

**BABILLA BORINE D'ANGELO**

**Análise e sugestão de melhoria do processo de controle de qualidade de  
software**

**São Paulo  
(2015)**

**BABILLA BORINE D'ANGELO**

**Análise e sugestão de melhoria do processo de controle de qualidade de software**

Monografia apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do certificado de Especialista em Gestão e Engenharia da Qualidade – MBA / USP

Orientador: Prof. Dr. Adherbal Caminada Netto

São Paulo  
(2015)

**BABILLA BORINE D'ANGELO**

**Análise e sugestão de melhoria do processo de controle de qualidade de software**

Monografia apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do certificado de Especialista em Gestão e Engenharia da Qualidade – MBA / USP

Orientador: Prof. Dr. Adherbal Caminada Netto

São Paulo  
(2015)

D'ANGELO, B. B. **Título:** Análise e sugestão de melhoria do processo de controle de qualidade de software. São Paulo. 2015. 48 p. (MBA) Escola Politécnica, Universidade de São Paulo, São Paulo, 2015.

Dedico esse trabalho aos meus colegas Thiago Alessandro Silva de Jesus e Diego Dias Ferreira, por disporem do seu tempo para discutir as soluções práticas aplicadas neste trabalho, à minha família, em especial meu filho Felipe, que me acompanhou muitas e muitas noites nas aulas e me fez não desistir demonstrando o orgulho que sente por eu nunca deixar de lutar.

## **AGRADECIMENTOS**

Agradeço aos colegas deste curso, pela troca substancial de experiências profissionais e humanas e aos professores, que sempre demonstraram muito conhecimento e me fizeram acreditar que o estudo ainda é sempre o caminho da evolução.

## RESUMO

Com a crescente demanda do mercado consumidor em todos os segmentos, as empresas e fornecedores de serviços, se vêem obrigados a criarem e lançarem novos produtos e serviços de alta qualidade, em prazos e com custos cada vez mais reduzidos, para se manterem competitivas e lucrativas. Para atingirem a meta da competitividade e lucratividade, fazem uso da tecnologia que, na maioria das vezes, se torna funcional através do uso de diferentes tipos de *softwares*. O controle de qualidade de *software* é uma fase da demanda de desenvolvimento de *software* que faz parte da garantia da qualidade e que tem como objetivo reduzir o número de defeitos existentes nos *softwares* que são criados para atender a inúmeros fins, para minimizar os diferentes tipos de prejuízos que podem causar às empresas, caso tais defeitos se manifestem durante seu respectivo uso. Nos dias atuais, ainda que as empresas e fornecedores de serviços concordem que desenvolver um produto ou fornecer um serviço com qualidade aceitável seja essencial para se manter no mercado, quando o prazo ou o custo da demanda de desenvolvimento se encontram ameaçados, a primeira ação é questionar o quanto a qualidade, de fato, é percebida e importante. Com isso, as atividades do controle de qualidade de *software* quase sempre são desvalorizadas nos cronogramas das demandas, muitas vezes tendo seu tempo e escopo reduzido, ou sendo eliminadas por completo. O estudo consiste na apresentação do processo de controle de qualidade de *software* atual da empresa ***Amil Health Service Provider***, na identificação e análise de pontos problemáticos através do uso de ferramentas da qualidade e na sugestão de melhorias no processo através de mudanças nas atividades, com base nas boas práticas do mercado de TI.

Palavras-Chave: Teste. Controle de Qualidade. *Software*.

## ABSTRACT

With the growing consumer market demand in all segments, companies and service providers are bound to create and launch new high-quality products and services, with deadlines and lower costs to keep themselves competitive and profitable. To reach the competitiveness and profitability goal, they make use of technology which, in the most of times, becomes functional through the use of different types of software. The Software Quality Control is a phase of the software development project, and this one is part of the Software Assurance that has as its main goal the decrease of the number of existing bugs and issues in the softwares that are created to attend to countless purposes, to minimize different kinds of damages that may be caused to the companies, in case of these bugs are found during their use. Nowadays, even the companies and the service providers agree that developing a product or supply a service with an acceptable level of quality is essential to keep themselves in the business, when the deadline or the development's project cost are threatened, the first action is to ask how much the quality, in fact, is important and viewed. With that, the activities of the software quality control are always devalued in the projects schedules, most of the times having their scope decreased, or being eliminated at all. The study consists in the presentation of the current software quality control of the company ***Amil Health Service Provider*** and in the identification and analysis of trouble spots through the use of quality tools and suggesting process improvements through changes in the activities, based on the best practices of the IT market.

Keywords: Testing. Quality Control. Software.

## LISTA DE ILUSTRAÇÕES

Figura 1	-	Relação entre fases de desenvolvimento e testes e entregáveis	9
Figura 2	-	Macro processo do controle de qualidade de <i>software</i>	10
Figura 3	-	Macro processo de controle de qualidade de <i>software</i> da <b>Amil Health Service Provider</b>	20
Figura 4	-	Fases do processo de planejamento do teste	21
Figura 5	-	Fases do processo de modelagem do teste	22
Figura 6	-	Fases do processo de execução do teste	23
Figura 7	-	Fases do processo de apresentação de resultados do teste	24

## LISTA DE TABELAS

Tabela 1	- Matriz FOFA	25
Tabela 2	- Lista de prioridades e respectivas de ações de melhoria	28
Tabela 3	- Comparativo de custos de modelo de contratação do serviço de controle de qualidade de <i>software</i>	30
Tabela 4	- Quadro comparativo da capacidade produtiva – segundo semestre de 2014 e primeiro semestre de 2015	31
Tabela 5	- Quadro comparativo de eficiência na identificação de defeitos de produto – segundo semestre de 2014 e primeiro semestre de 2015	32
Tabela 6	- Quadro comparativo na capacidade de atendimento de demandas por colaborador e hora – segundo semestre de 2014 e primeiro semestre de 2015	33
Tabela 7	- Quadro comparativo de custos - segundo semestre de 2014 e primeiro semestre de 2015	34

## LISTA DE GRÁFICOS

Gráfico 1	- Comparativo de custos com recursos humanos	30
Gráfico 2	- Demonstrativo do aumento da capacidade produtiva por demandas	31
Gráfico 3	- Redução do total de defeitos encontrados e média de defeitos abertos por analista de teste	32
Gráfico 4	- Aumento da capacidade produtiva por pessoa	33
Gráfico 5	- Apresentação de custos por hora de trabalho e por demanda	34

## LISTA DE ABREVIATURA E SIGLAS

<b>BR</b>	Brasil
<b>BSTQB</b>	Brazilian Software Testing Qualifications Board
<b>CTFL</b>	Certified Testing Foundation Level
<b>CTAL - TM</b>	Certified Testing Advanced Level – Test Manager
<b>ISTQB</b>	International Software Testing Qualifications Board
<b>FOFA</b>	Forças, Oportunidades, Fraquezas, Ameaças
<b>PJ</b>	Pessoa Jurídica
<b>CLT</b>	Consolidação das Leis Trabalhistas
<b>S1</b>	Semestre 1
<b>S2</b>	Semestre 2
<b>NBR</b>	Norma Brasileira
<b>ISO</b>	Organization for Standardization (Organização Internacional de Normalização).
<b>IEC</b>	International Electrotechnical Commission (Comissão Eletrotécnica Internacional)
<b>TI</b>	Tecnologia da Informação
<b>CQS</b>	Controle de Qualidade de Software
<b>Qtde</b>	Quantidade
<b>R\$</b>	Real
<b>RACI</b>	Responsible, Accountable, Consult, Inform

