilização do produto se dá de forma contínua, ou seja, não são previstos picos de utilização, e seu acesso é voltado para número restrito de usuários.

## **5.2 – DESCRIÇÃO SUCINTA DO PROCESSO**

O levantamento de requisitos foi feito por meio de reuniões com usuários para levantamento da documentação utilizada no processo. A princípio foi feito um levantamento macro, visando a um entendimento do negócio e definição de escopo. Posteriormente foram levantadas todas as atividades executadas pela área e a forma de sua execução. Deste levantamento foi possível definir as funcionalidades a serem contempladas pelo sistema em desenvolvimento e as regras de negócio que deverão ser respeitadas. Um instrumento auxiliar utilizado no levantamento foi a prototipação, que permitiu a visualização do sistema que estava sendo modelado. Uma vez validado o protótipo, foram especificados os CDU submetidos também à validação dos usuários.

Assim, apresentamos a seguir as etapas deste processo.

### **1. Levantamento de Requisitos**

- Cadastramento de órgãos governamentais;
- Cadastramento de planos de ação das diversas unidades que compõem o órgão;

- Agrupamento de ações;
- Verificação do planejamento estratégico;
- Inclusão de indicadores de desempenho;
- Acompanhamento do desempenho de atividades;
- Apuração de prováveis desvios.

## **2. Entendimento do negócio**

O MPG é um sistema que visa automatizar o processo de Planejamento Estratégico de órgãos governamentais.

## **3. Escopo do projeto**

O escopo do módulo de Planejamento é permitir a realização e o acompanhamento do planejamento estratégico de órgãos governamentais.

## **4. Elaboração do documento de visão**

Este documento, que tem como objetivo apresentar uma visão macro do negócio, contém os seguintes tópicos:

- Objetivo;
- Situação atual do processo a ser automatizado;
- Contexto do negócio;
- Escopo do projeto;
- Objetivos do projeto;
- Premissas;
- Restrições;
- Visão geral do sistema a ser desenvolvido;
- Atores envolvidos no processo;

- Requisitos funcionais;
- Requisitos não funcionais.

## **5. Elaboração de protótipo e Validação de protótipo**

Visando a facilitar o levantamento de requisitos (apresentação dos fluxos envolvidos no processo de Planejamento Estratégico) foi elaborado protótipo contendo todas as telas do futuro sistema com seus respectivos campos. Simultaneamente à construção do protótipo, foram levantadas regras de negócio. O protótipo foi submetido à aprovação do usuário. Após ajustes, o mesmo foi validado.

## **6. Elaboração e Validação de CDU**

Após validação do protótipo, foram especificados os CDU (Anexo A), sendo um para cada funcionalidade do sistema. Eles possuem: descrição da funcionalidade, objetivo, forma de acesso à funcionalidade, modelo das telas envolvidas no processo, descrição do conteúdo dos campos da tela e procedimento do sistema para cada ação do ator.

## **7. Modelagem de banco de dados**

Baseando-se no protótipo e nas regras de negócio levantadas com usuários e documentadas nos CDU, foi efetuada a modelagem do banco de dados. Nessa modelagem foram identificados as entidades relevantes ao negócio e os relacionamentos entre elas.

## **8. Implementação**

A implementação foi feita em linguagem JAVA e SGBD Oracle, seguindo o padrão de implementação com componentes pré-definidos, com base nos CDU, no protótipo e no modelo de dados.

## **9. Elaboração dos CDT, execução dos testes e validação do Sistema**

Após a validação dos CDU, foram especificados os CDT, contendo a descrição da funcionalidade, baseando-se nos CDU. Após a implementação dos CDU, esses foram liberados para que a equipe de testes executasse os testes no sistema. Depois da aplicação dos testes e correção de erros de implementação, foram liberados para provação e homologação do usuário. A aprovação desse projeto é realizada por uma bateria de testes sobre o produto para assegurar que o mesmo esteja em conformidade com as necessidades e requisitos definidos pelo usuário.

## **10. Pontos de Função**

O levantamento da quantidade dos Pontos de Função do MGP para fins de avaliação do tamanho do sistema levou em consideração a quantidade de funções transacionais e de dados. Considerando os componentes ALI, AIE, CE, EE e SE, o levantamento para o MGP retornou a quantidade de 687 pontos de função, de forma não ajustada.

## **11. Descrição sucinta do MGP**

- CDU: para o desenvolvimento do produto foi necessário construir um sistema com 30 CDU.
- CDT: 30 (a construção de CDT baseia-se na construção de 1 CDT para cada CDU).
- Pontos de Função: o levantamento sem os critérios de ajustes forneceu a informação de que o sistema apresenta 687 pontos de função.
- Tempo para Implementação: 9 meses.
- Tempo para testes internos: 11 semanas.
- Tempo para homologar o sistema com usuários: 4 meses.

## **5.3 – DESCRIÇÃO DO PRODUTO SGQ**

Para efeito de comparação foi escolhido o Sistema de Gestão de Qualidade (SGQ). Este consiste em uma aplicação de software voltada ao gerenciamento de indicadores de qualidade de prestação de serviço. O cliente é uma entidade governamental. O objetivo do produto é constituir um portal para a prestação de informação por parte de empresas privadas, bem como facilitar a visualização e consolidação das informações prestadas.

Durante o processo de desenvolvimento desse produto foram envolvidas várias equipes de profissionais cujos perfis variavam entre arquitetos, analistas, desenvolvedores e testadores.

## Equipes:

- Gerência do Projeto
- Equipe de Análise
- Equipe de Implementação

A distribuição dos papéis a cada uma das equipes, bem como o número de profissionais envolvidos é ilustrado na Figura 5.2

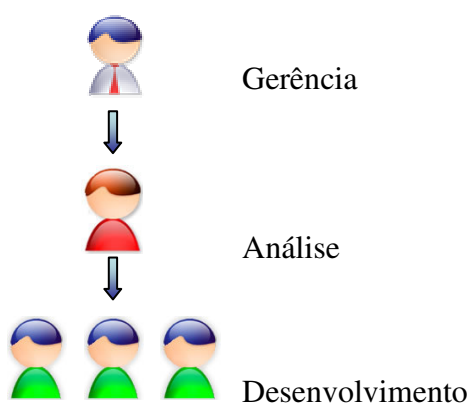


Figura 5.2 – Papéis das equipes do Produto SGQ.

O Analista responsável também desempenhou o papel de desenvolvedor. A utilização do produto é feita de forma contínua, ou seja, o SGQ foi desenvolvido para utilização interna e externa à entidade. A utilização interna é feita continuamente, não sendo fácil a identificação de picos de uso. Já a utilização externa pode ser dividida na parte de entrada de dados, com picos definidos, e consulta de relatórios, feita de forma mais contínua. O acesso é voltado para número restrito de usuários com exceção da consulta a relatórios, que é aberta ao público.

### 1. Fluxo de processos

O cliente apresenta uma demanda para a área de tecnologia da informação (TI). A área designa um gerente de projeto e uma equipe para implementar o produto. A equipe de implementação levanta os requisitos com o cliente. A princípio é feito um levantamento macro, visando a um entendimento do negócio e definição de escopo. Posteriormente, a equipe verifica a existência ou não de sistemas que possam atender à necessidade do cliente, sem ser necessária a criação de mais um sistema. Caso seja necessário, a equipe de



implementação desenvolve os CDU (Anexo B) e apresenta um protótipo de produto a ser validado pelo cliente. Se for validado, esse produto passa para a fase de produção.

## 2. Pontos de Função

O levantamento da quantidade dos Pontos de Função do SGQ para fins de avaliação do tamanho do sistema levou em consideração a quantidade de funções transacionais e de dados. Considerando os componentes ALI, AIE, CE, EE e SE, o levantamento para o SGQ retornou a quantidade de 844 pontos de função. Similarmente ao MGP, a contagem não considerou nenhum fator de ajuste.

## 5.4 – COMPARAÇÃO ENTRE OS PRODUTOS MGP E SGQ

O processo empregado para produzir o MGP baseou-se no Processo Unificado, mais propriamente o RUP. A empresa responsável possui certificação CMMi nível (2) dois (CMMi-2). Esse processo de gerenciamento foi desenhado, bem como suas atribuições foram definidas de modo a contemplar todas as fases do RUP, assim como as disciplinas definidas no Processo Unificado. Nesse processo, a disciplina de teste está alocada na Coordenação de Qualidade.

O processo utilizado pelo SGQ não se baseou em um processo unificado. A empresa responsável pelo sistema buscava obter certificação CMM nível dois (CMM-2). Esse processo de gerenciamento foi desenhado de modo que as atividades fiquem concentradas em um número pequeno de profissionais. Nesse processo, o teste é alocado na própria equipe de implementação. A comparação entre os sistemas pode ser vista na Tabela 5.1.

Tabela 5.1 – Tabela comparativa MGP x SGQ

Sistema	MGP	SGQ
Nome	Módulo de Gestão de Planejamento	Sistema de Gestão de Qualidade
Processo	Unificado (RUP)	-
Modelo de Qualidade da Empresa	CMMi-2	CMM-2
Profissionais	8	4
CDU	30	54
CDT	30	-

<b>Pontos de Função</b>	687	844
<b>Implementação</b>	38 semanas	26 semanas
<b>Teste Interno</b>	11 semanas	-
<b>Teste de Homologação</b>	4 semanas	-
<b>Builds</b>	4	3
<b>Usuários</b>	69	192

O processo unificado utilizado pelo sistema MGP prevê uma disciplina específica para a atividade de testes. O emprego do processo de teste consome em torno de 28% do tempo necessário para a implementação do produto, conforme pode ser visto na Figura 5.3.

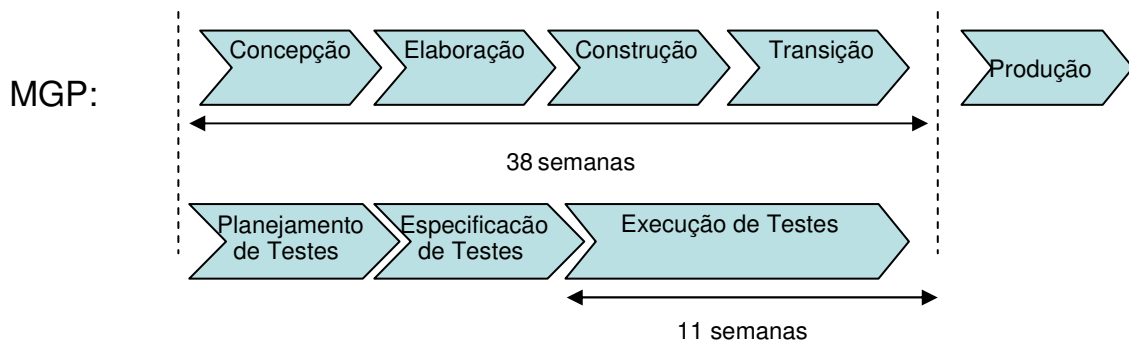


Figura 5.3 – Tempo gasto para Testes no MGP.

O processo utilizado pelo sistema SGQ não prevê uma disciplina específica para os testes. Esses são realizados pela própria equipe de análise e desenvolvimento e, por conseguinte, fazem parte da implementação do sistema. Não existe uma definição clara e específica para a elaboração de teste tampouco para os registros dos erros e falhas encontrados. A manifestação de registro para eventuais erros acontece na fase de homologação, que é feita por parte dos usuários. Mesmo assim, não há uma rotina definida de quais passos devem ser seguidos.

#### **5.4.1 – Fase de Desenvolvimento onde foram realizados os testes**

A escolha do tipo de teste a ser aplicado ao MGP pautou-se nos seguintes aspectos:

- Não identificação de picos de utilização do sistema, na medida em que o perfil de utilização do MGP não identificava períodos de maior uso. Isso justificou o não emprego dos testes de volume e carga.
- O MGP foi concebido para fazer integrar um rol de outros sistemas dentro de um ambiente seguro, ou seja, todos os acessos devem passar por cadastramento, poucos profissionais tem acesso ao código, o ambiente é protegido por firewall e outros quesitos de segurança. Nesse sentido os testes de segurança também não foram empregados.
- O acesso ao sistema foi dado a um número limitado e relativamente pequeno de usuários, o quê fez justificou a não aplicação dos testes de segurança, volume e carga.
- Devido aos aspectos funcionais definidos durante as fases iniciais do projeto, a aplicação de testes funcionais foi definida como obrigatória.

Assim sendo, os seguintes testes foram contemplados na especificação dos CDT:

- Integração;
- Sistema;
- Aceitação;
- Acessibilidade;
- Usabilidade;
- Integridade do Banco de Dados;
- Regressão;
- Configuração e;
- Instalação.

O processo de teste foi descrito no Capítulo 3. Os dados obtidos consideram os erros encontrados na fase de desenvolvimento, e foram classificados em razão de sua criticidade e complexidade.

A criticidade do erro é classificada nos seguintes níveis:

- **Simple** – Os erros simples são constituídos como sendo erros que não impedem o bom funcionamento da aplicação, ou seja, ele não impede que se execute os fluxos de eventos do sistema. Exemplos desses erros: Mensagens que não estão de acordo com as especificações dos CDU, sendo que esses erros podem ser corrigidos paralelo aos testes da aplicação.
- **Críticos** – Como o próprio nome já diz, são erros que impedem o funcionamento da aplicação, devendo ser corrigidos com prioridade máxima, pois impossibilitam que os testes sejam realizados com rapidez. Exemplo: página fora do ar, falta de comunicação com o Banco de Dados, etc.

A complexidade do tratamento do erro é classificada em razão da complexidade do próprio CDU, ou seja:

- **Baixa:** acesso a poucos arquivos lógicos, poucos atributos nas tabelas e telas.
- **Média:** acesso à quantidade maior de arquivos lógicos, quantidade maior de atributos nas tabelas e telas.
- **Alta:** quantidade grande de arquivos lógicos acessados, telas com grande quantidade de campos, realização de operações complexas para exibição/gravação de dados.

Para o MGP foram encontrados 101 erros nos 30 Casos de Teste (Anexo C) e 3 check list utilizados durante a fase de desenvolvimento. A cada erro (ou conjunto de erros) encontrado e corrigido, era feito um teste de regressão total no sistema. Os erros são listados na Tabela 5.2.

Tabela 5.2 – Erros encontrados no Desenvolvimento do MGP

<b>Número</b>	<b>Erro</b>	<b>Severidade</b>	<b>Prioridade</b>
1	Implementação	Críticos	Média
2	Implementação	Críticos	Média
3	Implementação	Críticos	Média
4	Implementação	Críticos	Média
5	Implementação	Críticos	Média
6	Implementação	Críticos	Média
7	Implementação	Críticos	Alta
8	Implementação	Críticos	Alta
9	Implementação	Simples	Alta
10	Implementação	Simples	Alta
11	Implementação	Simples	Alta
12	Implementação	Simples	Média
13	Implementação	Simples	Média
14	Implementação	Críticos	Alta
15	Implementação	Críticos	Alta
16	Implementação	Críticos	Alta
17	Implementação	Críticos	Alta
18	Implementação	Críticos	Alta
19	Implementação	Críticos	Média
20	Implementação	Críticos	Média
21	Implementação	Críticos	Média
22	Implementação	Críticos	Baixa
23	Implementação	Críticos	Baixa
24	Implementação	Críticos	Baixa
25	Implementação	Críticos	Baixa
26	Implementação	Críticos	Baixa
27	Implementação	Críticos	Média
28	Implementação	Críticos	Média
29	Implementação	Críticos	Média
30	Implementação	Simples	Média
31	Implementação	Simples	Baixa
32	Implementação	Críticos	Baixa
33	Implementação	Críticos	Baixa
34	Implementação	Críticos	Baixa
35	Implementação	Críticos	Baixa
36	Implementação	Simples	Baixa
37	Implementação	Simples	Média
38	Implementação	Simples	Média
39	Implementação	Simples	Média
40	Implementação	Críticos	Média
41	Implementação	Críticos	Média
42	Implementação	Críticos	Média
43	Implementação	Críticos	Média
44	Implementação	Críticos	Média
45	Implementação	Críticos	Média
46	Implementação	Críticos	Média

47	Implementação	Críticos	Média
48	Implementação	Críticos	Média
49	Implementação	Críticos	Média
50	Implementação	Críticos	Média
51	Implementação	Críticos	Média
52	Implementação	Críticos	Média
53	Implementação	simples	Baixa
54	Implementação	simples	Alta
55	Implementação	simples	Alta
56	Implementação	simples	Alta
57	Implementação	simples	Alta
58	Implementação	simples	Alta
59	Implementação	simples	Alta
60	Implementação	simples	Alta
61	Implementação	simples	Alta
62	Implementação	Críticos	Alta
63	Implementação	Críticos	Alta
64	Implementação	Críticos	Alta
65	Implementação	Críticos	Média
66	Implementação	Críticos	Média
67	Implementação	Críticos	Média
68	Implementação	Críticos	Média
69	Implementação	Críticos	Média
70	Implementação	Críticos	Média
71	Implementação	Críticos	Média
72	Implementação	Críticos	Média
73	Implementação	Críticos	Média
74	Implementação	Críticos	Média
75	Implementação	Críticos	Média
76	Implementação	Críticos	Média
77	Implementação	Críticos	Alta
78	Implementação	Críticos	Alta
79	Implementação	Críticos	Alta
80	Implementação	Críticos	Alta
81	Implementação	Críticos	Alta
82	Implementação	Críticos	Alta
83	Implementação	Críticos	Alta
84	Implementação	Críticos	Alta
85	Implementação	Críticos	Alta
86	Implementação	Críticos	Alta
87	Implementação	Críticos	Alta
88	Implementação	Críticos	Alta
89	Layout	Críticos	Média
90	Layout	Críticos	Baixa
91	Layout	Simple	Baixa
92	Layout	Simple	Baixa
93	Layout	Simple	Média
94	Layout	Simple	Média

95	Regra de Negócio	Críticos	Média
96	Regra de Negócio	Críticos	Média
97	Regra de Negócio	Simple	Alta
98	Regra de Negócio	Críticos	Alta
99	Regra de Negócio	Críticos	Baixa
100	Regra de Negócio	Simple	Média
101	Regra de Negócio	Críticos	Baixa

A classificação quanto a Severidade e a Origem dos erros podem ser vistas nas Figura 5.4 e Figura 5.5, respectivamente.

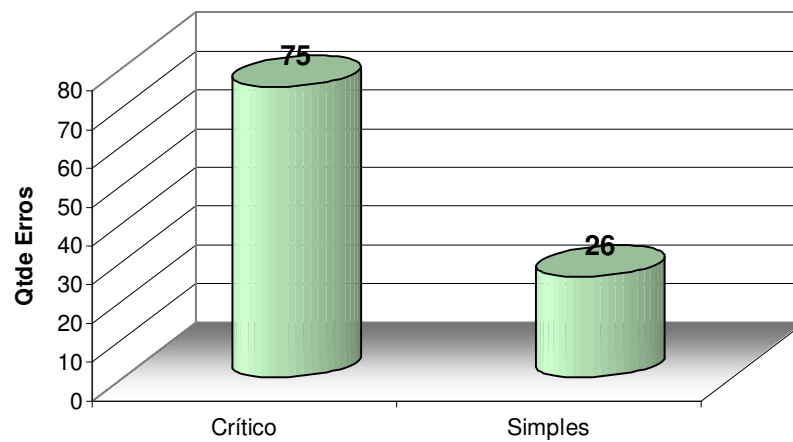


Figura 5.4 – Criticidade.

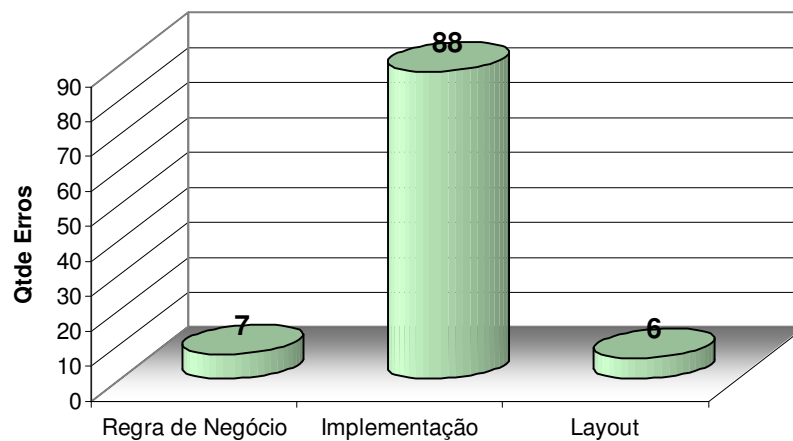


Figura 5.5 – Origem dos erros.

Como os erros do SGQ não são registrados, não foi possível obter os dados relativos ao número de erros identificados pela equipe de implementação.

#### **5.4.2 – Análise Comparativa**

A comparação, para ser eficiente, deve observar que aquilo que se vai comparar deve possuir características iguais ou, pelo menos, normalizáveis. Entretanto, tanto o MGP quanto o SGQ possuem processos, equipes e finalidades diferentes, o que torna difícil, a princípio, o estabelecimento de um critério para comparação.

Todavia, pode-se ter uma noção de avaliação se a comparação entre os processos utilizados pelos dois sistemas se basear nos seguintes aspectos:

- Tamanho do Sistema;
- Tempo de implementação; e
- Recursos utilizados.

O tamanho do sistema foi avaliado pelo número de pontos de função que cada sistema possui. Nesse sentido, pode-se perceber que o SGQ (844 pontos de função) é maior que o MGP (657 pontos de função).

O tempo para implementação considerou o intervalo compreendido entre a solicitação do sistema feita pelo cliente até a sua efetiva entrega, ou seja, desde o levantamento de requisitos para o futuro sistema até a finalização da fase de homologação. Para o SGQ foram consumidas 26 semanas enquanto que para o MGP foram 38 semanas.

No aspecto: “recursos utilizados”, foi considerado o número de profissionais envolvidos no processo. A avaliação não se deteve em ponderar os diversos papéis envolvidos, e sim apenas fazer uma avaliação acerca do número absoluto de profissionais envolvidos no desenvolvimento.

Considerando esses dois aspectos, pode-se verificar que o processo utilizado para a produção do SGQ foi mais eficiente tanto em termos de tempo de implementação quanto em recursos envolvidos, pois, mesmo o SGQ sendo maior, sua implementação necessitou de menos tempo e menos recursos que a implementação do MGP.



### 5.4.3 – Fase de Produção - Coleta de Dados

Ambos os sistemas foram avaliados durante o período de seis meses. Esse período foi considerado como suficiente para possibilitar que erros ou falhas dos sistemas pudessem ser detectados durante a fase de produção. Considerou-se que a maioria dos erros detectados, devido ao número de usuários utilizando os sistemas diariamente e na medida em que todas as funcionalidades seriam testadas nas mais diversas condições de modo a atender a estes usuários.

Em virtude do SGQ não possuir um processo para o acompanhamento de solicitações, foi definido que os processos de ambos, MGP e SGQ, teriam as suas solicitações feitas em registro próprio. O MGP utilizou a ferramenta SCARAB para registro e o SGQ utilizou uma planilha para registro. Ambos os registros tinha os seguintes campos de identificação das solicitações de manutenção:

- a. data e horário de solicitação;
- b. data e horário de efetivação;
- c. tipo de solicitação(corretiva ou evolutiva).
- d. complexidade (alta, média ou baixa)

Ademais, foi definido que a equipe de manutenção responsável pelo sistema específico, faria a classificação de complexidade da solicitação, observando os critérios definidos na fase de desenvolvimento, que são classificados com base no número de tabelas e elementos relacionados ao Caso de Uso para o qual foi feita a solicitação de manutenção.

Os dados relativos às métricas descritas no capítulo 4 foram coletados e agrupados em períodos mensais para possibilitar uma análise evolutiva da estabilidade do projeto no decorrer do tempo, sendo:

- M1 - Tempo médio para efetivar uma solicitação de manutenção corretiva
- M2 - Tempo médio para efetivar uma solicitação de manutenção evolutiva
- M3 - Tempo médio entre ocorrência de Falhas
- M4 - Taxa de Falhas por Ponto de Função
- M5 - Taxa de manutenções evolutivas por Ponto de Função

Durante o período de estudo, puderam ser obtidos os dados para a aplicação em métricas, conforme mostra a Tabela 5.3 e Tabela 5.4. Foram coletados o tempo gasto nas manutenções e o nível de complexidade das mesmas. Exemplos de solicitações corretiva (C) e evolutiva (E) para o MGP podem ser vistas nos Anexos D e E, respectivamente. Para o SGQ, um exemplo de solicitação corretiva (C) pode ser vista no anexo F.

Tabela 5.3 – Solicitações de manutenção do MGP

MGP					
Mês	Tipo	Data Solicitação	Data Efetivação	Tempo(h)	Complexidade
mês 1	E	5/3/07 18:26	8/5/07 11:19	1529	Alta
mês 2	E	3/4/07 8:26	9/4/07 16:19	152	Baixa
mês 2	E	10/4/07 13:13	12/4/07 11:21	46	Média
mês 2	E	11/4/07 8:26	1/5/07 16:19	488	Média
mês 2	E	16/4/07 8:26	28/4/07 16:19	296	Média
mês 2	E	16/4/07 17:26	28/4/07 15:19	286	Média
mês 2	E	16/4/07 17:45	28/4/07 16:48	287	Média
mês 2	E	16/4/07 18:58	28/4/07 16:49	286	Média
mês 2	E	16/4/07 19:26	28/4/07 12:19	281	Média
mês 2	E	21/4/07 19:21	30/4/07 11:25	208	Média
mês 2	E	22/4/07 8:26	26/4/07 16:19	104	Média
mês 2	C	24/4/07 16:23	29/4/07 14:21	118	Média
mês 2	E	30/4/07 13:41	1/5/07 17:27	28	Baixa
mês 3	E	18/5/07 8:26	23/5/07 16:19	128	Baixa
mês 4	C	1/6/07 19:21	29/6/07 11:25	664	Alta
mês 5	E	3/7/07 8:26	9/7/07 16:19	152	Baixa
mês 5	C	19/7/07 10:00	20/7/07 19:00	33	Baixa

Tabela 5.4 – Solicitação de manutenções do SGQ

SGQ					
Mês	Tipo	Data Solicitação	Data Efetivação	Tempo(h)	Complexidade
mês 1	C	07/01/07 08:43	07/01/07 17:54	9	Baixa
mês 1	E	10/01/07 15:49	12/01/07 18:02	51	Média
mês 1	E	12/01/07 09:09	12/01/07 09:59	0	Baixa
mês 1	E	20/01/07 09:55	21/01/07 15:58	30	Baixa
mês 1	C	31/01/07 10:18	06/04/07 10:48	1560	Alta
mês 1	E	31/01/07 10:18	26/03/07 11:36	1297	Média
mês 1	E	31/01/07 15:36	09/04/07 15:58	1632	Média
mês 2	E	02/02/07 16:41	01/03/07 17:40	649	Alta
mês 2	E	07/02/07 11:12	25/03/07 17:36	1110	Alta
mês 2	C	10/02/07 16:27	21/03/07 15:13	935	Alta
mês 2	C	22/02/07 11:42	07/05/07 17:51	1782	Alta
mês 2	C	28/02/07 11:14	26/04/07 20:05	1377	Alta
mês 3	C	01/03/07 15:22	01/03/07 16:18	1	Baixa
mês 3	C	05/03/07 15:35	26/03/07 11:46	500	Média

mês 3	C	06/03/07 08:53	19/03/07 17:47	321	Média
mês 3	C	08/03/07 18:34	19/03/07 11:20	257	Média
mês 3	C	13/03/07 10:20	19/03/07 17:13	151	Média
mês 3	E	14/03/07 08:57	04/04/07 09:53	505	Média
mês 3	E	14/03/07 09:41	25/03/07 09:41	264	Baixa
mês 3	E	14/03/07 15:09	20/03/07 10:34	139	Baixa
mês 3	C	14/03/07 15:43	19/03/07 16:41	121	Média
mês 3	C	14/03/07 18:03	04/04/07 09:53	495	Média
mês 3	C	18/03/07 12:23	28/03/07 09:00	237	Média
mês 3	E	18/03/07 16:22	20/03/07 10:33	42	Baixa
mês 3	C	19/03/07 09:21	19/03/07 16:26	7	Baixa
mês 3	E	19/03/07 09:41	19/03/07 12:18	3	Baixa
mês 3	C	19/03/07 11:01	07/05/07 17:50	1182	Alta
mês 3	C	20/03/07 10:17	25/03/07 09:44	119	Média
mês 3	E	20/03/07 10:19	25/03/07 16:44	126	Baixa
mês 3	C	20/03/07 11:57	21/03/07 09:21	22	Média
mês 3	C	20/03/07 16:31	25/03/07 09:44	113	Média
mês 3	C	22/03/07 09:04	23/04/07 15:51	774	Média
mês 3	C	22/03/07 09:55	07/05/07 17:51	1112	Média
mês 3	C	22/03/07 10:03	25/03/07 11:46	73	Média
mês 3	C	25/03/07 11:02	26/03/07 11:37	24	Baixa
mês 3	C	26/03/07 11:56	04/04/07 09:07	214	Média
mês 3	E	26/03/07 15:07	26/03/07 17:08	2	Baixa
mês 3	C	27/03/07 17:07	29/03/07 10:11	41	Média
mês 3	C	27/03/07 17:07	02/04/07 14:33	141	Média
mês 3	E	28/03/07 09:34	06/04/07 11:37	218	Baixa
mês 3	C	28/03/07 14:11	06/04/07 10:45	212	Média
mês 3	C	28/03/07 14:26	04/04/07 10:12	164	Média
mês 3	C	28/03/07 16:17	02/04/07 10:36	114	Média
mês 3	C	28/03/07 17:07	30/03/07 10:11	41	Média
mês 3	E	29/03/07 11:50	03/04/07 15:04	124	Baixa
mês 4	C	02/04/07 11:47	07/05/07 17:52	846	Média
mês 4	C	03/04/07 14:21	07/05/07 17:52	819	Média
mês 4	C	03/04/07 14:24	03/04/07 18:22	4	Baixa
mês 4	E	03/04/07 15:47	18/04/07 09:40	354	Baixa
mês 4	C	03/04/07 16:30	04/04/07 09:37	17	Baixa
mês 4	C	04/04/07 15:24	09/04/07 11:04	116	Média
mês 4	C	05/04/07 11:54	13/04/07 09:40	190	Média
mês 4	C	05/04/07 11:54	06/04/07 15:13	28	Média
mês 4	E	05/04/07 15:04	05/04/07 18:03	3	Baixa
mês 4	E	05/04/07 15:55	10/05/07 10:41	835	Média
mês 4	C	09/04/07 14:17	13/04/07 09:35	91	Média
mês 4	C	10/04/07 10:19	10/04/07 16:08	6	Média
mês 4	C	10/04/07 10:19	13/04/07 09:36	71	Média
mês 4	E	10/04/07 19:00	23/04/07 11:06	304	Baixa
mês 4	C	16/04/07 14:02	19/04/07 10:29	68	Média
mês 4	C	16/04/07 14:50	17/04/07 18:42	28	Média
mês 4	E	17/04/07 14:50	18/04/07 18:42	28	Baixa

mês 4	E	17/04/07 16:39	26/04/07 20:05	220	Baixa
mês 4	E	19/04/07 14:12	23/04/07 09:36	91	Baixa
mês 4	C	19/04/07 14:12	26/04/07 19:56	173	Média
mês 4	E	19/04/07 16:35	23/04/07 18:10	98	Baixa
mês 4	E	20/04/07 16:31	23/04/07 10:50	66	Baixa
mês 4	E	21/04/07 14:50	22/04/07 18:42	28	Baixa
mês 4	C	23/04/07 17:07	07/05/07 17:58	336	Média
mês 4	C	24/04/07 09:58	26/04/07 18:03	57	Média
mês 4	C	25/04/07 10:29	01/05/07 15:05	149	Média
mês 4	C	25/04/07 15:37	26/04/07 19:44	28	Baixa
mês 4	C	26/04/07 11:46	26/04/07 17:46	6	Baixa
mês 4	E	26/04/07 12:28	26/04/07 17:51	5	Baixa
mês 4	C	26/04/07 14:30	26/04/07 16:15	2	Baixa
mês 4	E	26/04/07 15:43	26/04/07 17:46	2	Baixa
mês 4	E	26/04/07 17:11	26/04/07 19:47	2	Baixa
mês 4	C	26/04/07 20:05	27/04/07 15:54	19	Média
mês 5	E	01/05/07 09:10	03/05/07 19:21	58	Baixa
mês 5	E	01/05/07 10:51	03/05/07 16:44	54	Baixa
mês 5	E	02/05/07 16:00	03/05/07 22:23	30	Baixa
mês 5	C	03/05/07 11:05	10/05/07 15:43	172	Média
mês 5	C	06/05/07 14:37	07/05/07 11:49	21	Média
mês 5	C	06/05/07 15:07	10/05/07 11:53	92	Média
mês 5	C	06/05/07 16:01	08/05/07 19:03	51	Média
mês 5	C	07/05/07 08:20	07/05/07 11:37	3	Baixa
mês 5	C	07/05/07 11:35	13/05/07 17:07	150	Média
mês 5	C	07/05/07 16:01	13/05/07 17:06	145	Média
mês 5	C	07/05/07 16:01	07/05/07 18:15	2	Média
mês 5	C	07/05/07 16:23	14/05/07 15:55	167	Média
mês 5	C	07/05/07 17:09	07/05/07 17:18	0	Baixa
mês 5	C	07/05/07 17:12	08/05/07 09:28	16	Média
mês 5	C	07/05/07 18:17	08/05/07 15:35	21	Média
mês 5	C	08/05/07 08:48	08/05/07 09:48	1	Baixa
mês 5	C	08/05/07 17:12	14/05/07 15:55	142	Média
mês 5	C	09/05/07 10:25	14/05/07 15:50	125	Média
mês 5	C	09/05/07 11:57	14/05/07 15:44	124	Média
mês 5	C	10/05/07 14:26	10/05/07 16:23	2	Baixa
mês 5	C	13/05/07 16:19	14/05/07 09:51	17	Média
mês 5	C	13/05/07 16:49	14/05/07 08:23	16	Média
mês 5	C	14/05/07 15:10	16/05/07 11:46	44	Média
mês 5	C	17/05/07 15:59	20/05/07 10:36	67	Média
mês 5	C	20/05/07 15:05	21/05/07 10:23	19	Média
mês 5	C	22/05/07 12:14	23/05/07 08:54	20	Média
mês 5	C	22/05/07 17:34	23/05/07 08:40	15	Média
mês 5	C	27/05/07 10:33	27/05/07 16:24	6	Baixa
mês 6	E	07/06/07 15:12	12/06/07 08:54	113	Baixa
mês 6	C	08/06/07 09:22	08/06/07 09:53	0	Baixa
mês 6	C	08/06/07 11:22	13/06/07 10:25	119	Média
mês 6	C	19/06/07 16:07	19/06/07 17:43	1	Baixa