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>7</b>
1.1	Objetivo	7
1.2	Escopo	7
1.2.1	Abrangência	7
1.2.2	Limitações	7
<b>2</b>	<b>FUNDAMENTAÇÃO</b>	<b>8</b>
<b>2.1</b>	<b>O controle de qualidade de <i>software</i></b>	<b>8</b>
2.1.1	Fases, entradas e saídas do processo de controle de qualidade de <i>software</i>	8
2.1.2	Gerenciamento do processo de controle de qualidade de <i>software</i>	15
2.1.3	Métricas do processo de controle de qualidade de <i>software</i>	16
<b>3</b>	<b>CASO</b>	<b>18</b>
3.1	Caracterização da organização	18
3.2	Descrição do problema	18
3.3	Apresentação do processo atual	19
3.3.1	Macro processo de controle de qualidade de <i>software</i>	19
3.4	Identificação e análise dos pontos críticos do processo	24
3.5	Priorização dos pontos críticos do processo	28
3.6	Realizações e contribuições	29
<b>4</b>	<b>CONCLUSÃO</b>	<b>35</b>
4.1	Sugestões para trabalhos futuros	35
<b>5</b>	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>37</b>
<b>6</b>	<b>REFERÊNCIAS COMPLEMENTARES</b>	<b>38</b>
<b>7</b>	<b>GLOSSÁRIO</b>	<b>39</b>

## 1 INTRODUÇÃO

### 1.1 Objetivo

O objetivo do trabalho é descrever o processo completo de controle de qualidade de *software* utilizado na empresa escolhida para desenvolvimento do caso, explicar a importância de cada uma de suas fases e entregáveis, e apresentar suas respectivas métricas para então, após análise das mesmas, sugerir melhorias ao processo com foco nos desvios críticos.

### 1.2 Escopo

O processo completo de controle de qualidade de *software* utilizado é considerado extenso e complexo por ser composto por outros sub-processos, de altos níveis de criticidade, com diversas entradas e com grande número entregáveis gerados.

Devido ao tempo disponibilizado para o desenvolvimento deste trabalho, será selecionado um sub-processo e as métricas que demonstram os desvios do mesmo.

O escopo do trabalho é apresentar o macro-processo do controle de qualidade de *software* da **Amil Health Service Provide** e identificar os pontos críticos, sugerir mudanças com a inclusão ou alteração de atividades, sem a necessidade de alteração das fases do processo.

Após realização do levantamento dos dados e análise dos mesmos, serão definidos os pontos críticos a serem tratados pelo estudo.

Serão então sugeridas mudanças, levando em consideração as boas práticas do mercado de TI, a fim de promover o aumento da qualidade de todos os sistemas desenvolvidos pela **Amil Health Service Provide** e por seus fornecedores.

#### 1.2.2 Limitações

A organização escolhida para o desenvolvimento do estudo é uma organização real, altamente reconhecida no mercado da área de planos de saúde e odontológico. Entretanto, para que não seja ferido o código de ética, alguns dados apresentados serão alterados.

## 2 FUNDAMENTAÇÃO

### 2.1 O controle de qualidade de *software*

O controle de qualidade de *software* é uma das disciplinas contidas na garantia da qualidade de *software*.

Qualidade é a “totalidade de características de uma entidade que lhe confere a capacidade de satisfazer as necessidades explícitas e implícitas” [ISO 9126].

Segundo [PRESSMAN], Qualidade de *Software* é a conformidade a requisitos funcionais e de desempenho que foram explicitamente declarados, a padrões de desenvolvimento claramente documentados e a características implícitas que são esperadas de todo *software* desenvolvido por profissionais.

O objetivo do controle de qualidade de *software* é medir o grau da qualidade dos produtos estáticos (documentos) e dinâmicos (*software*) gerado pelo processo de desenvolvimento de *software* tais como, planos de desenvolvimento de *software*, especificações de requisitos funcionais (requisitos funcionais) e não-funcionais (requisitos de ordens técnicas), diagramas representacionais, modelos entidades-relacionamentos, protótipos, código-fonte, manuais e o próprio *software*, através de inúmeras técnicas de testes e também através do uso de ferramentas desenvolvidas para este fim.

Os resultados apresentados pelo controle de qualidade também são direcionadores para a melhoria tanto do processo de medição quanto do processo de desenvolvimento, além de fornecerem dados para a melhoria da garantia da qualidade.

#### 2.1.1 Fases, entradas e saídas do processo de controle de qualidade de *software*

As fases, entradas e saídas do processo de controle de qualidade de *software* variam de acordo com o modelo de processo de desenvolvimento de *software* adotado pela organização.

Existem muitos modelos de desenvolvimento reconhecidos mundialmente porém, como a premissa do uso de modelos é adequar-se aos propósitos, metas e diretrizes

da organização, podem existir fases e entregáveis diferentes para cada cenário observado.

O importante a citar nesse ponto é que, para cada fase do processo de desenvolvimento de *software*, existe uma fase correspondente do controle de qualidade e que, para cada produto gerado como saída pelo desenvolvimento de *software*, deverá existir um correspondente, como saída do controle de qualidade de *software* que apresente a qualidade da saída gerada pelo primeiro processo. Por exemplo, se a organização utiliza um processo seqüencial para o desenvolvimento do *software* na fase de planejamento da demanda de desenvolvimento, deverá ser gerado pelo controle de qualidade um documento que demonstre o resultado da avaliação, em termos quantitativos e qualitativos, do Plano de Desenvolvimento do *Software*. Ver Figura 1 a seguir.

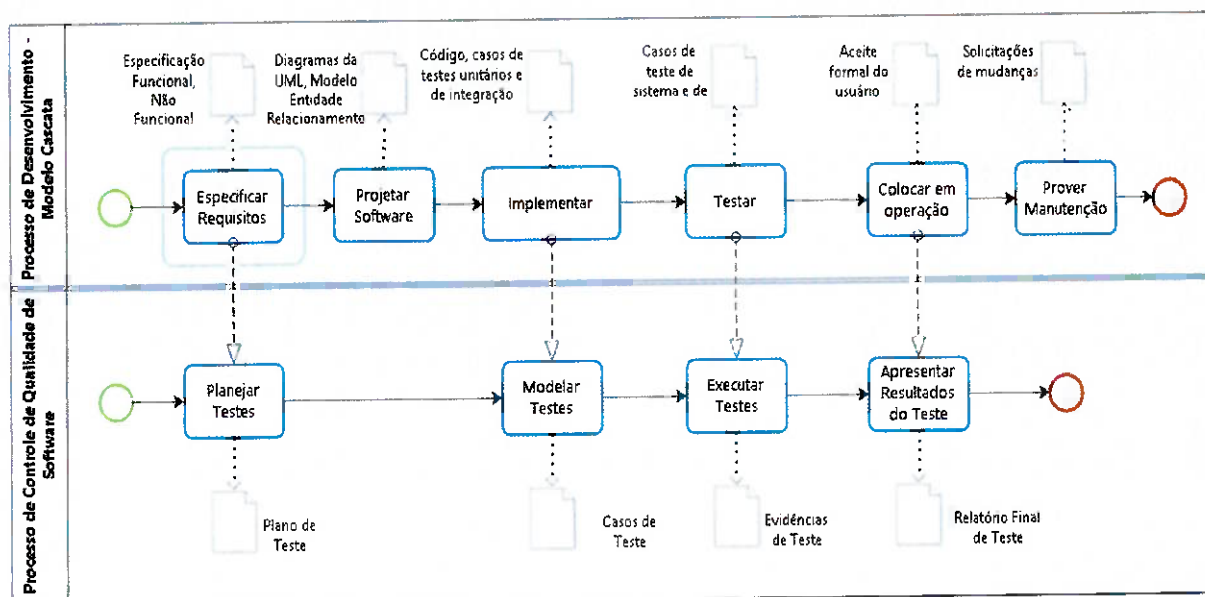


Figura 1 – Relação entre fases de desenvolvimento e testes e entregáveis

A Figura 2 a seguir mostra um exemplo do processo formal de controle de qualidade de *software* [BSTQB]:

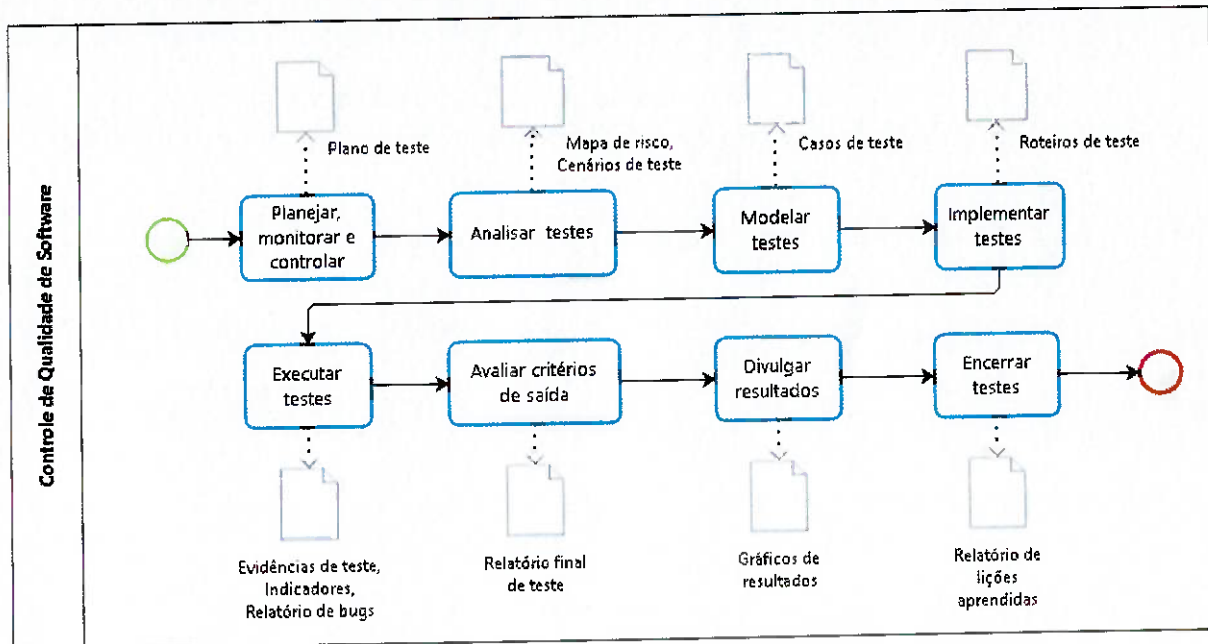


Figura 2 – Macro processo do controle de qualidade de *software*

Descrição das fases, entradas e saídas gerados por ordem de ocorrência no processo:

✓ Planejamento, monitoramento e controle

A fase inicial é de suma importância para a demanda. Nesta fase, é realizada a primeira reunião de entendimento sobre a demanda de desenvolvimento e de controle de qualidade de *software* com todos os envolvidos, onde todas as premissas, restrições, escopo, não escopo e marcos iniciais da demanda, papéis e responsabilidades são definidos.

Entradas:

*Todos os documentos existentes da demanda.*

Saídas:

*Plano de controle de qualidade, ou plano de teste.* Contém o planejamento, que é elaborado considerando alguns pontos tais como a abordagem, escopo, requisitos, níveis de teste (teste unitário, teste de integração, teste de sistema e teste de

aceitação), técnicas, infra-estrutura (ambiente e ferramentas), mão-de-obra, prazos, métricas, critérios de início e término dos níveis de teste, critérios de aceite da demanda. Este documento é submetido à aprovação formal dos envolvidos para que as atividades sejam iniciadas.

#### ✓ Análise

Nesta fase são identificadas as fontes de informação de importância para o produto a ser desenvolvido. As fontes consideradas são as documentações históricas, o conhecimento das pessoas, a documentação específica da demanda, sistemas similares, etc.

Esta análise proporciona alguns benefícios para a demanda, tais como: clareza no relacionamento entre os produtos de trabalho de teste e de desenvolvimento, prevenção de defeitos em função da descoberta dos mesmos ainda nas documentações, antes do início do desenvolvimento do *software*, maior entendimento dos requisitos explícitos e implícitos além das suas respectivas complexidades.

Entradas:

*Plano de teste.*

Saídas:

*Mapa de riscos de projeto e produto* para que os testes do *software* sejam desenvolvidos de uma forma mais assertiva, ou seja, que sejam planejados e criados testes eficazes e eficientes, que busquem encontrar maior número de defeitos nas partes mais importantes dos documentos gerados nas fases do desenvolvimento e do *software*, com a menor quantidade de testes possível.

*Cenários de teste*, que são situações que podem ocorrer no sistema que precisam ser testadas (o que deve ser testado).

Normalmente relacionamos os cenários de testes com os requisitos de *software* definidos previamente em conjunto com a área usuária solicitante do sistema.

Tais cenários de teste, após serem identificados, são incluídos no entregável "Plano de Controle de Qualidade de *Software*" e também devem ser aprovados pelos envolvidos no projeto.

### ✓ Modelagem

A modelagem é a fase onde são identificados quais as situações ou variações que a funcionalidade ou o sistema como um todo podem assumir a fim de realizar os testes nas situações mais importantes ou críticas para o negócio da organização. Essas situações devem ser derivadas dos cenários de testes identificados na fase anterior e o relacionamento entre essas entidades deve ser definido de forma explícita.

#### Entradas:

*Mapa de risco de projeto e produto;*

*Cenários de teste.*

#### Saídas:

*Casos de teste.* São as variações das condições possíveis que um cenário de teste pode assumir. Os casos de teste devem apresentar como as funcionalidades do sistema serão testadas.

### ✓ Implementação

Esta é a fase onde são detalhados os casos de testes, especificados seus pré-requisitos ou premissas, os dados necessários a serem utilizados nos testes e os passos do teste. Os casos de teste são constituídos por uma seqüência lógica de ações ou passo, que a pessoa que está executando o teste deverá seguir, afim de obter o resultado esperado ou a resposta esperada do sistema. Nesta fase os casos de testes são agrupados, ordenados e priorizados.

Também nesta fase o ambiente de teste, as ferramentas envolvidas diretamente nos testes e as ferramentas de apoio devem ser verificados para que estejam operacionais na fase de execução.

#### Entradas:

*Casos de teste.*

#### Saídas:

*Roteiros de teste.*

### ✓ Execução

Fase onde são executados os testes das funcionalidades do sistema seguindo os casos de teste, os dados informados para os testes e são observados os resultados obtidos para uma comparação com os resultados esperados. Caso os resultados previstos e realizados do teste sejam diferentes, um problema ou um *bug* deve ser registrado através dos mecanismos determinados pela organização para que uma ação, seja ela de correção ou não, possa ser tomada.

Entradas:

*Roteiros de teste.*

Saídas:

*Evidências de teste.* São imagens, filmes, áudio e outros artefatos que comprovam que o teste foi executado de acordo com os critérios definidos e o seu respectivo resultado.

*Relatório de bugs registrados.* Relatório com informações completas a respeito de todos os problemas encontrados durante os testes.

*Indicadores.* Apresentações gráficas que demonstram a qualidade dos testes e do produto testado através de indicadores.