**Legenda:**

**E:** Solicitação requisitando ações evolutivas

**C:** Solicitação requisitando ações corretivas

O cálculo do tempo total mensal das manutenções corretivas e o tempo médio mensal, representado pela métrica M1, são mostrados na Tabela 5.5, para o SGQ, e na Tabela 5.6, para o MGP. A Figura 5.6 representa a evolução mensal da M1 para os dois sistemas.

Tabela 5.5 – Dados para cálculo de M1 no SGQ

Mês	C(qtde)	$\Sigma$ Tempo	M1
mês 1	2	1617	808,50
mês 2	3	4286	1428,67
mês 3	24	6268	261,17
mês 4	20	3150	157,50
mês 5	25	1438	57,52
mês 6	3	120	40,00

Tabela 5.6 – Dados para cálculo de M1 no MGP

Mês	C(qtde)	$\Sigma$ Tempo	M1
mês 1	0	0	0
mês 2	1	118	118
mês 3	0	0	0
mês 4	1	664	664
mês 5	1	33	33
mês 6	0	0	0

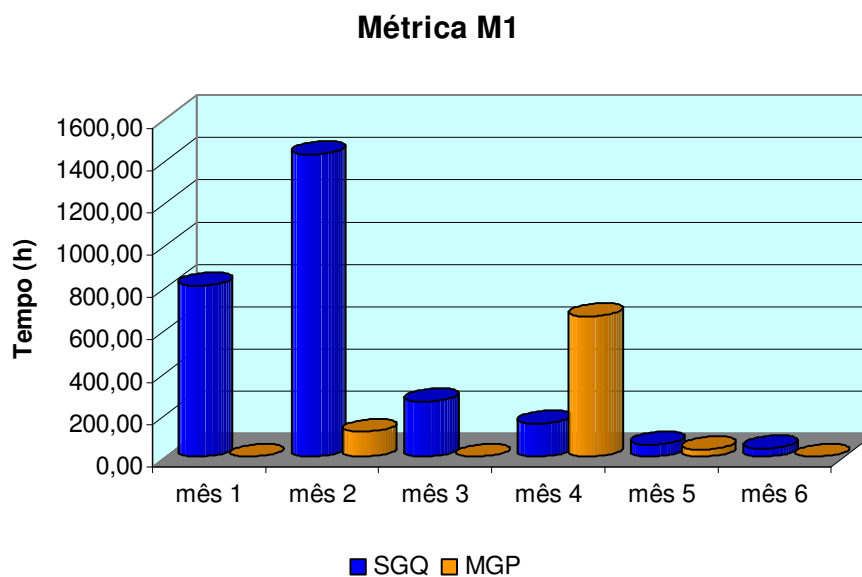


Figura 5.6 – Evolução mensal da métrica M1.

A Tabela 5.7, para o SGQ, e a Tabela 5.8, para o MGP, apresentam o tempo total mensal e o tempo médio mensal (M2) das manutenções evolutivas. A evolução mensal da métrica M2 para os dois sistemas pode ser vista na Figura 5.7.

Tabela 5.7 – Dados para cálculo de M2 no SGQ

Mês	Qtde	$\Sigma$ Tempo	M2
mês 1	5	3130	626,00
mês 2	2	1831	915,50
mês 3	9	1351	150,11
mês 4	13	2060	158,46
mês 5	2	142	71,00
mês 6	2	113	56,50

Tabela 5.8 – Dados para cálculo de M2 no MGP

Mês	Qtde	$\Sigma$ Tempo	M2
mês 1	1	1529	1529,0
mês 2	11	2462	223,8
mês 3	1	128	128,0
mês 4	0	0	0,0
mês 5	1	152	152,0
mês 6	0	0	0,0

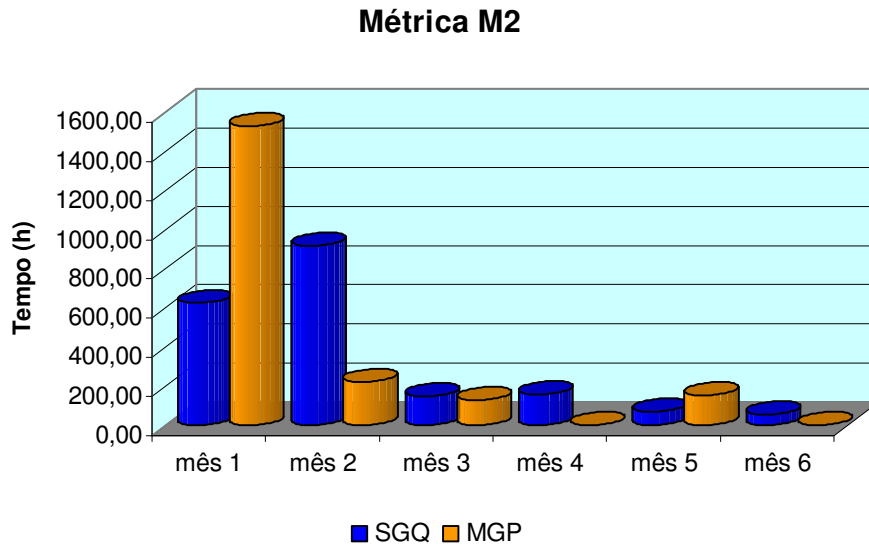


Figura 5.7 – Evolução mensal da métrica M2.

O tempo médio entre ocorrência de falhas (M3) é mostrado na Tabela 5.9, juntamente com o somatório dos tempos entre falhas consecutivas. A Figura 5.8 apresenta a comparação entre a métrica M3 dos dois sistemas, MGP e SGQ.

Tabela 5.9 – Dados para cálculo de M3

	<b>C(qtde)</b>	<b><math>\Sigma</math> Tempo</b>	<b>M3</b>
<b>MGP</b>	3	2058	1029,0 0
<b>SGQ</b>	77	3968	52,21

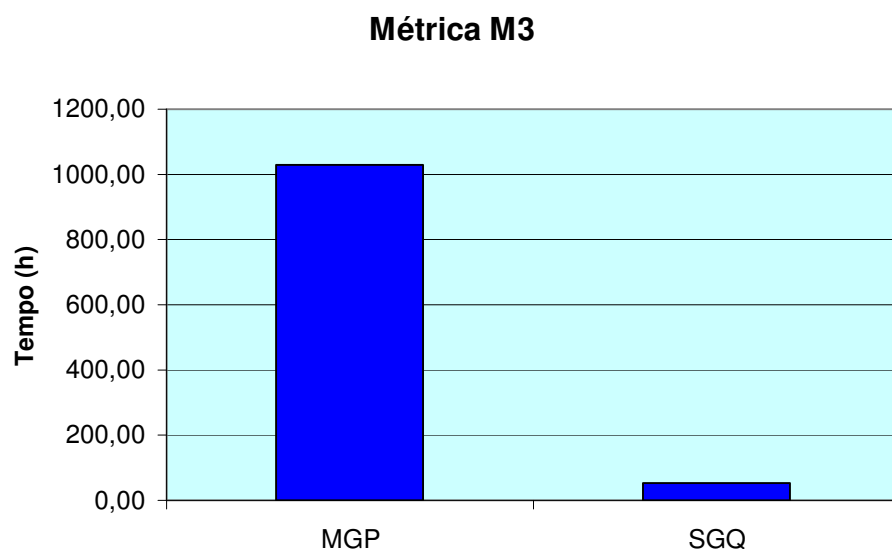


Figura 5.8 – Resultado da métrica M3 para o MGP e o SGQ.

Os valores de M4 e M5 a seguir são representados como número de solicitações corretivas ou evolutivas por 1000 pontos de função, respectivamente. Além disso, foi considerado o número fixo de Pontos de Função, isto é, esse número equivale à quantidade de pontos quando o sistema entra na fase de produção. Dessa forma, não foi feita uma recontagem dos pontos após as manutenções evolutivas.

Os dados da M4, quantidade de solicitações mensais e o valor de M4, são apresentados na Tabela 5.10 para o MGP e na Tabela 5.11 para o SGQ.

Tabela 5.10 – Dados para cálculo de M4 para o MGP

Mês	C(qtde)	PF	M4
mês 1	0	687	0
mês 2	1	687	1,46
mês 3	0	687	0
mês 4	1	687	1,46
mês 5	1	687	1,46
mês 6	0	687	0



Tabela 5.11 – Dados para cálculo de M4 para o SGQ

Mês	C(qtde)	PF	M4
mês 1	2	844	2,37
mês 2	3	844	3,55
mês 3	24	844	28,44
mês 4	20	844	23,70
mês 5	25	844	29,62
mês 6	3	844	3,55

A Figura 5.9 mostra o resultado comparativo da métrica M4 para os dois sistemas.

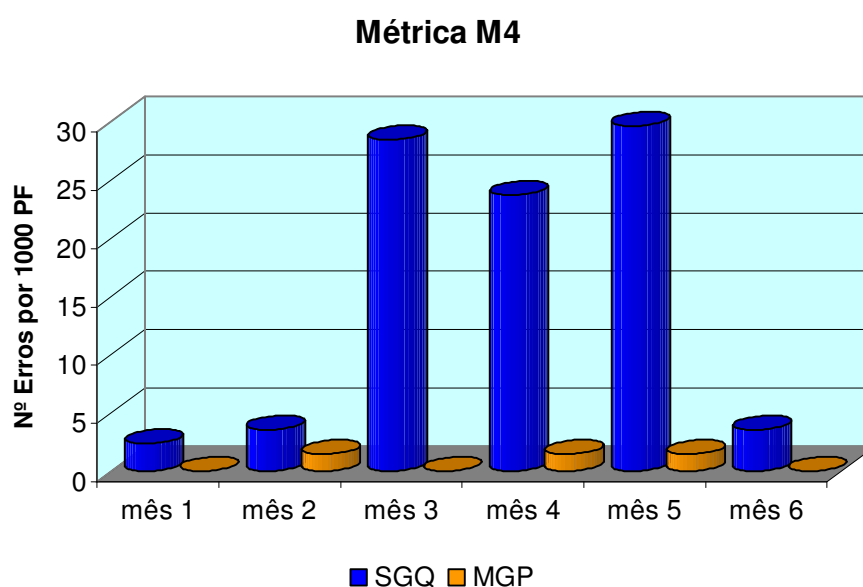


Figura 5.9 – Evolução mensal da métrica M4.

A quantidade de manutenções evolutivas mensais pode ser vista na Tabela 5.12, para o MGP, e na Tabela 5.13, para o SGQ, juntamente com o cálculo de M5.

Tabela 5.12 – Dados para cálculo de M5 para o MGP

Mês	Qtde	PF	M5
mês 1	1	687	1,46
mês 2	11	687	16,01
mês 3	1	687	1,46
mês 4	0	687	0,00
mês 5	1	687	1,46
mês 6	0	687	0,00

Tabela 5.13 – Dados para cálculo de M5 para o SGQ

Mês	Qtde	PF	M5
mês 1	5	844	5,92
mês 2	2	844	2,37
mês 3	9	844	10,66
mês 4	13	844	15,40
mês 5	2	844	2,37
mês 6	2	844	2,37

A Figura 5.10 mostra a comparação entre os dois sistemas por meio da evolução mensal da métrica M5.

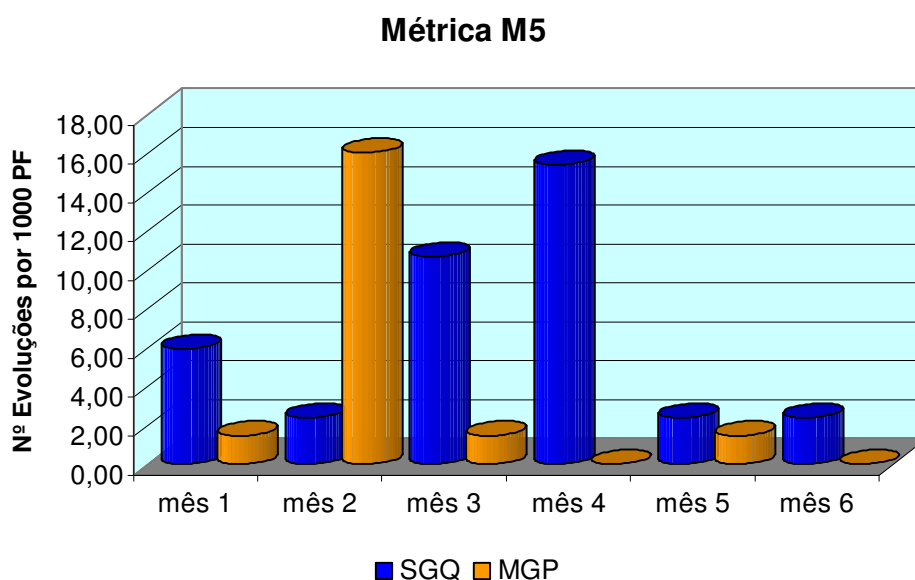


Figura 5.10 – Evolução mensal da métrica M5.

O dimensionamento das equipes de manutenção era reavaliado mensalmente para ambos os sistemas. Esse dimensionamento levou em consideração o número de solicitações recebidas e o tempo médio de efetivação das solicitações corretivas, ambos os critérios relativos ao mês anterior.

Para o MGP, a equipe de manutenção era composta inicialmente de 2 profissionais. Em termos gerais esse sistema não apresentou necessidade de manutenções corretivas, sendo que no primeiro mês foi feita apenas uma solicitação de manutenção corretiva. Esse fator

associado com o tempo de resposta permitiu que a gerência responsável reduzisse a equipe de manutenção a apenas uma pessoa, número que foi mantido devido ao resultado da métrica durante o período de análise.

Para o SGQ, a equipe de manutenção era composta inicialmente de 1 profissional. No mês 1, o número de manutenções corretivas, associado com o valor de M1, fez com que a equipe fosse aumentada no mês 2 para 2 pessoas. Nesse mês, houve um aumento do número de correções e o valor de M1 também aumentou, assim, para o mês 3, a equipe foi aumentada para 4 pessoas. O número de correções aumentou nos demais meses, mas mesmo com o valor de M1 diminuindo para os meses 4, 5 e 6, a gerência decidiu manter a equipe em 4 pessoas por razões de precaução.

O dimensionamento das equipes pode ser visto na Tabela 5.14.

Tabela 5.14 – Dimensionamento das equipes

	Mês 1	Mês 2	Mês 3	Mês 4	Mês 5	Mês 6
MGP	2	1	1	1	1	1
SGQ	1	2	4	4	4	4

#### 5.4.4 – Análise Comparativa dos Resultados

A análise dos resultados basear-se-á na evolução de cada métrica em separado, no inter-relacionamento entre as métricas e na comparação de desempenho entre os processos para implementação do MGP e SGQ.

A princípio, foi possível perceber que, para se avaliar o aspecto manutenção de um sistema, seria necessário um conjunto de métricas que fossem concebidas de modo a avaliar quesitos em separado (tempo de resposta das equipes, a frequência de ocorrência de solicitações etc.), para que cada métrica pudesse complementar as outras. Por exemplo: a métrica tempo médio de resposta aumentou devido ao maior número de solicitações no período, para uma mesma equipe.

A análise do desempenho das métricas foi feita a partir de grupamentos. Nesse sentido, M1 e M4 formam um grupo para análise. Isso se justifica pelo fato de que essas métricas têm como informações básicas para sua obtenção o número de solicitações corretivas, ou seja, solicitações do tipo C, conforme pode ser visto na Figura 5.11.

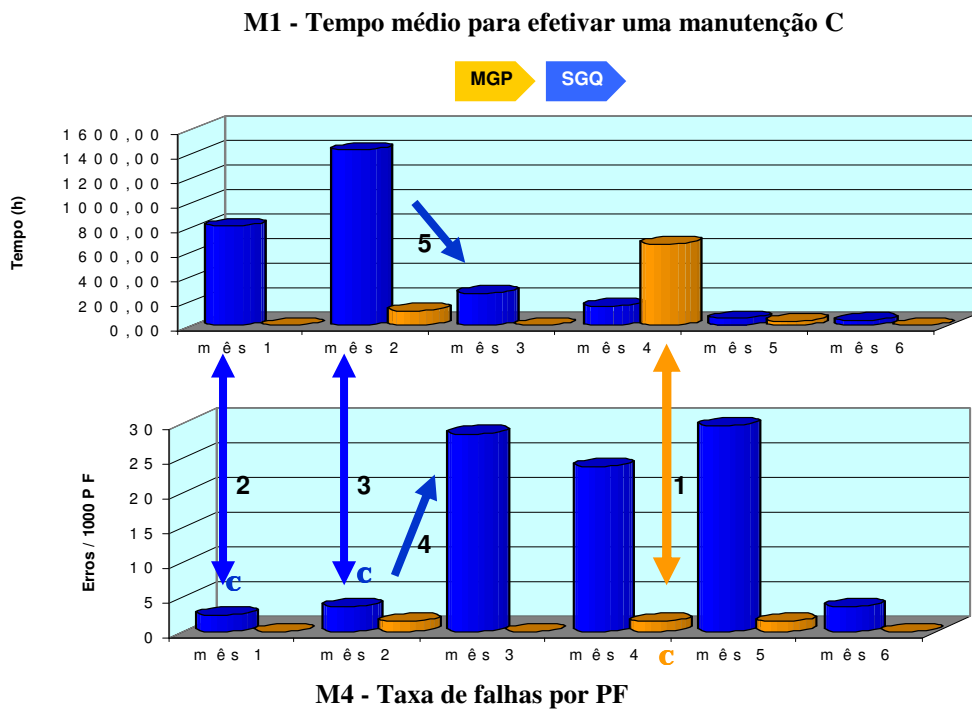


Figura 5.11 – Análise conjunta das métricas M1 e M4.

Para o MGP, o comportamento de M1 não apresenta uma tendência clara. Contudo, os dados mensais apresentam valores pequenos, com exceção de um pico ocorrido no 4º mês do período de estudo, conforme ilustrado na seta 1. A análise permitiu chegar à conclusão que o aumento deveu-se a dois fatores, quais sejam: decréscimo no número de profissionais envolvidos, ou seja, o número passou de 2 profissionais para 1, juntamente com o aumento da complexidade da solicitação corretiva.

Para o SGQ, a tendência decrescente de M1 significa que o tempo médio para a efetivação de uma solicitação do tipo corretiva diminua. Esse comportamento pode significar que:

- a. o número desse tipo de solicitações diminuiu;
- b. a complexidade das mesmas diminuiu; ou,
- c. o número de profissionais dedicados à tarefa de manutenção do sistema aumentou.

Contudo, analisando o comportamento da M4, pode-se afastar a diminuição do número de solicitações. A conclusão que se pode chegar para justificar esse comportamento, aparentemente controverso, foi o aumento no número de profissionais envolvidos, ou seja,

o número passou de 1 para 4 profissionais, juntamente com o decréscimo da complexidade das solicitações corretivas, conforme pode ser visto pelas setas 2, 3, 4 e 5.

A comparação entre os processos relativamente à evolução da métrica M1 foi baseada numa análise ponto a ponto entre os valores mínimos, médios e máximos de M1 para os dois sistemas em estudo. Os dados são mostrados na Tabela 5.15, a seguir:

Tabela 5.15 – Comparação para M1

	<b>SGQ</b>	<b>MGP</b>
<b>Média</b>	458,89	135,83
<b>Mínimo</b>	40,00	0,00
<b>Máximo</b>	1428,67	664,00

Analisando-se os dados referentes ao desempenho da métrica M1, pôde-se observar que o processo utilizado para o desenvolvimento do MGP em comparação ao SGQ apresentou resultados melhores no que se refere aos valores mínimos, médios e máximos.

Como conclusão indireta, também foi possível identificar que o custo para manutenção do SGQ foi maior que para o MGP, em razão do número de profissionais envolvidos. Ainda assim, mesmo com o acréscimo de profissionais para o SGQ, este apresentou tempos superiores para a efetivação das correções solicitadas.

O grupo seguinte a ser analisado é formado pelas métricas M2 e M5. A justificativa, a exemplo do grupo anterior, deve-se ao fato de que as informações básicas para sua obtenção são referentes a solicitações evolutivas, ou seja, solicitações do tipo E, conforme pode ser visto na Figura 5.12 .

## M2 - Tempo médio para efetivar uma manutenção E

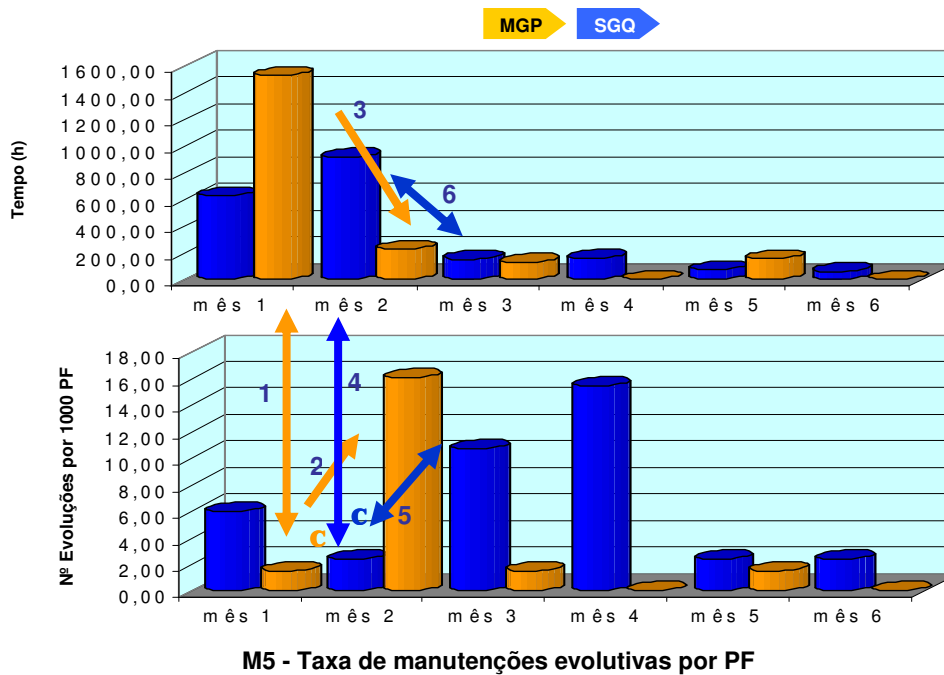


Figura 5.12 – Análise conjunta das métricas M2 e M5.

Para o MGP, o valor relativamente grande para a M2 relativamente ao valor de M5 deve-se à complexidade das manutenções evolutivas, conforme pode ser visto pela seta 1. Em geral, o comportamento de M2 apresenta tendência de decréscimo no tempo de efetivação das solicitações evolutivas. Entretanto, o número de solicitações evolutivas (por PF) representado pelo comportamento de M5 apresenta comportamento constante, com exceção de um pico ocorrido no 2º mês do período de estudo, conforme pode ser visto pela indicação das setas 2 e 3.

Para o SGQ, a tendência crescente de M2 para os meses 1 e 2 significa que o tempo médio para a efetivação de uma solicitação do tipo evolutiva aumenta. Contudo, o valor de M5 diminui para esses dois meses. Esse comportamento deve-se ao fato do aumento da complexidade no mês 2, conforme indicação da seta 4.

Contudo, analisando-se o comportamento da M5, pode-se afastar a diminuição do número de solicitações. Aliás, o comportamento dessa métrica não apresenta uma tendência clara. A conclusão que se pode chegar para esse comportamento deve-se ao aumento no número de profissionais envolvidos, conforme já mencionado (o número passou de 1 para 4) e o

decréscimo da complexidade das solicitações evolutivas, conforme indicação das setas 5 e 6.

A comparação entre os processos relativamente à evolução da métrica M2 foi baseada numa análise ponto a ponto entre os valores mínimos, médios e máximos medidos para os dois sistemas em estudo. Os dados são mostrados na Tabela 5.16, a seguir:

Tabela 5.16 – Comparação para M2

	<b>SGQ</b>	<b>MGP</b>
<b>Média</b>	329,60	338,80
<b>Mínimo</b>	56,50	0,00
<b>Máximo</b>	915,50	1529,00

Analisando-se os dados referentes ao desempenho da métrica M2, pôde-se observar que o processo utilizado para o desenvolvimento do MGP em comparação ao SGQ apresentou resultados demais melhores no que se refere aos valores mínimos e médios. Contudo, o maior tempo médio para se efetuar uma evolução ocorreu com o processo do MGP. A justificativa encontrada deveu-se ao nível de complexidade envolvida, que foi avaliado em baixo, médio e alto. Para uma melhor compreensão, outros fatores devem ser considerados.

Importante ainda ressaltar outra informação que pode advir do inter-relacionamento entre as métricas: M1 e M5, e M2 e M4. Esse relacionamento deve-se à qualidade da solução adotada para cada solicitação. Isso quer dizer que tempos pequenos para atendimento de solicitação podem não necessariamente ter comprometimento com a qualidade, gerando, posteriormente, novas falhas no período ou nos períodos subsequentes. Não foi evidenciada na aplicação em questão a ocorrência desses efeitos colaterais e da implementação de soluções paliativas ou emergenciais.

Tabela 5.17 – Comparação para M3

	<b>M3</b>
<b>MGP</b>	1029
<b>SGQ</b>	52,21

O tempo médio entre ocorrência de uma falha para o MGP foi muito superior ao SGQ, Tabela 5.17. Isso significa que praticamente a cada 2 (dois) dias o SGQ apresentou um

erro, enquanto as falhas para o MGP ocorreram aproximadamente a cada 40 (quarenta) dias, durante o período de estudo.

Entretanto, é importante ressaltar que a M3, assim como as demais métricas, foi medida logo após a fase de homologação do sistema, ou seja, no período inicial da fase de produção. Logo, o sistema ainda estava na fase transitória, portanto, sujeito à falhas e ajustes. Nesse caso, seria importante verificar a evolução do comportamento dessa métrica durante um período mais longo de observação. Isso possibilitaria que o sistema saísse da fase transitória e entrasse numa situação mais estável dentro da fase de produção.

Desta forma, é necessário considerar que a passagem da fase de desenvolvimento para a fase de produção ainda está sujeita a alguma instabilidade, tanto no que se refere à experiência dos usuários quanto na efetiva execução dos requisitos levantados no início do projeto. Essa situação ensejou a necessidade de avaliação mensal de algumas métricas, em vez de aplicá-las ao período de análise como um todo, pois a obtenção de resultados em pequenos intervalos possibilitou a percepção de evolução nos resultados das métricas, que neste caso pode ser observada mensalmente.

A comparação entre os processos relativamente à evolução da métrica M4 foi baseada numa análise ponto a ponto entre os valores mínimos, médios e máximos medidos para os dois sistemas em estudo. Os dados são mostrados na Tabela 5.18, a seguir:

Tabela 5.18 – Comparação para M4

	<b>SGQ</b>	<b>MGP</b>
<b>Média</b>	15,21	0,73
<b>Mínimo</b>	2,37	0,00
<b>Máximo</b>	29,62	1,46

Analisando-se os dados referentes ao desempenho da métrica M4, pôde-se observar que o processo utilizado para o desenvolvimento do MGP em comparação ao SGQ apresentou resultados melhores no que se refere aos valores mínimos, médios e máximos. O SGQ apresentou mais necessidade de correção que o MGP, mesmo sendo a relação entre número de correções mensais e o número de pontos de função.



A comparação entre os processos relativamente à evolução da métrica M5 foi baseada numa análise ponto a ponto entre os valores mínimos, médios e máximos medidos para os dois sistemas em estudo. Os dados são mostrados na Tabela 5.19, a seguir:

Tabela 5.19 – Comparação para M5

	<b>SGQ</b>	<b>MGP</b>
<b>Média</b>	6,52	3,40
<b>Mínimo</b>	2,37	0,00
<b>Máximo</b>	15,40	16,01

Analisando-se os dados referentes ao desempenho da métrica M5, pôde-se observar que o processo utilizado para o desenvolvimento do MGP em comparação ao SGQ apresentou resultados melhores no que se refere aos valores mínimos e médios. No que se refere ao número máximo de evoluções em relação aos pontos de função, os valores foram muito próximos, ainda que o número de evoluções por ponto de função tenha sido maior para o MGP que para o SGQ. O SGQ apresentou mais necessidade de mudanças, evoluções e adaptações que o MGP, tanto em termos absolutos como relativos.

Em termos gerais, a partir da análise das métricas, pode-se verificar que a qualidade apresentada pelo MGP foi superior ao SGQ, tanto no que se refere à estabilidade do sistema, à estabilidade na apresentação de falhas, quanto no número de falhas envolvidas. Ademais, deve-se levar em consideração que a manutenção do SGQ demandou custos com o aumento do número de profissionais.

Considerando que ambos os processos têm equipes com graus de maturidade similares, pode-se concluir que a adoção de testes bem estruturados na fase de implementação tem um impacto muito positivo na fase de produção de sistemas.

Conclui-se ainda que a economia advinda do não emprego de testes durante a fase de produção, bem como o menor tempo para a implementação podem ocasionar a passagem para a produção de um sistema com qualidade inferior, necessitando continuamente de manutenção.

## 6 – CONCLUSÕES E TRABALHOS FUTUROS

Uma das preocupações deste trabalho é ressaltar a importância da garantia da qualidade em produtos de software. A percepção de qualidade é dependente da qualidade do produto, que por sua vez é decorrência da qualidade do processo. Nesse, estão incluídas as visões da gerência de qualidade de software, testes e métricas.

Deve-se dar importância à realização de testes no desenvolvimento e homologação de software, indicando métodos de teste a serem utilizados para avaliar o maior número de requisitos de qualidade, considerando os recursos e tempo disponíveis. A medição da qualidade do processo e do produto proporciona aos gerentes e profissionais de Engenharia de software a oportunidade de melhorar CDU e aperfeiçoar o processo de software, além de ajudar no planejamento e no controle de um projeto, possibilitando também a avaliação da qualidade do produto que é criado.

Foram adotadas algumas instâncias para se garantir a qualidade do produto, tanto em suas características objetivas, aquelas relativas à quantidade de erros apresentados, quanto em suas características subjetivas. As instâncias escolhidas foram:

- O embasamento em um modelo de processo (RUP).
- O emprego de estratégias de testes com o enfoque em aspectos funcionais.
- A utilização de métricas para avaliação da qualidade no que se refere à manutenibilidade.