### ✓ Avaliar critérios de saída

A fase de avaliação dos critérios de saída é muito importante, pois é a fase onde é determinado se os testes do sistema já devem ser interrompidos. Nesta fase há uma avaliação cuidadosa das métricas e indicadores, que são a base para a tomada de decisão. Avaliar se os testes já podem ser encerrados significa avaliar se o custo do teste já ultrapassou o custo associado ao risco de uma falha não identificada pelo teste ocorrer em produção. Além dos indicadores, que demonstram qualidade e custo atingidos pelo produto, existem outros fatores que determinam os critérios de saída, tais como: riscos associados a exigências legais, criticidade para manutenção do negócio ou manutenção da competitividade, entre outros.

Entradas:

*Relatório de bugs registrados. Indicadores.*

Saídas:

*Relatório final de testes.* Compreende as demonstrações de análises, através de relações entre as metas planejadas e os indicadores alcançados, as justificativas das decisões e os riscos residuais do produto, para serem mitigados posteriormente.

✓ **Divulgar resultados**

Divulgar os resultados do teste faz-se necessário para que todos os envolvidos conheçam a qualidade do produto produzido e a eficácia e eficiência do processo tanto de controle de qualidade de *software*, quanto de desenvolvimento. A frequência e a forma de divulgação e comunicação de tais resultados devem ser definidas na fase de planejamento e podem variar entre projetos, produtos, equipes, organizações, etc.

Entradas:

*Relatório final de testes.*

Saídas:

*Gráficos de resultados.* Resultados dos testes organizados em gráficos, para facilitar a visualização e entendimento dos resultados do projeto.

✓ **Atividades de fechamento de teste**

Fase onde as atividades de encerramento do projeto são cuidadosamente executadas, os produtos gerados são revisados e direcionados aos seus respectivos responsáveis, o plano de teste é atualizado para utilização em projetos futuros, a base histórica é atualizada para reutilização em outros projetos, as reuniões de lições aprendidas para amadurecimento das equipes e do processo também são realizadas e devidamente documentadas e divulgadas.

Essa fase é dividida em quatro grandes atividades, conforme a metodologia V de teste [BSTQB]:

- ✓ **Verificação de conclusão de teste:** é a atividade onde se deve garantir que todas as tarefas que foram planejadas para o projeto de teste foram concluídas.

- ✓ Transferência de artefatos de teste: é a atividade que garante que todos os produtos de trabalho gerados ao longo do projeto são entregues aos seus respectivos interessados.
- ✓ Aprendizados: são reuniões que devem ocorrer para que sejam discutidos assuntos importantes sobre o projeto, tais como boas práticas executadas, que trouxeram resultados significativos para o projeto e que devem ser repetidas futuramente, práticas adotadas que não trouxeram resultados conforme o esperado e que devem ser modificadas no futuro, entre outros.
- ✓ O arquivamento de todos os produtos de trabalho gerados no sistema de gestão de configurações.

#### Entradas:

*Todos os produtos de trabalho gerados ao longo do projeto.*

#### Saídas:

*Base de dados histórica.* Repositório de dados que contém todas as informações pertinentes ao projeto e o registro das versões dos documentos, além da versão da configuração do projeto entregue.

### **2.1.2 Gerenciamento do processo de controle de qualidade de *software***

Normalmente o processo de controle de qualidade de *software* agrega valor no sucesso geral do projeto através da prevenção de problemas graves no funcionamento do *software* desenvolvido.

O gerenciamento do processo consiste na identificação dos itens a seguir:

- ✓ Do modelo de desenvolvimento adotado;
- ✓ Das fases de teste (em alinhamento as fases do desenvolvimento);
- ✓ Dos níveis de teste;
- ✓ Das partes envolvidas e suas expectativas em relação ao produto a ser desenvolvido;
- ✓ Das necessidades de documentação;
- ✓ Das políticas da organização;
- ✓ Dos riscos;

- ✓ Dos recursos financeiros e intelectuais necessários;
- ✓ Da infra-estrutura ou ambientes operacionais de desenvolvimento e testes;
- ✓ Das ferramentas;
- ✓ Do escopo;
- ✓ Do tempo;
- ✓ Do orçamento;
- ✓ Dos dados a serem utilizados para simulações das situações de teste;
- ✓ Das evidências dos resultados;
- ✓ Dos indicadores de eficiência e eficácia do processo;
- ✓ Dos indicadores de qualidade do produto;
- ✓ Dos critérios de aceitação do cliente.

### **2.1.3 Métricas do processo de controle de qualidade de *software***

As métricas são combinações de medidas simples, que demonstram de forma quantitativa e qualitativa a capacidade de um *software* de executar uma função ou possuir um dado atributo. Elas apóiam substancialmente no entendimento, monitoramento e controle dos projetos e processos de *software* de uma organização.

As métricas de controle de qualidade de *software* são divididas por categorias, e podem estar em mais de uma categoria simultaneamente [BSTQB]:

- ✓ Métricas do projeto: com foco na medida do progresso do projeto em relação ao planejamento inicial.
- ✓ Métricas do produto: com foco na medida em relação à existência e operação de um dado atributo do produto.
- ✓ Métricas do processo: com foco na medida no processo de desenvolvimento e de controle de qualidade de *software*.
- ✓ Métricas do pessoal: com foco na capacidade de eficácia e eficiência dos indivíduos e grupos em relação às responsabilidades recebidas no projeto.

As métricas de eficácia de teste verificam a extensão dos produtos de trabalhos gerados pelo processo de teste.

As métricas de eficiência de teste verificam o esforço, duração e/ou gastos de recursos necessários para atingir esses produtos de trabalho e os resultados.

A seguir estão listados alguns exemplos de métricas de eficácia que podem ser adotadas no processo de controle de qualidade de *software*:

- ✓ Percentual de requisitos cobertos pelas condições de teste;
- ✓ Conjunto de casos de testes que foram planejados, mas que não foram executados e nem foram descartados antes do fim do projeto;
- ✓ Taxa de defeitos não descobertos pelos testes, mas que ocorreram na produção;
- ✓ Percentual de defeitos detectados;
- ✓ Percentual de defeitos críticos detectados;
- ✓ Percentual de defeitos críticos que escaparam para a produção;
- ✓ Percentual de defeitos críticos encontrados o mais cedo possível na execução;
- ✓ Percentual de riscos identificados e cobertos pelos testes executados;
- ✓ Quantidades de casos de teste executados por analista de teste;
- ✓ Quantidades de casos de teste executados por analista funcional;
- ✓ Quantidades de defeitos encontrados por analista de teste;
- ✓ Quantidades de defeitos encontrados por analista funcional;
- ✓ Quantidades de defeitos não encontrados por analista de teste;
- ✓ Quantidades de defeitos não encontrados por analista funcional.

Cada projeto é diferente de outro. Portanto, as métricas que são apropriadas para um projeto específico podem não ser apropriadas para outros projetos.

As métricas a serem utilizadas normalmente são identificadas através dos objetivos do projeto e através dos riscos associados que possam comprometer o seu sucesso e devem ser discutidas e bem definidas entre todos os envolvidos no projeto.

As métricas devem ser definidas e combinadas ainda na fase de planejamento, coletadas e armazenadas durante todo o projeto, além de serem armazenadas de forma que seu acesso seja simples e de forma a serem facilmente entendidas.

Após a coleta e armazenamento, as métricas devem ser analisadas e devem ser a base de construção de indicadores gráficos que serão apresentados aos envolvidos para identificação dos pontos problemáticos do projeto, processo e produto.

Para este trabalho, serão usadas métricas relacionadas com o retorno de investimento, ou seja, métricas capazes de demonstrar a relação custo *versus* o benefício do controle de qualidade de *software*. São métricas de eficácia e eficiência que justificam a existência da área dentro da organização através dos prejuízos evitados no ambiente produtivo.