Neste trabalho verificou-se que processos mais elaborados e processos simplificados apresentam suas vantagens e desvantagens, ambos com pontos fortes e fracos. Em termos gerais, a competição entre as duas abordagens reside na avaliação dos aspectos velocidade, qualidade e custo. É difícil a obtenção de um produto que atenda de forma equilibrada a esses três quesitos. Em geral, apenas dois deles são satisfeitos em prejuízo do terceiro quesito. Ou seja, um produto pode ser entregue rápido e com qualidade, entretanto o custo pode extrapolar as expectativas; ou ainda, ele pode ser produzido a custo satisfatório e com boa qualidade, mas nesse caso haverá um risco de demanda maior de tempo.

O que se pode perceber é que cabe ao cliente, juntamente com a empresa desenvolvedora, definir de forma precisa qual o aspecto de qualidade que será o enfoque do seu processo.

Nesse sentido, aspectos como tempo disponível para o projeto, recursos financeiros, tipo de clientela/usuários, tipo de aplicação, entre outros, devem modelar qual será a resposta mais apropriada.

Um dos processos, por exemplo, adotou o RUP como pilar. O RUP tem ênfase na diminuição dos riscos e nas adaptações das mudanças, através da iteratividade. O desenvolvimento iterativo está muito mais alinhado com trabalho da equipe de desenvolvimento do que as abordagens tradicionais, porém tem um custo em complexidade de administração. Não adianta utilizar o processo do RUP se não forem bem estruturadas as fases e as disciplinas. Esse tipo de desenvolvimento envolve um maior planejamento para não ocorrerem muitos riscos no projeto, visto que, além de um planejamento global, um planejamento detalhado deve ser realizado a cada iteração. Isto envolve uma contínua avaliação e correção dos problemas. Os artefatos também devem ser modificados, revisados e aprovados a cada iteração, por isso deve ser feita uma ponderação antes de se pensar em qual tipo de processo utilizar para alcançar a qualidade, considerando os seguintes aspectos:

- tempo;
- estabilidade;
- custo.

A grande vantagem do processo não unificado em relação ao o processo unificado foi o tempo para implementação, haja vista que esse não necessitou se deter aos protocolos e aos fluxos de atividades inerentes àqueles dos processos distribuídos. Todavia, o emprego de ferramentas automáticas para a realização dos testes seria uma boa abordagem para a redução da distância entre os tempos gastos pelos dois processos, pois a automação afetaria sobremaneira o processo unificado com testes manuais, e muito pouco um processo não unificado.

Neste trabalho pôde-se verificar a importância de testes nos produtos de software. Nesse sentido, os principais benefícios obtidos pela utilização de testes num processo de software podem ser agrupados nas seguintes vantagens:

- Para os clientes, uma vez que o sistema não apresentou grandes necessidades de correções.

- Para os desenvolvedores, na medida em que as falhas foram identificadas antes da fase de produção. Além disso, beneficia-se a própria imagem da empresa, pois vincula seu nome a produtos que apresentam um alto grau de imunidade a falhas.

A utilização de métrica foi importante, pois proporcionou informações relevantes quanto ao desempenho das atividades de manutenção nos sistemas MGP e SGQ. Com a utilização das métricas de manutenibilidade, foi possível avaliar a estabilidade do sistema, o tempo de resposta efetivo de equipes de manutenção, além de auxiliar a gerência de projeto no dimensionamento dessas equipes no seu nível de maturidade e na quantidade de sistemas pelos quais as equipes são responsáveis.

Entretanto, para a classificação da manutenibilidade dos sistemas MGP e SGQ há a necessidade de se aplicar o conjunto de métricas aqui propostas a outros sistemas. Com a aplicação em diversos sistemas, poder-se-ia construir uma base de análise do desempenho das métricas com vistas ao estabelecimento de ajustes e metas de desempenho para cada uma das métricas, respeitando-se, obviamente, a necessidade dos usuários/sistema específicos.

Como evolução a ser perseguida para trabalhos futuros, tem-se a possibilidade de avaliar a complexidade das solicitações, tanto evolutivas quanto corretivas, em conjunto com as métricas. Essa avaliação provê indicadores importantes sobre o tempo gasto para a compreensão e efetivação da solicitação, além de apresentar outros aspectos de análise que permitem a melhor inferência sobre os aspectos de manutenibilidade.

Acredita-se que o objetivo principal deste estudo foi atingido, uma vez ter sido possível testar, avaliar a importância dos testes e aplicar o conceito de métricas para manutenibilidade. Sabe-se que por ser este estudo de caráter exploratório, o delineamento mais claro da aplicação das métricas e o estabelecimento de metas carecem de aplicação a um conjunto maior de produtos de software. A extensão deste estudo viabilizaria o preenchimento de lacunas e questões que porventura permaneçam em aberto.

Alguns aspectos que podem gerar pesquisas futuras, em relação a testes e métricas, seriam:

- a adoção de ferramentas automáticas para a realização dos testes, quer sejam pagas ou de uso gratuito (free);
- a extensão do estudo a um universo maior de produtos de software;

- a comparação entre os dois processos, em termos dos custos envolvidos em cada intervenção corretiva no sistema na fase de produção, por exemplo: eventuais prejuízos, multas etc.

## 7 – REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Programa Brasileiro da Qualidade e Produtividade. MICT: Brasília-DF, Brasil, 1990.
- [2] Programa Brasileiro da Qualidade e Produtividade. Fase 2: Projetos Estratégicos. MICT: Brasília-DF, Brasil, 1996.
- [3] Programa Brasileiro da Qualidade e Produtividade. Fase 3: Qualidade: Mostre, Exija, Qualidade: Mostre, Exija, Viva. INMETRO e IBQP: Brasília-DF, Brasil, 1998.
- [4] Dicionário Houaiss da língua portuguesa. (2001). Objetiva, Rio de Janeiro, 1 CD-ROM.
- [5] ABNT – Associação Brasileira de Normas Técnicas. (1994a). *NBR ISO 8402/1994: Gestão da qualidade e garantia da qualidade – Terminologia*, Rio de Janeiro, ABNT.
- [6] Hughes, B. e Cotterel, M. (2002). *Software project management* (3<sup>rd</sup>. ed.), McGraw-Hill, London.
- [7] ABNT – Associação Brasileira de Normas Técnicas. (1994b). *NBR ISO 9000-1/1994: Normas de gestão da qualidade e garantia da qualidade*, Rio de Janeiro, ABNT.
- [8] Maranhão, M. (2001). *ISO 9000: manual de implantação versão ISO 2000* (3<sup>a</sup>. ed.). Rio de Janeiro, Qualitimark.
- [9] Rocha, A.R.C. et al. (1994). Uma experiência na definição do processo de desenvolvimento e avaliação de software segundo as Normas ISO, Relatório Técnico ES-302/94, COPPE/UFRJ, Rio de Janeiro.
- [10] E. 802 – (2/2007) Framework and methodologies for the determination and application of QoS parameters. International Telecommunication Union-ITU.
- [11] Jacobson, I. et al. (1999). *The unified software development process*, Addison-Wesley, Reading, USA.
- [12] Rational Software Corporation White Paper. Rational Unified Process – Best Practices for Software Development Teams. Rational Software Corporation, 1998.
- [13] Eriksson, H.E. e Penker, M. (1998). *UML Toolkit*. John Wiley, New York, USA.
- [14] Jacobson, I. (1998). “Objectory is the unified process”. In: *Component Strategies*, 1(10), 67-72.
- [15] Humphrey, W. (1987). *Characterizing the software process: a maturity framework*, version 1.0. Technical report CMU/SEI-87-TR-11. CMU/SEI, Software Engineering Institute – Carnegie Mellon University, Pittsburgh, USA.

- [16] Humphrey, W. Technical report CMU/SEI-00-TR- 24 (2000). Institute - Carnegie Mellon University.
- [17] Software Engineering Institute. (2000). *CMMI Model Components Derived from CMMI<sup>sm</sup> – SE/SW*, version 1.0. Technical report CMU/SEI-00-TR-24. Software Engineering Institute – Carnegie Mellon University, Pittsburgh, USA.
- [18] PAULK, M. et al. (1993). *Capability Maturity Model for Software*, Version 1.1. Software Engineering institute Carnegie Mellow University, Pittsburgh, PA. Disponível em: <<http://www.sei.cmu.edu/pub/documents/93.reports/pdf/tr24.93.pdf>>. Acesso em: 18 set. 2004.
- [19] CHIRISSIS, M., SHRUM S.(2003) Introduction. In:\_\_\_\_\_.*CMMI: guidelines for process integration and product improvement*. 2 ed. Boston: Addison-Wesley, Cap.1
- [20] International Standard Organization. (1999). *ISO/IEC TR 15504 – Information Technology – Software process assessment*, International Organization for Standardization, Montreal.
- [21] Machado, C. F.; Burnett, R. (2001). “Gerência de projetos na engenharia de software em relação às práticas do PMBOK”. In: *Anais da XII CITS: QS Métricas para Qualidade e Produtividade de Software*, Editora Universitária Champagnat, 172-181.
- [22] ABNT – Associação Brasileira de Normas Técnicas. (1994a). *NBR ISO 8402/1994: Gestão da qualidade e garantia da qualidade – Terminologia*, Rio de Janeiro, ABNT.
- [23] International Standard Organization. (2001). *ISO/IEC 15288: Information Technology – System life cycle processes*, versão FDIS, Montreal, ISO/IEC JTC1 SC7.
- [24] ABNT – Associação Brasileira de Normas Técnicas. (1998). *NBR ISO 12207: Tecnologia de informação – Processos de ciclo de vida de software*, Rio de Janeiro, ABNT.
- [25] Boehm, B. (1988). “A spiral model of software development and enhancement, *IEEE Computer*”, 21(5), 61-72.
- [26] Royce, W. (1998). *Software project management framework*. Addison Wesley Longman, Massachusetts.
- [27] Takashina, N.T. e Flores, C.X. (1996). *Indicadores da qualidade e do desempenho: como estabelecer metas e medir resultados*, Qualitymark, Rio de Janeiro.
- [28] Molinare, L. (2006). *Testes de software: produzindo sistemas melhores e mais confiáveis* (3ª ed.), Érika, São Paulo.
- [29] Bartie, A. (2002). *Garantia de qualidade de software* (4ª ed.), Campus, São Paulo.
- [30] Bastos, A.; Rios, E.; Cristalli, R.; Moreira, T. (2006). *Base de conhecimento em teste de software*, Traço & Photo, Rio de Janeiro.

- [31] Beizer, B. (1995). *Black-box testing*. Wiley, USA.
- [32] Black, R. (1999). *Managing the testing process*. Microsoft Press, USA, 1999.
- [33] Myers, G. (1979). *The art of software testing*. Wiley, New York, USA.
- [34] Pressman, R.S. (1995). *Engenharia de software*, Makron Books, São Paulo.
- [35] Calçado, V.L.X.S.; Calçado, I.R.; Camapum, J.F.; Jr., A.R.M. (2007). Avaliação do resultado em produtos de software a partir da utilização de testes no processo de Desenvolvimento. *6<sup>a</sup> Conferencia Iberoamericana en Sistemas, Cibernética e Informática*: CISCI, Orlando, Florida – E.U.A, Volume III, 169-174.
- [36] Fernandes, A. (1995). *Gerência efetiva de software através de métricas*. Atlas, São Paulo.
- [37] Vazquez, C.; Simões, G. e Albert, R. (2003). *Análise de pontos de função, medição, estimativas e gerenciamento de projetos de software*, Érica, São Paulo.
- [38] Pressman, R.S. (2000). *Software engineering: a practioner's approach* (5<sup>th</sup>. ed.), McGraw-Hill, New York.
- [39] Carvalho, A.M.B.R. e Chiossi, T.C.S. (2001). *Introdução à Engenharia de Software*, Editora da Unicamp, Campinas, SP.
- [40] Basili, V. R.; Caldiera, G. (1994). Goal Question Metric Paradigm, Encyclopedia of software engineering.
- [41] Basili, V.R.; Rombach, H.D.: 14 (1988). “The TAME Project: Towards Improvement-Oriented Software Environments”. *IEEE Transactions on Software Engineering*, 6, pp. 758-773
- [42] Basili, V.R.; Rombach, H.D. Rombach, (1994a) “*Goal Question Metric Paradigm*”; Encyclopedia of software Engineering (Marciniak, ‘ J.J’, Editor) Volume 1, John Weley & Sons, pp.528, 532
- [43] Basili, V.R.; Rombach, H.D. Rombach, Goal Question Metric Method - A practical Guide for Quality Improvement of software Developoment
- [44] Ramos, C.S, Oliveira K.M, Anquetil, N., (2004) “*Conhecimento Sistemas Legados através de Métricas de Software*”. “III Simpósio Brasileiro de Qualidade de Software – Brasília, DF, Brasil”, 1-15.
- [45] Goal Question Metric Method. (2006) *A practical Guide for Quality Improvement of software Developoment*. Disponível em: [www.gqm.nl](http://www.gqm.nl):. Acessado em: 10 novembro.
- [46] International Organization for Standardizaton (ISO); International Electrotechnical Commission (IEC). (2003). *ISO/IEC 9126-1: modelo de qualidade*.
- [47] International Organization for Standardizaton (ISO); International Electrotechnical Commission (IEC). (2003). *ISO/IEC TR 9126-2: métricas externas*.



- [48] Calçado, V.L.X.S.; Calçado, I.R.; Santos, G.C.; Camapum, J. F. (2007). “The use of metrics in the quality assessment of software product maintenance”. *The 11<sup>th</sup> World Multiconference on Systemics, Cybernetics and Informatics, WMSCI 2007*, Orlando, Florida, USA, July 8 – 11, Volume III, 162-165.

## **ANEXOS**

## A – CASO DE USO DO MGP

ESPECIFICAÇÃO DE CASO DE USO		Código: CDU 001
<b>Identificação:</b>	CDU 001 –Diretriz Estratégica	
<b>Cliente:</b>	Órgão Governamental	
<b>Projeto:</b>	Módulo de Gestão de Planejamento	
<b>Versão do CDU:</b>	1.0	
<b>Aceite:</b>		

### Diretriz Estratégica

#### Descrição Sucinta

Este caso de uso permite a inclusão, alteração e exclusão de Diretrizes Estratégicas e de suas Ações.

#### Atores

- Usuário do Órgão Governamental

#### Pré-Condições

- O Usuário deve estar logado no sistema;
- O Usuário deve ter permissão de acesso à funcionalidade.

#### Pós-Condições

- Não se aplicam.

#### Fluxo Básico

1. O caso de uso tem início quando o Ator seleciona a opção “Diretriz Estratégica - Listar”;
2. O Sistema monta o menu da página de acordo com as permissões do usuário;
3. O Sistema exibe a Tela “*Diretriz Estratégica – Listar*” (RN 01) contendo:

Filtro para Pesquisa de Diretrizes Estratégicas

Campo	Descrição	Editável?	Obrigatório?	Máscara
Nome	Permite Filtrar pelo Nome da Diretriz Estratégica. Se não for informado, não filtra por este campo. (RN 01.2)	Sim	Não	Não se aplica
Situação	Permite Filtrar as Diretrizes Estratégicas pela Situação: Ativa;	Sim	Não	Não se aplica

	ou Inativa. Se não for selecionada uma opção, não filtra por este campo.			
--	--	--	--	--

- Lista de Diretrizes Estratégicas ordenadas pelo “Nome”. Cada Diretriz Estratégica possui os seguintes campos:

<b>Campo</b>	<b>Descrição</b>	<b>Máscara</b>
Nome	Nome da Diretriz Estratégica	Não se aplica
Situação	Situação da Diretriz Estratégica (Ativa ou Inativa)	Não se aplica

- Caso os registros na base de dados, ou da pesquisa sejam superiores a 10, o sistema exibirá a lista fragmentada através do mecanismo de paginação definido na arquitetura do sistema.
  - Se nenhum filtro for informado, o Sistema deve mostrar todos os registros;
4. O Usuário pode **escolher**:
- Voltar à “**Página Principal**” do sistema clicando sobre o respectivo link na linha de navegação do sistema. Ao selecionar esta opção o Caso de Uso é encerrado;
  - **Filtrar** os registros da tela. Aciona o *Fluxo Alternativo “Filtrar”* (A 01);
  - **Alterar uma Diretriz Estratégica** clicando em “*alterar*” para algum registro da tela. Aciona o *Fluxo Alternativo “Alterar Diretriz Estratégica”* (A 02);
  - **Excluir uma Diretriz Estratégica** clicando em “*excluir*” para algum registro da tela. Aciona o *Fluxo Alternativo “Excluir Diretriz Estratégica”* (A 03);
  - **Incluir uma Diretriz Estratégica** clicando em “*incluir*”. Aciona o *Fluxo Alternativo “Incluir Diretriz Estratégica”* (A 04);
5. O Caso de uso é encerrado

## Fluxos Alternativos

### (A 01) Filtrar

1. O Usuário informa os campos “Nome” e “Situação”;
2. O Usuário escolhe a opção **Filtrar**;
3. O Sistema valida os dados informados;
4. Caso **não ocorram restrições**;
  - a. O Sistema exibe os registros que atendem aos filtros especificados na Tela (se nenhum valor foi especificado no Filtro, o Sistema deve buscar todos os registro no banco de dados);
5. Caso alguma **restrição tenha ocorrido** (E 03);
  - a. O Sistema exibe o motivo da restrição;
  - b. O Sistema posiciona o cursor no primeiro campo a ser informado novamente pelo usuário;
6. O Sistema retorna ao “
7. Fluxo Básico”;
8. O fluxo é encerrado.