### 3 CASO

#### 3.1 Caracterização da organização

A **Amil Health Service Provide** foi criada a partir da Casa de Saúde São José, que, na época de sua aquisição, em 1972, era uma pequena clínica na cidade fluminense de Duque de Caxias.

Cinco anos depois, já se transformava na maior maternidade privada no estado do Rio de Janeiro. Em seguida, foram também adquiridas as clínicas Somicol (atual Hospital de Clínicas Mario Lioni) e Santa Rita.

Para administrá-las, foi criada a Empresa de Serviços Hospitalares (Esho), cuja finalidade era centralizar o controle de compras, faturamento, pessoal e tesouraria. Após algum tempo, a Casa de Saúde São José deixou de integrar o grupo, que atravessou a Baía de Guanabara e comprou a Policlínica São Sebastião, no Barreto, em Niterói, em 1976. Os recursos oriundos da Esho permitiram a criação da **Amil Health Service Provider**, no Rio de Janeiro, em 1978.

Atualmente a **Amil Health Service Provide** é uma empresa provedora de produtos de saúde sendo considerada uma das maiores, senão a maior, empresa do ramo hospitalar e dental, em rede nacional e internacional.

Seus valores estão direcionados para integridade, compaixão, relacionamentos, inovação e desempenho, e sua missão é ajudar as pessoas a viver de forma mais saudável e contribuir para que o sistema de saúde funcione melhor para todos

#### 3.2 Descrição do problema

O problema apresentado pela diretoria da **Amil Health Service Provide** está diretamente ligado ao retorno de investimento. A organização possui um fornecedor de serviço que executa os testes dos sistemas, porém, tal fornecedor não conseguia atender 100% das demandas de teste, ou seja, não conseguia executar testes em todos os projetos que a organização desenvolvia mesmo com a capacidade produtiva de 20 pessoas, entre eles um líder e 19 analistas de teste.

Por este motivo, muitos defeitos de sistema não eram encontrados pelo fornecedor, manifestando-se no ambiente produtivo gerando prejuízo. Foi então questionada a

necessidade de manter o contrato com o fornecedor, visto que o investimento realizado não estava trazendo economia direta e indireta para a organização.

Para que o contrato e a prestação de serviço fossem mantidos e o fornecedor não fosse penalizado através de multas previstas legalmente, o corpo diretivo do fornecedor apresentou uma análise a fim de identificar os pontos críticos do processo e um plano de melhorias contendo ações que pudessem solucioná-los em um curto prazo de tempo (seis meses).

### **3.3 Apresentação do processo atual**

#### **3.3.1 Macro processo de controle de qualidade de *software***

O processo atual da organização não contempla todas as fases sugeridas pelo [BSTQB], porém atende as políticas de qualidade definidas.

O processo descrito na Figura 3 a seguir apresenta as principais fases e as principais atividades de cada uma delas, respectivamente.

A área responsável pelo desenvolvimento dos projetos de *software* da organização é composta de analistas funcionais, que são funcionários diretos da organização. Esses analistas entram em contato com o líder de teste do fornecedor, através de correio eletrônico, solicitando o início do planejamento do projeto de teste de uma determinada demanda.

O líder de teste do fornecedor analisa a solicitação, direciona para o analista de teste de sua equipe, que detém maior domínio nas regras de negócio relacionadas com o projeto recebido, e o processo de controle de qualidade de *software* é iniciado.

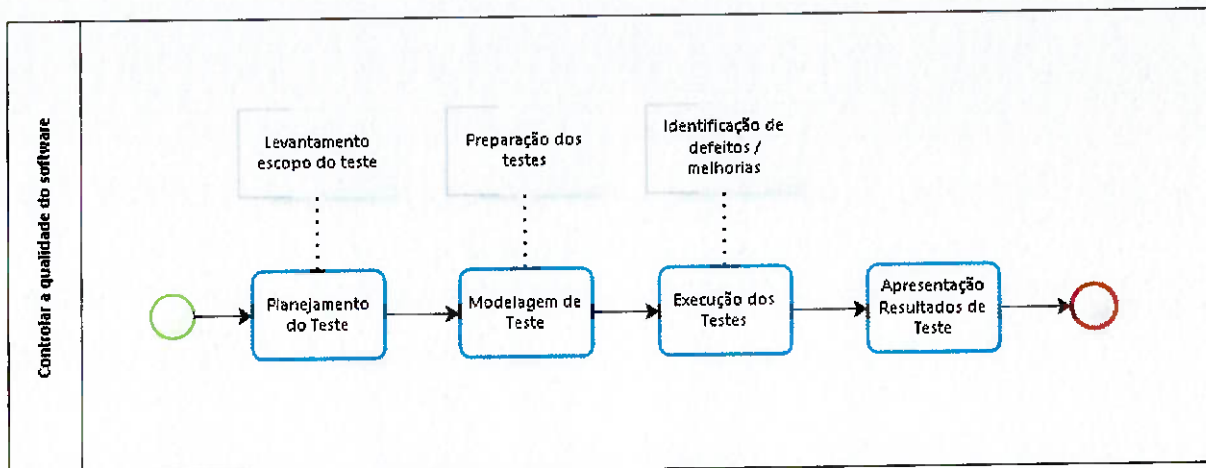


Figura 3 – Macro processo de controle de qualidade de *software* da **Amil Health Service Provider**

### ✓ Planejamento do Teste

Nesta fase, o analista de teste identifica as fontes de requisitos do projeto, organiza e agrupa, faz o entendimento das necessidades de teste da demanda, realiza reuniões de alinhamento com os analistas funcionais e elabora a estimativa de esforço (tempo) de teste para:

- Modelagem de testes (detalhamento dos casos de teste a serem executados);
- Preparação do ambiente de teste: identificação das pré-condições e massas de teste;
- Execução dos ciclos de teste (são três ciclos de teste: no primeiro são executados todos os testes identificados para o projeto; no segundo ciclo, os defeitos encontrados no ciclo um (1) que foram corrigidos, são “retestados” a fim de confirmar que tais defeitos foram, de fato, corrigidos e não ocorrem mais; no terceiro ciclo, todos os testes do projeto são realizados novamente, a fim de verificar que as correções dos defeitos não impactaram no funcionamento das outras partes do sistema e que todas as funcionalidades continuam em operação normal.

Além da estimativa de esforço de teste, o analista de teste elabora uma lista de casos de teste, para que os mesmos sejam aprovados pelo analista funcional do projeto, a fim de garantir que todos os testes críticos foram previstos.

Após a elaboração destes artefatos, o analista de teste os envia ao líder de teste, para que o mesmo obtenha a aprovação formal com a organização contratante e

para que o cronograma oficial do projeto seja gerado, com base na estimativa de esforço de teste e com base na estimativa de esforço de desenvolvimento (enviado pelo fornecedor que desenvolve o sistema) pelo analista funcional.

Uma vez que todos os artefatos sejam aprovados e que o cronograma oficial seja publicado, a próxima fase do processo é iniciada (ver Figura 4 a seguir).

**Entradas:**

*Especificação de requisitos funcionais e Protótipos*, onde são definidos, através de descrições textuais e modelos, os requisitos do produto de *software*.

**Saídas:**

*Lista de casos de testes, Estimativa do esforço de testes, Cronograma das demandas.*

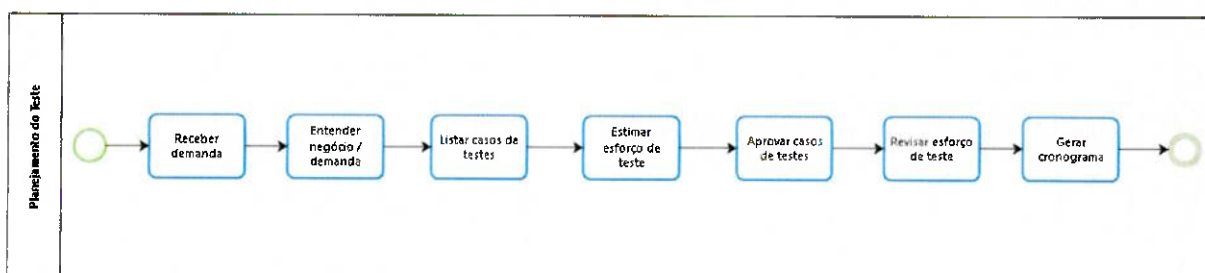


Figura 4 – Fases do processo de planejamento do teste

### ✓ Modelagem do Teste

Na fase de modelagem, o analista de teste realiza o detalhamento dos casos de teste, ou seja, descreve todos os passos que devem ser executados para a realização do teste e também descreve quais devem ser os resultados esperados pelo sistema para que as respostas reais possam ser comparadas com as respostas esperadas. As pré-condições e as massas de dados, que são os dados que devem ser utilizados no teste, também são descritas. Após a finalização destas atividades, as saídas geradas são armazenadas em uma área de rede, para que sejam visualizadas pelo analista de teste que irá executar os testes na próxima fase (ver Figura 5 a seguir).

Entradas:

*Especificação de requisitos funcionais, Protótipos e Lista de casos de testes* – todos os artefatos da fase anterior são utilizados como fonte de informação para o detalhamento dos casos de testes e identificação de pré-condições e massas de dados de teste.

Saídas:

*Casos de testes detalhados e Massa de dados de testes.*

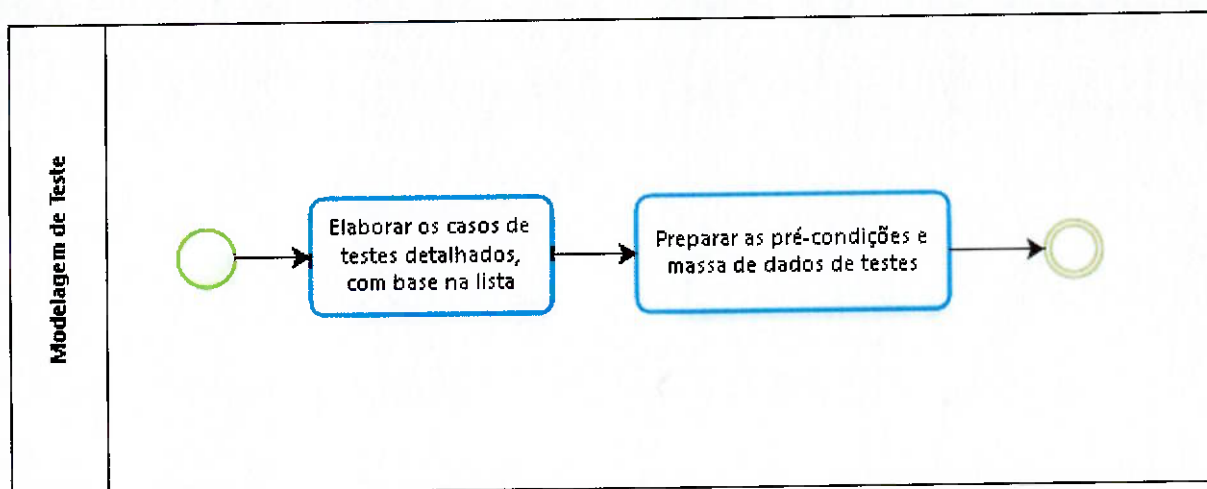


Figura 5 - Fases do processo de modelagem do teste

### ✓ Execução dos Testes

Antes de ser iniciada, existe a necessidade de confirmar se o projeto de desenvolvimento foi finalizado e se projeto já está disponível no ambiente de teste. Essa confirmação é solicitada pelo líder de teste do fornecedor ao analista funcional da organização. Caso positivo, o líder de teste informa ao analista de teste responsável pela demanda de que o projeto de teste pode ser iniciado; caso negativo, o líder de teste solicita ao analista funcional o replanejamento do cronograma da demanda e aguarda a liberação do ambiente.

Quando o ambiente de teste é liberado, o ciclo 1 de teste é executado; se forem encontrados defeitos, os defeitos são registrados em uma ferramenta específica. Para isso, os casos de teste são bloqueados e o ciclo de teste é encerrado para que os defeitos sejam corrigidos pelo fornecedor de desenvolvimento. Após o período

previsto em cronograma para correção ser finalizado, as correções são aplicadas no ambiente de teste e o ciclo de teste 2 é iniciado.

O ciclo de teste dois (2) tem o objetivo de certificar que todos os defeitos encontrados anteriormente foram corrigidos. Após a confirmação das correções, todos os casos de teste que foram bloqueados pelos defeitos são executados. Caso algum defeito ainda persista, um novo ciclo de correção é realizado e, logo em seguida, um novo ciclo de teste é realizado. Caso sejam encontrados defeitos no *software* ainda no ciclo três (3), um novo ciclo de correção é iniciado, e assim sucessivamente.

Durante a execução dos ciclos de teste são enviados relatórios parciais dos testes, para que o líder de teste possa orientar a equipe em algumas tomadas de decisões. Ao final de cada ciclo de teste são enviados relatórios finais de teste para todos os envolvidos no projeto, para acompanhamento. O processo prevê a geração de um relatório de defeitos no final desta fase para que os defeitos encontrados sejam analisados e evitados nos projetos futuros (ver Figura 6 a seguir).

#### Entradas:

*Casos de testes e Massa de dados de testes.*

#### Saídas:

*Relatório de execução dos casos de testes – Parcial/Final e Evidências de testes e Relatório de defeitos.*

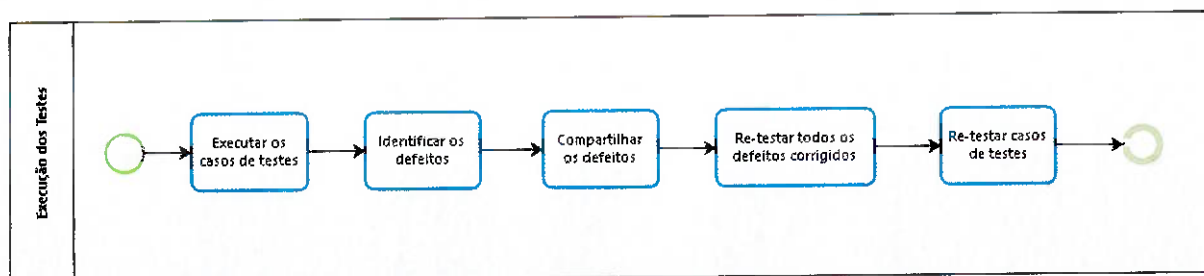


Figura 6 - Fases do processo de execução do teste

### ✓ Apresentação de Resultados do Teste

Nesta fase do processo o relatório de defeitos é avaliado por todos os envolvidos do projeto, a fim de realizarem a reunião de encerramento e de lições aprendidas para que os mesmos defeitos não ocorram em projetos futuros.

O documento chamado de Plano de TI é elaborado pelo analista de teste que foi responsável pelo projeto de teste da demanda e deve conter as evidências de teste das principais funcionalidades do sistema que foram testadas no projeto, sendo enviado ao analista funcional responsável pelo projeto (ver Figura 7 a seguir).