**(A 02) Alterar Diretriz Estratégica**

1. O Sistema monta o menu da página de acordo com as permissões do usuário logado;
2. O Sistema exibe a Tela “Alterar” contendo:

a. Campos da Diretriz Estratégica

<b>Campo</b>	<b>Descrição</b>	<b>Editável?</b>	<b>Obrigatório?</b>	<b>Máscara</b>
Nome da Diretriz Estratégica	Nome da Diretriz Estratégica. (E 01) (E 06)	Sim	Sim	Não se aplica
Situação	Situação da Diretriz Estratégica (Ativa ou Inativa). (E 01)	Sim	Sim	Não se aplica

- b. Lista de Ações Estratégicas de Curto Prazo da Diretriz, ordenadas pelo campo “Nome”. Cada ação estratégica possui os seguintes campos:

<b>Campo</b>	<b>Descrição</b>	<b>Editável?</b>	<b>Obrigatório?</b>	<b>Máscara</b>
Nome da Ação Estratégica	Nome da Ação Estratégica. (E 01) (E 07)	Sim	Sim	Não se aplica
Situação	Situação da Ação Estratégica (Ativa ou Concluída) (E 01)	Sim	Sim	Não se aplica

3. O Sistema carrega na Tela os dados do registro selecionado;
4. O Usuário altera os dados que desejar;
5. O Usuário pode **escolher**:
  - a. **Voltar à “Página Principal”** do sistema clicando sobre o respectivo link na linha de navegação do sistema.
    - i. O Caso de Uso é encerrado;
  - b. **Voltar à Tela “Diretriz Estratégica – Listar”** clicando sobre o respectivo link na linha de navegação do sistema.
    - i. O sistema vai para o “  
ii. Fluxo Básico”;
    - iii. O fluxo é encerrado;
  - c. **Excluir uma Ação**, clicando em “*Excluir*” para a ação.
    - i. Aciona o *Fluxo Alternativo “Excluir Ação”* (A 05);
    - ii. O sistema vai para o início deste Fluxo (Alterar);
  - d. **Incluir uma Ação**, clicando em “*incluir*” para a ação.
    - i. Aciona o *Fluxo Alternativo “Incluir Ação”* (A 06);
    - ii. O sistema vai para o início deste Fluxo (Alterar);
  - e. **Cancelar** a ação clicando no botão “Cancelar”.
    - i. Os dados da tela são descartados;
    - ii. O fluxo é encerrado;
    - iii. O sistema vai para o “

- iv. Fluxo Básico”;
- f. **Salvar** as alterações
  - i. O Sistema verifica as restrições (E 01) (E 06) (E 07);
  - ii. Caso alguma **restrição tenha ocorrido**:
    - (1) O Sistema mostra a mensagem de erro correspondente;
    - (2) O Sistema posiciona o cursor no primeiro campo a ser informado novamente pelo usuário.
  - iii. Caso **não tenham ocorrido restrições**
    - (1) O Sistema grava as modificações da Diretriz Estratégica e das suas Ações no banco de dados;
    - (2) O Sistema mostra a mensagem de sucesso (O 01).
    - (3) O Sistema vai para o “
    - (4) Fluxo Básico”;
- 6. O fluxo é encerrado.

**(A 03) Excluir Diretriz Estratégica**

1. O Sistema exibe a mensagem de confirmação de exclusão (O 02);
2. O Usuário confirma a exclusão;
3. O Sistema verifica as restrições (E 04);
4. Caso **existam restrições**:
  - a. O Sistema apresenta a mensagem de exceção;
5. Caso **não existam restrições**:
  - a. O Sistema exclui a Diretriz Estratégica;
  - b. O Sistema mostra a mensagem de sucesso (O 03);
  - c. O Sistema vai para o “
  - d. Fluxo Básico”;
6. O fluxo é encerrado;

**(A 04) Incluir Diretriz Estratégica**

1. O Sistema monta o menu da página de acordo com as permissões do usuário logado.
2. O Sistema exibe a Tela “*Incluir*” contendo:
  - a. Campos da Diretriz Estratégica

<b>Campo</b>	<b>Descrição</b>	<b>Editável?</b>	<b>Obrigatório?</b>	<b>Máscara</b>
Nome da Diretriz Estratégica	Nome da Diretriz Estratégica. (E 01) (E 06)	Sim	Sim	Não se aplica
Situação	Situação da Diretriz Estratégica (Ativa ou Inativa). (E 01)	Sim	Sim	Não se aplica

- b. Lista de Ações Estratégicas de Curto Prazo da Diretriz, ordenadas pelo campo “Nome”. A Lista está inicialmente vazia. Cada ação estratégica possui os seguintes campos:

<b>Campo</b>	<b>Descrição</b>	<b>Editável?</b>	<b>Obrigatório?</b>	<b>Máscara</b>
Nome da Ação Estratégica	Nome da Ação Estratégica. (E 01) (E 07)	Sim	Sim	Não se aplica
Situação	Situação da Ação Estratégica (Ativa ou Concluída) (E 01)	Sim	Sim	Não se aplica

3. O Usuário informa os dados da diretriz estratégica;
4. O Usuário informa os dados da ação estratégica;
5. O Usuário pode **escolher**:
  - a. **Voltar à “Página Principal”** do sistema clicando sobre o respectivo link na linha de navegação do sistema.
    - i. O Caso de Uso é encerrado;
  - b. **Voltar à Tela “Diretriz Estratégica – Listar”** clicando no respectivo link na linha de navegação do sistema;
    - i. O Sistema vai para o “Fluxo Básico”;
    - ii. O fluxo é encerrado;
  - c. **Excluir uma Ação**, clicando em “*excluir*” para a ação.
    - i. Aciona o *Fluxo Alternativo “Excluir Ação”* (A 05);
    - ii. O sistema permanece no fluxo Incluir;
  - d. **Incluir uma Ação**, clicando em “*incluir*” para a ação.
    - i. Aciona o *Fluxo Alternativo “Incluir Ação”* (A 06);
    - ii. O Sistema permanece no fluxo Incluir;
  - e. **Cancelar** a ação clicando no botão “Cancelar”.
    - i. Os dados da tela são descartados;
    - ii. O fluxo é encerrado;
    - iii. O Sistema vai para o “Fluxo Básico”;
  - f. **Salvar** as alterações
    - i. O Sistema verifica as restrições (E 01) (E 06) (E 07);
    - ii. Caso alguma **restrição tenha ocorrido**:
      - (1) O Sistema mostra a mensagem de erro correspondente;
      - (2) O Sistema posiciona o cursor no primeiro campo a ser informado novamente pelo usuário.
    - iii. **Caso não tenham ocorrido restrições**
      - (1) O Sistema grava as modificações da Diretriz Estratégica e das suas Ações no banco de dados;
      - (2) O Sistema mostra a mensagem de sucesso (O 04).
      - (3) O Sistema vai para o início do *Fluxo “Incluir Diretriz Estratégica”* (A 04) para a inclusão de um novo registro;
6. O fluxo é encerrado

#### **(A 05) Excluir Ação**

1. O Sistema exibe a mensagem de confirmação de exclusão (O 02);

2. O Usuário confirma a exclusão;
3. O Sistema verifica as restrições (E 05);
4. Caso **existam restrições**:
  - a. O Sistema apresenta a mensagem de exceção;
5. Caso **não existam restrições**:
  - a. O Sistema exclui a Ação da Diretriz Estratégica da Tela (Observação: a exclusão da ação não é gravada no banco de dados neste momento);
6. O fluxo é encerrado;

**(A 06) Incluir Ação**

1. O Sistema verifica as restrições (E 01);
2. Caso **existam restrições**:
  - a. O Sistema apresenta a mensagem de exceção;
3. Caso **não existam restrições**:
  - a. O Sistema inclui a Ação da Diretriz Estratégica na Tela (Observação: a inclusão da ação não é gravada no banco de dados neste momento);
    - i. A Situação da Ação deve ser “*Ativa*”;
  - b. O Sistema acrescenta uma linha na tabela com a ação incluída;
  - c. O Sistema prepara a última linha da tabela de Ações para uma nova inclusão;
4. O fluxo é encerrado;

**Fluxos de Exceções**

**(E 01) Campo Obrigatório**

1. Se um campo obrigatório não foi preenchido ou selecionado, o sistema exibirá a mensagem “*O preenchimento do campo (nome do campo) é obrigatório*”.

**(E 02) Acesso à Funcionalidade fora do Período Permitido**

1. O Usuário tentou acessar esta Funcionalidade fora do Período Permitido (o período é definido no caso de uso “Alterar Prazos” (CDU\_PLA\_002)).
2. O Sistema exibe a mensagem “*Esta Funcionalidade não pode ser utilizada hoje. O período permitido para Editar as Diretrizes Estratégicas é de {Data Início} até {Data Fim}*”, sendo que *Data Início* e *Data Fim* devem ser recuperadas do banco de dados para esta funcionalidade.

**(E 03) A consulta não retorna nenhum valor**

1. O Sistema exibe a mensagem “A consulta não retornou nenhum registro”.

**(E 04) Tentativa de Exclusão de uma Diretriz Estratégica Referenciada em outros pontos do Sistema**

1. O Sistema deve verificar se alguma Ação Estratégica da Diretriz é referenciada em algum ponto do sistema (por exemplo, Objetivos de Planos Básicos Estruturantes ou Setoriais podem referenciar uma ação de diretriz estratégica);
2. O Sistema exibe a mensagem “Esta Diretriz Estratégica não pode ser excluída, pois pelo menos uma de suas Ações Estratégicas de Curto Prazo está sendo utilizada”.

**(E 05) Tentativa de Exclusão de uma Ação Estratégica Referenciada em outros pontos do Sistema**

1. O Sistema deve verificar se a Ação Estratégica é referenciada em algum ponto



do sistema (por exemplo, Objetivos de Planos Básicos Estruturantes ou Setoriais podem referenciar uma ação de diretriz estratégica);

2. O Sistema exibe a mensagem “Esta Ação Estratégica não pode ser excluída, pois está sendo utilizada”.

**(E 06) Nome de Diretriz Estratégica repetido**

1. O Usuário tentou incluir um nome de Diretriz Estratégica que já existe.
2. O Sistema emite a mensagem “Não foi possível Salvar. Diretriz Estratégica já cadastrada no Sistema”.

**(E 07) Nome de Ação Estratégica repetido**

1. O Usuário tentou incluir um nome de Ação Estratégico que já existe em uma mesma Diretriz Estratégica.
2. O Sistema emite a mensagem “Não foi possível salvar. Existem Ações Estratégicas repetidas nesta Diretriz”.

## Regras de Negócio

**(RN 01) Regras de Negócio do Filtro da Tela “Diretriz Estratégica – Listar”**

(RN 01.1) **O Estado do Filtro deve ser mantido na navegação pelas telas da Diretriz Estratégica.** O Filtro só é reiniciado (nenhuma opção selecionada) quando o usuário acessar a função “Diretriz Estratégica - Listar” ou “Diretriz Estratégica - Incluir” do menu do Sistema ou quando o próprio usuário limpar as opções de Filtro e escolher a opção Filtrar.

(RN 01.2) **O campo ‘Nome’ no Filtro** deve permitir a pesquisa por parte das descrições utilizando o caracter \* no início dos termos de pesquisa, no fim ou em ambos. Por exemplo: o usuário pode pesquisar por ‘Diretriz\*’ para buscar dos os registros que comecem com ‘Diretriz’;

## Observações

**(O 01) Mensagem de Alteração Realizada com Sucesso**

- O sistema exibe em uma caixa de diálogo a mensagem “*Diretriz Estratégica Alterada com Sucesso*”.

**(O 02) Mensagem de Confirmação de Exclusão**

- O sistema exibe em uma caixa de diálogo a mensagem “*Tem certeza que deseja excluir este registro?*”.

**(O 03) Mensagem de Exclusão Realizada com Sucesso**

- O sistema exibe em uma caixa de diálogo a mensagem “*Diretriz Estratégica Excluída com Sucesso*”.

**(O 04) Mensagem de Inclusão Realizada com Sucesso**

- O sistema exibe em uma caixa de diálogo a mensagem “*Diretriz Estratégica Incluída com Sucesso*”.

### Controle de Versão

Versão	Data	Descrição da Alteração	Motivo	Autor
0.1	13/03/2006	Versão Inicial	-	Vera Lúcia

## B – CASO DE USO DO SGQ

Versão	Data	Descrição da Alteração	Autor
001	11/12/2006	Criação do Caso de Uso	Vera Lúcia

### CDU- 002 - Desativação de Unidades Primárias

- **Descrição Sucinta**

Esse caso de uso permite ao gestor do sistema a desativação de Unidades Primárias.

- **Atores**

Entidade Pública

- **Pré-Condições**

O Usuário deve estar logado no Sistema de Gestão de Qualidade – SGQ

O Usuário deve ter perfil para acesso ao CDU-002 – Desativação de Unidades Primárias

- **Pós-Condições**

- **Fluxo Básico**

- O usuário seleciona a opção “**Desativação de Unidades Primárias**”
- O sistema exibe a tela “Desativação de Unidades Primárias” onde o usuário selecionar o seguinte campo:
  - Empresas
    - O ComboBox em questão, será preenchido somente com as empresas já cadastradas;
- O Ator tem a opção de “**Limpar campos**” para limpar a caixa de seleção e ir para o fluxo básico letra (a);
- O Ator tem a opção de “**Confirmar**” para gravar os dados do cadastro (O 01)(RN 01);
- O Ator tem a opção de “**Consulta Unidades Primárias**” (A 01) para cada item da lista (Lista Unidades Primárias);
- O Ator tem a opção de “**Desativar Unidades Primárias**” (A 02) para cada item da lista (Lista Unidades Primárias);
- O caso de uso é encerrado.

## Fluxos Alternativos

### (A 01) Consultar Unidades Primárias

1. O Ator seleciona opção de filtro “Empresa”
2. O Ator informa o dado a ser filtrado;
3. O Ator seleciona a opção “Filtrar” (E 01) (E 02) (E 03) (O 03)
4. O Sistema exibe a lista das Empresas filtradas na tela;
5. O Sistema vai para o “
6. Fluxo Básico”;
7. O fluxo é encerrado

### (A 02) Desativar Unidades Primárias

1. O sistema exibe a tela com os seguintes campos:
  - a. Empresas;
  - b. Código da Unidade Primária;
  - c. Opção Desativar;
  - d. Campo Justificativa (O 04) (RN 02)
2. O Ator tem a opção de “Confirmar” para gravar os dados do cadastro (E 01);
3. O sistema realiza desativação das unidades primárias;
4. O sistema exibe a mensagem de Desativação da empresa com Sucesso (O 02);
5. O Ator tem a opção de “Limpar campos” para limpar o check-list e caixa de seleção e retorna para fluxo básico.

## Fluxos de Exceções

### (E 01) Se nenhuma opção de consulta for selecionada

3. O sistema exibe a mensagem *“Selecione umas das opções de filtro para realizar a consulta”*.

### (E 02) Se a consulta não retornar nenhum valor

4. O sistema exibe a mensagem *“A consulta não retornou nenhum registro.”* O sistema limpa os campos de consulta e o caso de uso é recommçado.

### (E 03) Se não for informada pelo menos uma Empresa

5. O sistema não efetua a operação e exibe a mensagem *“Informe pelo menos uma Empresa, o campo é obrigatório”*.

## Observações

- (O 01) (RN 01) Regras de negócio da tela Desativação de Unidades Primárias - Confirmar”: Na confirmação, o Ator, obrigatoriamente, deverá informar pelo menos uma empresa conforme (E 03)

**(O 02) Mensagem de Desativação Realizada com Sucesso**

Para a Desativação de uma empresa, o sistema exibe, em uma caixa de diálogo, a mensagem **“Desativação da Unidade Primárias com Sucesso”**.

**(O 03) Consulta por Empresa**

A pesquisa poderá ser feita por qualquer palavra, ou parte dela, que for preenchida pelo usuário.

**(O 04) (RN 02) Campo Justificativa**

Ao selecionar a opção desativar, obrigatoriamente o campo Justificativa será habilitado e terá que ser preenchido, antes de selecionar o botão confirmar, caso contrário, o sistema exibirá uma mensagem “O campo justificativa deve ser preenchido”;

## C – CASO DE TESTE DO MGP

ESPECIFICAÇÃO DE CASO DE TESTE		Código: CTS 01
<b>Identificação:</b>	CTS 01– Diretriz Estratégica	
<b>Cliente:</b>	Órgão Governamental	
<b>Projeto:</b>	Módulo de Gestão de Planejamento – MGP	
<b>Versão do CTS:</b>	0.1	
<b>Aceite:</b>		

### DIRETRIZ ESTRATÉGICA

#### Descrição do Caso de Teste:

- Este caso de teste permite testar as funções do caso de uso CDU\_PLA\_001 – Diretriz Estratégica

#### Versão do CDU:

- Baseado na Versão 1.2 do Caso de Uso.

#### Tipo de abordagem efetuada nos testes:

- Teste de caixa preta, ou seja, testes funcionais realizados do ponto de vista do usuário do SIGA.

## ROTINAS DE TESTES

### (RT 01) Acesso a “Diretriz Estratégica - Listar”

**Fluxo do Caso de Uso:** Fluxo Básico

**Pré-Condição:** Usuário deve estar logado no Sistema

Cenário	Passos a Executar	Item	Resultado Esperado	Verificado?
<b>CN 01</b>	1- Acesse a Função Diretriz Estratégica - Listar com um usuário que possua permissão de acesso.	<b>01</b>	O sistema exibe a tela 'Diretriz Estratégica – Listar' com as opções de filtro por: <ul style="list-style-type: none"> <li>▪ Nome</li> <li>▪ Situação</li> </ul>	
		<b>02</b>	O campo “Situação” deve ter as opções: <ul style="list-style-type: none"> <li>▪ Ativa</li> <li>▪ Inativa</li> </ul>	
		<b>03</b>	Em seguida exibe a lista de Diretrizes Estratégicas com os campos: <ul style="list-style-type: none"> <li>▪ Nome</li> <li>▪ Situação</li> </ul>	
		<b>04</b>	A lista de Diretrizes deverá estar ordenada pelo campo Nome.	
<b>CN 02</b>	1- Acesse a função Diretriz Estratégica com um usuário que <b>não possua</b> permissão de acesso.	<b>01</b>	O sistema não permite o acesso.	

## (RT 02) Filtrar Diretriz Estratégica

**Fluxo do Caso de Uso:** A 01 – Filtrar

**Pré-Condição:** Acesse a Função Diretriz Estratégica com um usuário que possua permissão de acesso.

Cenário	Passos a Executar	Item	Resultado Esperado	Verificado?
CN 01	1- Informe os dados dos campos do Filtro conforme os Conjuntos de Valores de Teste abaixo	01	Ver conjuntos de valores de teste abaixo	
	2- Para cada conjunto de valores de teste, acione a opção 'Filtrar'.			

### Conjuntos de Valores de Teste

No.	Campos	Valor	Item	Resultado Esperado	Verificado?
1	Nome	Digite o nome pelo qual deseja fazer a filtragem	01	A lista de Diretrizes Estratégicas recuperadas do banco de dados exibe todos as diretrizes condizentes com o nome que foi informado.	
	Situação	Deixe em “ <b>Selecione</b> ”			

No.	Campos	Valor	Item	Resultado Esperado	Verificado?
2	Nome	Não informe o Nome	01	O sistema deverá trazer todas as Diretrizes Estratégicas existentes no sistema.	
	Situação	Deixe em “ <b>Selecione</b> ”			

No.	Campos	Valor	Item	Resultado Esperado	Verificado?
3	Nome	Informe o nome, <i>somente com os primeiros caracteres seguidos de asterisco (*)</i>  <b>Ex: Nome: Diretriz</b> Informe no filtro: <b>Dir*</b>	01	O sistema deverá trazer todos as Diretrizes que comecem com os caracteres <b>Dir</b> <b>(RN 01.2)</b>	
	Situação	Deixe em “ <b>Selecione</b> ”			

No.	Campos	Valor	Item	Resultado Esperado	Verificado?
4	Nome	Informe o nome, <i>com os últimos caracteres antecidos de asterisco (*)</i>  <b>Ex: Nome: Diretriz Estratégica de Reestruturação</b> Informe no filtro: <b>*Reestruturação</b>	01	O sistema deverá trazer todos as Diretrizes que terminem com os caracteres <b>Reestruturação</b> <b>(RN 01.2)</b>	
	Situação	Deixe em “ <b>Selecione</b> ”			

No.	Campos	Valor	Item	Resultado Esperado	Verificado?
5	Nome	Informe o nome, <i>com a descrição completa do nome, sem o (*)</i>  <b>Ex: Nome: Diretriz Estratégica de Reestruturação</b>  Informe no filtro: <b>Diretriz Estratégica de Reestruturação</b>	01	O sistema deverá trazer a Diretriz Estratégica <b>Diretriz Estratégica de Reestruturação</b> <b>(RN 01.2)</b>	
	Situação	Deixe em “ <b>Selecione</b> ”			



No.	Campos	Valor	Item	Resultado Esperado	Verificado?
6	Nome	Informe o nome, <i>pela descrição informada precedendo e sucedendo o ( * )</i>  <b>Ex: Nome: Diretriz Estratégica de Reestruturação</b> Informe no filtro: <b>*Estratégica*</b>	01	O sistema deverá trazer todos as Diretrizes que possuam os caracteres <b>Estratégica (RN 01.2)</b>	
	Situação	Deixe em <b>“Seleção”</b>			

No.	Campos	Valor	Item	Resultado Esperado	Verificado?
7	Nome	Deixe em Branco	01	O sistema deverá trazer todos as Diretrizes Estratégicas que estejam com a situação Ativa.	
	Situação	Selecione <b>Ativa</b>			

No.	Campos	Valor	Item	Resultado Esperado	Verificado?
8	Nome	Deixe em Branco	01	O sistema deverá trazer todos as Diretrizes Estratégicas que estejam com a situação Inativa	
	Situação	Selecione <b>Inativa</b>			

No.	Campos	Valor	Item	Resultado Esperado	Verificado?
9	Nome	Deixe em Branco	01	O sistema deverá trazer todas as Diretrizes Estratégicas com todas as situações existentes no sistema.	
	Situação	Não informe a Situação			

No.	Campos	Valor	Item	Resultado Esperado	Verificado?
10	Nome	Digite “asfdffsf”	01	O sistema exibe a <i>mensagem ‘A consulta não retornou nenhum registro ‘</i> , pois nenhum nome termina com ‘asfdffsf. (E 03)	
	Situação	Selecione <b>Ativa</b>			

No.	Campos	Valor	Item	Resultado Esperado	Verificado?
11	Nome	Digite <b>Dir*</b>	01	O sistema deverá trazer todas as Diretrizes Estratégicas que comecem com ‘Dir’ e que estejam Ativas.	
	Situação	Selecione <b>Ativa</b>			

Cenário	Passos a Executar	Item	Resultado Esperado	Verificado?
CN 02	1-Execute o <b>RT02 - CN 01</b> com o <b>Conjunto de Valores de Teste 9</b> . 2-O número de registros retornados deve ser maior que 10 para que o teste seja válido. 3- Teste o mecanismo de paginação.	01	Verifique se os controles de paginação estão funcionando corretamente.	
CN 03	1-Execute a <b>RT02 – (CN 01)</b> com o <b>Conjunto de Valores de Teste 11</b> . 2- Clique em “ <i>alterar</i> ” para algum registro da tela. 3- Em seguida clique no link “Diretriz Estratégica - Listar” para voltar à tela anterior.	01	As opções selecionadas no filtro anteriormente deve ser mantidas, no mesmo estado em que se encontrava anteriormente, ou seja, tudo o que foi selecionado no filtro deve estar na mesma situação. <b>(RN 01.1)</b>	

(RT 03) Alterar Diretriz Estratégica

Fluxo do Caso de Uso: A 02 – Alterar

Pré-Condição: Acessar a funcionalidade de Diretriz Estratégica – Listar

Cenário	Passos a Executar	Item	Resultado Esperado	Verificado?
CN 01	1 – Clique em “ <i>alterar</i> ” para algum registro	01	O sistema exibe a tela de Alterar as Diretrizes Estratégicas, exibindo os seguintes campos: <ul style="list-style-type: none"><li>▪ Nome</li><li>▪ Situação</li></ul>	
		02	Lista de Ações Estratégicas de Curto Prazo com os seguintes campos: <ul style="list-style-type: none"><li>▪ Nome</li><li>▪ Situação</li></ul>	
		03	A Lista de Ações Estratégicas deve estar ordenada pelo campo Nome	
CN 02	1 – Clique em “ <i>alterar</i> ” para algum registro 2 – Altere os campos conforme os Conjuntos de Valores de Teste abaixo	01	Ver conjuntos de valores de teste abaixo.	

Conjuntos de Valores de Teste do *Alterar Diretriz Estratégica*

No.	Campo	Valor	Item	Resultado Esperado	Verificado?
1	Nome da Diretriz	Limpe o campo Clique no botão “ <b>Salvar</b> ”	01	O sistema exibe a mensagem: “ <i>O preenchimento do campo Nome é obrigatório</i> ” (E 01)	

No.	Campo	Valor	Item	Resultado Esperado	Verificado?
2	Nome da Diretriz	Informe um nome de Diretriz Estratégica que já esteja cadastrada no sistema <b>Ex:</b> Se já existir: <i>Diretriz Estratégica</i> , Digite novamente: <i>Diretriz Estratégica</i>  Não altere os outros campos. Clique no botão “ <b>Salvar</b> ”	01	O sistema exibe a mensagem: “ <i>Não foi possível salvar. Diretriz Estratégica já cadastrada no sistema</i> ” (E 06)	

No.	Campo	Valor	Item	Resultado Esperado	Verificado?
3	Nome da Diretriz	Não altere o valor	01	O sistema exibe a mensagem: “ <i>O preenchimento do campo Situação é obrigatório</i> ” (E 01)	
	Situação da Diretriz	Deixe o campo Situação na opção “ <b>Selecione</b> ”			

#### Conjuntos de Valores de Teste do *Alterar Ações Estratégicas de Curto Prazo*

No.	Campo	Valor	Item	Resultado Esperado	Verificado?
4	Nome da Ação Estratégica	Limpe o campo de uma Ação Estratégica  Não Altere os outros campos. Clique no botão “ <b>Salvar</b> ”	01	O sistema exibe a mensagem: “ <i>O preenchimento do campo Nome é obrigatório</i> ” (E 01)	

No.	Campo	Valor	Item	Resultado Esperado	Verificado?
5	Nome da Ação Estratégica	<p>Informe um nome de uma Ação Estratégica que já esteja cadastrada na Diretriz Estratégica selecionada.</p> <p><b>Ex:</b> Se já existir: <i>Criar Brigadas</i></p> <p><b>Ex:</b> Digite novamente: <i>Criar Brigadas</i></p> <p>Não altere os outros campos.</p> <p>Clique no botão <b>“Salvar”</b></p>	01	O sistema exibe a mensagem: <b>“Não foi possível salvar. Existem Ações Estratégicas repetidas nesta Diretriz” (E 07)</b>	

No.	Campo	Valor	Item	Resultado Esperado	Verificado?
6	Situação da Ação Estratégica	<p>Altere a Situação de uma ação para <b>“Concluída”</b></p>	01	O sistema exibe a mensagem <b>“Diretriz Estratégica Alterada com Sucesso”</b> . (O 01)	
		<p>Não altere os outros campos.</p> <p>Clique no botão <b>“Salvar”</b></p>	02	Verifique que os dados foram alterados.	

No.	Campo	Valor	Item	Resultado Esperado	Verificado?
7	Nome da Diretriz Estratégica	Diretriz Estratégica de Estruturação	01	O sistema exibe a mensagem <b>“Diretriz Estratégica Alterada com Sucesso”</b> . (O 01)	
	Situação da Diretriz Estratégica	Selecione <b>Ativa</b>			
	Nome da Ação Estratégica	Digite <b>Reestruturação</b> para uma Ação	02	Verifique que os dados foram alterados.	
	Situação da Ação Estratégica	Selecione <b>Ativa</b> para a Ação alterada acima			

Cenário	Passos a Executar	Item	Resultado Esperado	Verificado?
CN 03	1 – Clique em <b>“alterar”</b> para algum registro 2 – Faça qualquer alteração nos registros. 3 – Clique no botão <b>“Cancelar”</b>	01	O sistema retorna para a página <b>“Diretriz Estratégica - Listar”</b>	
		02	Verifique que nenhum dado do registro foi alterado	

(RT 04) Rotina do Excluir Diretriz Estratégica

Fluxo do Caso de Uso: A 03 – Excluir Diretriz Estratégica

Pré-Condição: Acessar a funcionalidade de Diretriz Estratégica – Listar

Cenário	Passos a Executar	Item	Resultado Esperado	Verificado?
CN 01	1 – Clique em <b>“excluir”</b> para alguma Diretriz Estratégica.	01	O sistema exibe a mensagem: <b>“Tem certeza que deseja excluir esse registro?”</b> . (O 02)	
CN 02	1 – Clique em <b>“excluir”</b> para alguma Diretriz Estratégica que não esteja sendo referenciada no Sistema. 2 – Após o Sistema exibir a mensagem <b>“Tem certeza que deseja excluir esse registro?”</b> , clique em <b>“OK”</b> .	01	O sistema exibe a mensagem: <b>“Diretriz Estratégica excluída com sucesso”</b> . (O03)	
CN 03	1 – Clique em <b>“excluir”</b> para alguma Diretriz Estratégica que esteja sendo referenciada no Sistema ( <b>procure uma diretriz que tenha uma ação referenciada em um Plano Estratégico</b> ) 2 – Após o Sistema exibir a mensagem <b>“Tem certeza que deseja excluir esse registro?”</b> , clique em <b>“OK”</b> .	01	O sistema exibe a mensagem: <b>“Esta Diretriz Estratégica não pode ser excluída, pois pelo menos uma de suas Ações Estratégicas de Curto Prazo está sendo utilizada”</b> (E 04)	
CN 04	1 – Clique em <b>“excluir”</b> para alguma Diretriz Estratégica. 2 – Após o Sistema exibir a mensagem <b>“Tem certeza que deseja excluir esse registro?”</b> , clique em <b>“Cancelar”</b> .	01	Observe que o registro não foi excluído e continua na mesma situação em que estava anteriormente.	