Entradas:

*Relatório de defeitos.*

Saídas:

*Plano de TI.*

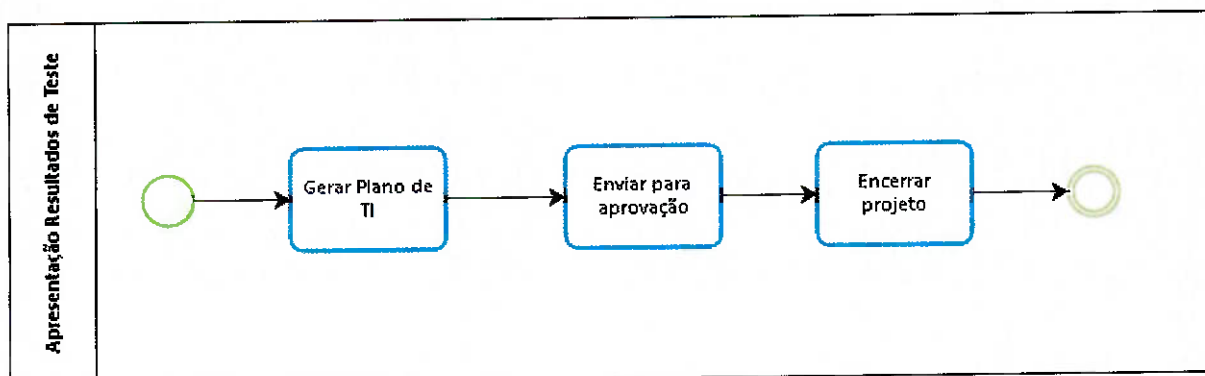


Figura 7 - Fases do processo de apresentação de resultados do teste

### 3.4 Identificação e análise dos pontos críticos do processo

Para a identificação dos pontos críticos e deficientes do processo foram realizadas algumas sessões de *brainstorming* com a equipe de controle de qualidade de *software* do fornecedor, analistas funcionais da organização cliente e fornecedor de desenvolvimento.

Após a identificação dos principais problemas, foi construída uma matriz FOFA (ver Tabela 1 a seguir) com o objetivo de construir o plano de ação para implantação de melhorias no processo.

Tabela 1 – Matriz FOFA

Forças	Fraquezas
Responsabilidades de diretoria definidas em relação à qualidade	Alto índice de defeitos nos produtos de <i>software</i> entregues pela fábrica de <i>software</i>
Participação dos departamentos nas questões ligadas à qualidade	Alto índice de rotatividade de pessoal da equipe de controle de qualidade de <i>software</i>
Equipe de controle de qualidade de <i>software</i> multidisciplinar	Ausência de definição de cargos e responsabilidades em relação à área de qualidade
Localização privilegiada	Controle manual de cronogramas de projetos
Conhecimento do processo da organização contratante	Especificações de requisitos funcionais mal escritas
Treinamento e conscientização do pessoal	Ausência de um processo de revisão de especificações de requisitos funcionais
Ferramentas de gestão de teste	Ausência de um processo de escrita e revisão dos casos de teste
Separação dos ambientes de teste e de homologação	Ausência de um processo de aprovação dos casos de teste
Flexibilidade para sugestão de melhorias do processo da organização contratante	Ausência de um cronograma de realização de <i>refresh</i> nos ambientes de teste e homologação
Base de dados para extração de métricas	Ausência de definição dos caminhos críticos das funções do sistema
Comunicação com as partes envolvidas (fornecedor de desenvolvimento e analistas funcionais da organização cliente)	Ineficiência da gestão de conflitos
Disponibilidade de ferramenta para automatização de tarefas de teste	Estimativas de desenvolvimento com prazos menores do que o necessário
Parceria com fornecedor das	Ausência de testes unitários

ferramentas de gestão e automatização de testes	
Incentivo financeiro em relação a capacitação da equipe de controle de qualidade de <i>software</i>	Inexistência de testes nos sistemas após a integração dos projetos
Acesso ao ambiente colaborativo de informações sobre o sistema e regras de negócio da organização contratante	Ausência de gestão de configuração dos ambientes de teste e homologação
	Inexistência de controle de massas de dados de testes dos projetos
	Inexistência de uma base de conhecimento da equipe de controle de qualidade de <i>software</i>
	Falta de padronização no uso das ferramentas de gestão de teste
	Estimativas de teste com esforços maiores do que os esforços do desenvolvimento
	Incapacidade de execução de testes em todas as demandas
	Falta de conhecimento no uso da ferramenta de gestão de teste por parte da fábrica de desenvolvimento
	Ambiente colaborativo com informações desatualizadas
	Falta de definição de prioridades dos projetos
	Falta de definição de criticidade e ordem de execução dos testes
<b>Oportunidades</b>	<b>Ameaças</b>
Implantação de ferramenta para gestão dos projetos e cronogramas	Indisponibilidade dos ambientes de teste e de homologação
Implantação de processo de gestão das	Troca do fornecedor de desenvolvimento

especificações de requisitos funcionais	de <i>software</i>
Definição de técnicas de identificação de casos de teste	Internalização das atividades de controle de qualidade de <i>software</i>
Implantação de um processo formal de revisão e aprovação dos artefatos gerados em todas as fases do processo de desenvolvimento	
Implantação de um processo formal de revisão e aprovação dos artefatos gerados em todas as fases do processo de teste	
Automatização das massas de dados de teste	
Criação de repositório de dados para reutilização de artefatos e massas	
Desenvolvimento de relatórios automatizados para acompanhamento da evolução da modelagem dos casos de teste	
Desenvolvimento de relatórios automatizados para acompanhamento da evolução da execução dos casos de teste	
Criação de ambiente de recuperação de desastre para o servidor da ferramenta de gestão de teste	
Definição de padrões para o processo de desenvolvimento	
Implantação de processo de testes integrados	
Implantação de processo de controle de versão	
Amadurecimento da equipe de controle	

de qualidade de <i>software</i> na automatização de testes	
---	--

### 3.5 Priorização dos pontos críticos do processo

Após a identificação e análise dos principais problemas encontrados no processo, foram selecionados alguns itens considerados prioritários, para atendimento das solicitações realizadas pela diretoria da organização (ver Tabela 2 a seguir).

Tabela 2 – Lista de prioridades e respectivas de ações de melhoria

Prioridade	Item	Ação Planejada
1	Alto índice de rotatividade de pessoal da equipe de controle de qualidade de <i>software</i>	Alteração na liderança do time, implantação de um programa de avaliação de desempenho
2	Ausência de definição de cargos e responsabilidades em relação à área de qualidade	Criação de matriz RACI com definições e publicação para os envolvidos
3	Controle manual de cronogramas de projetos	Implantação de ferramenta de gestão de projetos com fluxos de aprovações definidos e obrigatórios
4	Especificações de requisitos funcionais mal escritas	Treinamento dos departamentos envolvidos na elaboração dos artefatos
5	Ausência de um processo de revisão de especificações de requisitos funcionais	Implantação de fluxo de revisão e aprovação para os fornecedores de teste e desenvolvimento, além dos analistas funcionais
6	Ausência de um processo de aprovação dos casos de teste	Implantação de um processo onde os analistas funcionais revisam e aprovam os casos de

		teste através de uma ferramenta
7	Inexistência de testes nos sistemas após a integração dos projetos	Criação de um ambiente específico para testes integrados
8	Incapacidade de execução de testes em todas as demandas	Alteração na estrutura hierárquica com a contratação de um <i>staff</i> para apoio da liderança e treinamentos semanais com a equipe de controle de qualidade de <i>software</i>
9	Estimativas de teste com esforços maiores do que os esforços do desenvolvimento	Adoção do critério de 40% do esforço de teste em relação ao esforço do desenvolvimento e revisão do artefato modelo de estimativa de teste
10	Ausência de gestão de configuração dos ambientes de teste e homologação	Implantação do processo de gestão de configuração nos ambientes da organização contratante

### 3.6 Realizações e contribuições

Todas as melhorias sugeridas na Tabela 2, na coluna Ação Planejada, foram implantadas no primeiro semestre de 2015.