**(RT 05) Rotina do Incluir Diretriz Estratégica**

**Fluxo do Caso de Uso:** A 04 – Incluir Diretriz Estratégica

**Pré-Condição:** Usuário deve estar logado no Sistema

Cenário	Passos a Executar	Item	Resultado Esperado	Verificado?
CN 01	1- Acesse a Função Incluir Diretriz Estratégica com um usuário que possua permissão de acesso.	01	O sistema exibe a tela Diretriz Estratégica – Incluir com os seguintes campos: <ul style="list-style-type: none"> <li>▪ Nome</li> <li>▪ Situação</li> </ul>	
		02	Exibe a Tabela para inclusão das Ações Estratégicas de Curto Prazo da Diretriz, contendo os campos: <ul style="list-style-type: none"> <li>▪ Nome</li> <li>▪ Situação</li> </ul>	
CN 02	1- Acesse a função Diretriz Estratégica com um usuário que <b>não possua</b> permissão de acesso.	01	O sistema não permite o acesso.	
CN 03	1- Informe os dados dos campos do Incluir Diretriz Estratégica conforme os Conjuntos de Valores de Teste abaixo 2- Para cada conjunto de valores de teste, acione o botão “Salvar”	01	Ver conjuntos de valores de teste abaixo	

**Conjunto de Valores de Teste**

No.	Campo	Valor	Item	Resultado Esperado	Verificado?
1	Nome da Diretriz	Deixe o campo em Branco Clique em “Salvar”	01	O sistema exibe a mensagem: <i>“O preenchimento do campo Nome é obrigatório” (E 01)</i>	



No.	Campo	Valor	Item	Resultado Esperado	Verificado?
2	Nome da Diretriz	Informe um nome de uma nova Diretriz Estratégica	01	O sistema exibe a mensagem: <i>“O preenchimento do campo Situação é obrigatório” (E 01)</i>	
	Situação da Diretriz	Não preencha o campo “Situação” Clique no botão <b>“Salvar”</b>			

No.	Campo	Valor	Item	Resultado Esperado	Verificado?
3	Nome da Diretriz	Informe um nome de uma Diretriz Estratégica que já esteja cadastrada  <b>Ex:</b> Se já existir: <i><b>Diretriz Estratégica,</b></i>  Digite novamente: <i><b>Diretriz Estratégica</b></i>	01	O sistema exibe a mensagem: <i>“Não foi possível salvar. Diretriz Estratégica já cadastrada no sistema” (E 06).</i>	
	Situação da Diretriz	Selecione <b>“Ativa”</b>			
	Nome da Ação Estratégica	Informe <b>Nome da Ação Estratégica</b>			
	Clique em <b>“Incluir”</b> Clique em <b>“Salvar”</b>				

No.	Campo	Valor	Item	Resultado Esperado	Verificado?
4	Nome da Ação Estratégica	Deixe o campo em Branco	01	O sistema exibe a mensagem: <b>“O preenchimento do campo Nome da Ação Estratégica é obrigatório” (E 01)</b>	
	Clique no botão <b>“Incluir”</b>				

No.	Campo	Valor	Item	Resultado Esperado	Verificado?
5	Nome da Diretriz	Informe um nome de uma Diretriz Estratégica que não esteja cadastrada no Sistema  <b>Ex: Diretriz Estratégica de Reestruturação</b>	01	O sistema exibe a mensagem: <b>“Não foi possível salvar. Existem Ações Estratégicas repetidas nesta Diretriz” (E 07)</b>	
	Situação da Diretriz	Selecione <b>“Ativa”</b>			
	Nome da Ação Estratégica	Informe <b>Nome da Ação Estratégica</b>  Clique em <b>“Incluir”</b>			
	Nome da Ação Estratégica	Na nova linha em branco informe <b>Nome da Ação Estratégica</b>  Clique em <b>“Incluir”</b>			
	Clique em <b>“Salvar”</b>				

No.	Campo	Valor	Item	Resultado Esperado	Verificado?
6	Nome da Diretriz	Informe um nome de uma Diretriz Estratégica que não esteja cadastrada no Sistema  <b>Ex: Diretriz Estratégica de Reestruturação</b>	01	O sistema exibe a mensagem <i>“Diretriz Estratégica incluída com sucesso”</i> . (O 04)	
	Situação da Diretriz	Selecione <b>“Ativa”</b>			
	Nome da Ação Estratégica	Informe <b>Nome da Ação Estratégica</b>  Clique em <b>“Incluir”</b>	02	Verifique que os dados foram alterados.	
	Clique em <b>“Salvar”</b>				

Cenário	Passos a Executar	Item	Resultado Esperado	Verificado?
CN 04	1 – Preencha os campos da tela 2 – Pressione o botão <b>“Cancelar”</b>	01	Verifique que o Sistema não incluiu o registro	
		02	Verifique que o Sistema vai para a página <b>“Diretriz Estratégica - Listar”</b>	

(RT 06) Rotina do Excluir Ação Estratégica de Curto Prazo

Fluxo do Caso de Uso: A 05 – Excluir Ação

Pré-Condição: Usuário deve estar na tela de Alteração de uma Diretriz Estratégica

Cenário	Passos a Executar	Item	Resultado Esperado	Verificado?
CN 01	1 – Clique no link “ <i>excluir</i> ” para alguma Ação Estratégica	01	O sistema exibe a mensagem: “ <i>Tem certeza que deseja excluir esse registro?</i> ”. (O 02)	
CN 02	1 – Clique no link “ <i>excluir</i> ” para alguma Ação Estratégica que já esteja sendo referenciada no Sistema ( <b>procure uma ação esteja sendo utilizada em um Plano Estratégico</b> )  2 – Após o Sistema exibir a mensagem “ <i>Tem certeza que deseja excluir esse registro?</i> ”, clique em “OK”.	01	O sistema exibe a mensagem: “ <i>Está Ação Estratégica não pode ser excluída, pois está sendo utilizada</i> ” (E 05)	
CN 03	1 – Clique no link “ <i>excluir</i> ” para alguma Ação Estratégica que não esteja sendo referenciada no Sistema  2 – Após o Sistema exibir a mensagem “ <i>Tem certeza que deseja excluir esse registro?</i> ”, clique em “OK”.  3 – Clique em “ <b>Salvar</b> ”	01	Verifique que a Ação Excluída não está mais no Sistema.	

**(RT 07) Rotina do Incluir Ação Estratégica de Curto Prazo**

**Fluxo do Caso de Uso:** A 06 – Incluir Ação

**Pré-Condição:** Usuário deve estar na tela de Alteração de uma Diretriz Estratégica

Cenário	Passos a Executar	Item	Resultado Esperado	Verificado?
CN 01	1 – Inclua os registros de Ação Estratégica conforme os Conjuntos de Valores de Teste abaixo	01	Ver conjuntos de valores de teste abaixo.	

No.	Campo	Valor	Item	Resultado Esperado	Verificado?
1	Nome da Ação Estratégica	Informe o Nome de uma Ação Estratégica na última linha da tabela. Clique em <b>“Incluir”</b> .	01	O sistema inclui a Ação da Diretriz Estratégica.	
			02	A situação da Ação que foi inclusa deve ser igual a <b>“ativa”</b>	
			03	O sistema mostra as opções “alterar” e “excluir” para a Ação incluída.	
			04	Uma nova linha é disponibilizada para a inclusão de uma nova Ação	

**CONTROLE DE VERSÃO**

Versão	Data	Descrição da Alteração	Motivo	Autor
0.1	03/07/2006	Criação do Caso de Teste	-	Vera Lúcia

## D – MANUTENÇÃO CORRETIVA MGP

### SOLICITAÇÃO DE MANUTENÇÃO - SM Nr. 03/2007

<b>Responsável:</b> Analista de Requisito	<b>Fone/ramal:</b> Telefone do Analista de Teste / Testador
<b>Solicitante:</b> Analista de Teste / Testador	<input checked="" type="checkbox"/> <b>Corretiva</b> <input type="checkbox"/> <b>Evolutiva</b> <input type="checkbox"/> <b>Adaptativa</b>
<b>Manutenção:</b> <input type="checkbox"/> Banco <input checked="" type="checkbox"/> Sistema	<b>Sistema:</b> Órgão Governamental
<b>Versão do sistema:</b> 1.0	<b>Módulo:</b> Gestão de Planejamento - MGP
<b>Caso de Teste:</b> CTS_001_PLA –Diretriz Estratégica	
<b>Rotina do teste:</b> Todas as Rotinas do Caso de Teste	<b>Prioridade:</b> Alta

#### DADOS DA DEMANDA

#### Descrição do erro – Severidade: Erro Simples

- O tamanho do campo “Filtrar por Nome” está menor do que o especificado (164 caracteres)

The screenshot shows a web application interface for a list of records. At the top, there is a navigation bar with 'Página Principal >> Diretriz Estratégica - Listar'. Below it is a search filter with the text 'Filtrar por nome: Diretriz Est' and a dropdown menu for 'Situação: Selezione'. A table with two columns, 'Nome' and 'situação', contains several rows of data. A red arrow points to the search field, highlighting the issue mentioned in the text. The footer of the interface shows 'Página atual: 1 / 1' and 'Registros por página: 10'.

#### Tela 1 – Tamanho de campo Correção Corretiva Evolutiva

#### Parecer da Contratada:

#### Autorização do Chefe da Seção

\_\_\_\_\_ Data: \_\_\_/\_\_\_/\_\_\_

#### ATENDIMENTO

Data entrega p/ equipe de desenvolvimento: 19/7/07 | Hora entrega p/ equipe: 10:00

Analista Responsável: Nome do Analista de Teste

Data Conclusão: 20/7/07 | Hora conclusão: 19:00

## SOLICITAÇÃO DE MANUTENÇÃO - SM Nr. 02/2007

<b>Responsável:</b> Analista de Requisito	<b>Fone/ramal:</b> Telefone do Analista de Teste / Testador
<b>Solicitante:</b> Analista de Teste / Testador	<input checked="" type="checkbox"/> <b>Corretiva</b> <input type="checkbox"/> <b>Evolutiva</b> <input type="checkbox"/> <b>Adaptativa</b>
<b>Manutenção:</b> <input type="checkbox"/> Banco <input checked="" type="checkbox"/> Sistema	<b>Sistema:</b> Órgão Governamental
<b>Versão do sistema:</b> 1.0	<b>Módulo:</b> Gestão de Planejamento - MGP
<b>Caso de Teste:</b> CTS_001_PLA –Diretriz Estratégica	
<b>Rotina do teste:</b> Todas as Rotinas do Caso de Teste	<b>Prioridade:</b> Alta

### DADOS DA DEMANDA

**Descrição do erro – Severidade: Erro Crítico**

**Falha alterar uma Diretriz Estratégica**

Descrição do Erro:

1. Acesse Listar Diretriz Estratégica
2. Clique em Alterar para alguma Diretriz
3. Limpe o “Nome de Uma Ação Estratégica”
4. Clique em Salvar

**Erro**

O Sistema apresentou o erro: “Falha ao tentar iniciar uma transação no Hibernate”; Observe a figura abaixo.



**Tela 6 – Erro de filtro**

**Correção  Corretiva  Evolutiva  Adaptativa**

**Resultado Esperado**

O Sistema deveria apresentar a mensagem “O preenchimento do campo Nome da Ação Estratégica é obrigatório”

**Parecer da Contratada:**

**Autorização do Chefe da Seção**

\_\_\_\_\_ **Data:** \_\_\_\_/\_\_\_\_/\_\_\_\_

ATENDIMENTO	
Data entrega p/ equipe de desenvolvimento: 1/6/07	Hora entrega p/ equipe: 19:21
Analista Responsável: Nome do Analista de Teste	
Data Conclusão: 29/6/07	Hora conclusão: 11:25

## E – MANUTENÇÃO EVOLUTIVA MGP

### SOLICITAÇÃO DE MANUTENÇÃO - SM Nº 01/2007

<b>Responsável: Analista de Requisito</b>	<b>Fone/ramal:</b>
<b>Solicitante: Usuário / Cliente</b>	<input type="checkbox"/> Corretiva <input checked="" type="checkbox"/> Evolutiva <input type="checkbox"/> Adaptativa
<b>Manutenção: <input checked="" type="checkbox"/> Banco <input checked="" type="checkbox"/> Sistema</b>	<b>Sistema: Órgão Governamental</b>
<b>Versão do sistema: 2.0</b>	<b>Módulo: Gestão de Planejamento - MGP</b>
<b>Rotina do teste: Não se aplica</b>	<b>Prioridade: Alta</b>

#### DADOS DA DEMANDA

##### Descrição da Solicitação Evolutiva Manutenção Evolutiva - Complexidade Alta

- Retirar o campo Gerente do Objetivo Intermediário do Plano Básico.

##### Impacto Previsto:

Os seguintes casos de uso serão afetados pela modificação:

- CDU 12 – Elaborar Plano Básico
  - Fluxo de Inclusão de Alteração de Objetivo Intermediário
- CDU 15 – Detalhar Plano Básico
  - Fluxo de Detalhamento do Objetivo Intermediário
- CDU 17 – Imprimir Plano Básico
  - Impressão do Objetivo Intermediário
- CDU 23 – Finalizar Elaboração de Plano Básico
  - Consistência sobre o cargo do Gerente do Objetivo Intermediário será excluída
- CDU 25 – Manter Justificativa de Plano Básico
  - Fluxo de Justificativa do Objetivo Intermediário
- CDU 27 – Relatório de Avaliação de Execução de Plano Básico
  - Impressão da Avaliação do Objetivo Intermediário
- CDU 30 – Relatório de Gerentes/Coordenadores em Planos Básicos
  - Impressão do Gerente de Objetivo Intermediário

**Data: 05/03/2007**

**Hora: 18:26**

##### Parecer da Contratada:

##### Autorização do Chefe da Seção

**Usuário/cliente**

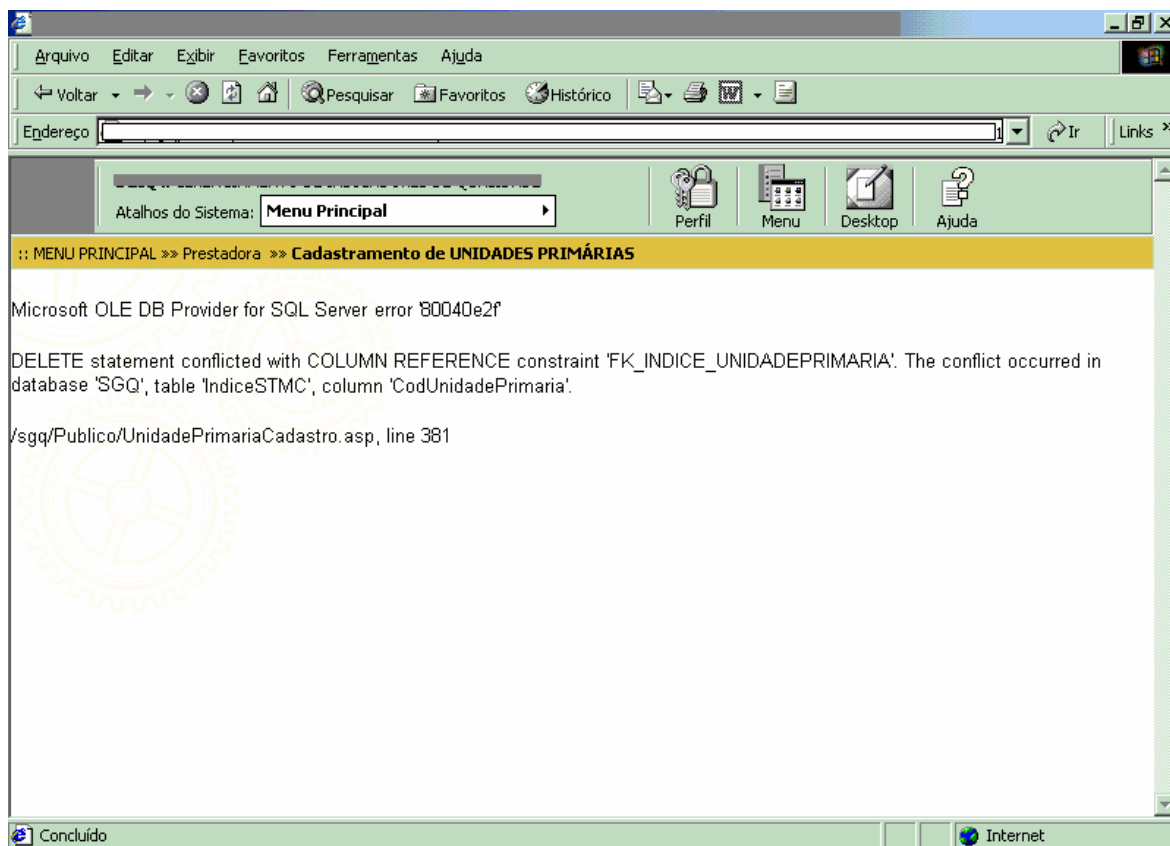
**Data: 05/03/2007**

ATENDIMENTO	
Data entrega p/ equipe de desenvolvimento: 06/03/2007	Hora entrega p/ equipe: 09:00
Analista Responsável: Nome do Requisito	
Data Conclusão: 08/05/2007	Hora conclusão: 11:19



## F – MANUTENÇÃO CORRETIVA DO SGQ

Mensagem de usuário relatando erro encontrado na utilização do SGQ na tela: Desativação de Unidades Primárias.



Solicitação: 7/1/2007 8h43

Efetivação: 7/1/2007 17h54

Complexidade da Manutenção: baixa.

## **G – ARTIGOS PUBLICADOS**