Os resultados podem ser avaliados através dos quadros comparativos (Tabelas 3 a 7) e gráficos (1 a 5) a seguir.

O quadro abaixo apresenta os cálculos dos custos em relação à forma de contratação da equipe de controle de qualidade de *software*. Os cálculos consideram vinte (20) colaboradores na equipe de controle de qualidade.

O primeiro item da Tabela 3 é o custo com terceirização. Neste modelo, um fornecedor de serviço de controle de qualidade de *software* é contratado e é responsável pela gestão da equipe e do risco. É o modelo atual da **Amil Health Service Provider**.

O segundo item da Tabela 3 é o custo considerando que a equipe de controle de qualidade faz parte do quadro de funcionários da **Amil Health Service Provider**, seguindo o regime CLT.

No último item da Tabela 3, o custo calculado considera que os colaboradores prestam serviço diretamente para a **Amil Health Service Provider**, sendo a gestão da equipe e do risco de responsabilidade da própria contratante. Esse modelo pode gerar problemas de âmbito legal entre as partes, tais como situações de vínculo empregatício. Portanto, esse modelo não é o adotado.

Tabela 3 – Comparativo de custos de modelo de contratação do serviço de controle de qualidade de *software*

Item	Valor Anual	Valor Mensal	Valor por Pessoa
Custo terceirização	R\$3.456.000,00	R\$288.000,00	R\$14.400,00
Custo CLT interno	R\$4.890.000,00	R\$407.500,00	R\$20.375,00
Custo PJ interno	R\$1.680.000,00	R\$140.000,00	R\$7.000,00

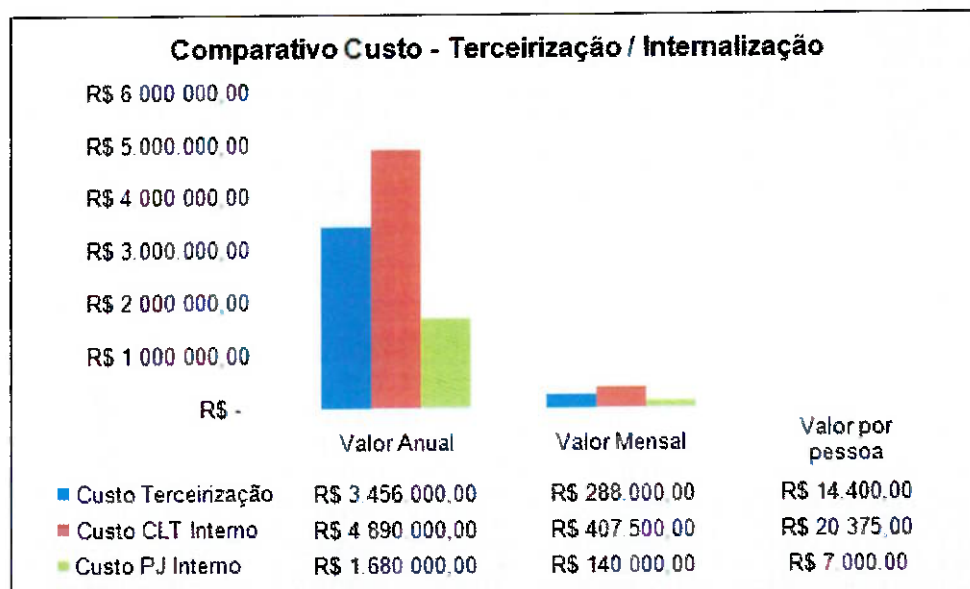


Gráfico 1 – Comparativo de custos com recursos humanos

A Tabela 4 a seguir apresenta os resultados comparativos da capacidade produtiva. As medidas foram realizadas no segundo semestre de 2014 e primeiro semestre de 2015.

Os resultados apresentados mostram que o número de demandas recebidas pela área de TI no primeiro semestre de 2015 cresceu em torno de 14% e, dessas demandas, o fornecedor de controle de qualidade de *software*, conseguiu atender 160% a mais de demandas do que atendeu no segundo semestre de 2014, conseguindo assim, demonstrar os resultados exigidos pelo corpo diretivo da **Amil Health Service Provider**.

Tabela 4 – Quadro comparativo da capacidade produtiva – segundo semestre de 2014 e primeiro semestre de 2015

<b>Categoria</b>	<b>Item</b>	<b>S2 - 2014</b>	<b>S1 - 2015</b>	<b>Margem</b>
Capacidade produtiva	Total de demandas - TI	164,5	142	14%
	Total de demandas atendidas CQS	40	104	160%
	% de demandas atendidas CQS	24%	73%	
	Média de demandas por pessoa	2	5,2	130%

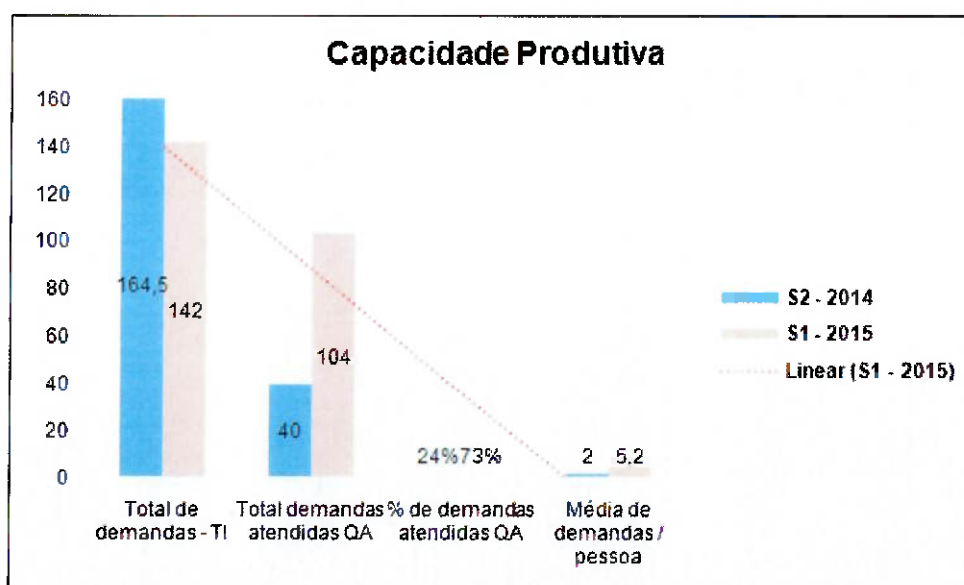


Gráfico 2 – Demonstrativo do aumento da capacidade produtiva por demandas

Além do aumento da capacidade produtiva, pode-se perceber na Tabela 5 que as ações geraram uma melhoria na qualidade dos produtos de *software*, pois a quantidade de defeitos encontrados foi reduzida em 30% de um semestre para o outro.

Tabela 5 – Quadro comparativo de eficiência na identificação de defeitos de produto – segundo semestre de 2014 e primeiro semestre de 2015

<b>Categoria</b>	<b>Item</b>	<b>S2 - 2014</b>	<b>S1 - 2015</b>	<b>Margem</b>
Defeitos	Total de defeitos encontrados	1280	891	-30%
	Média de defeitos por demanda	32	9	-73%
	Média de defeitos por pessoa	64	45	-30%

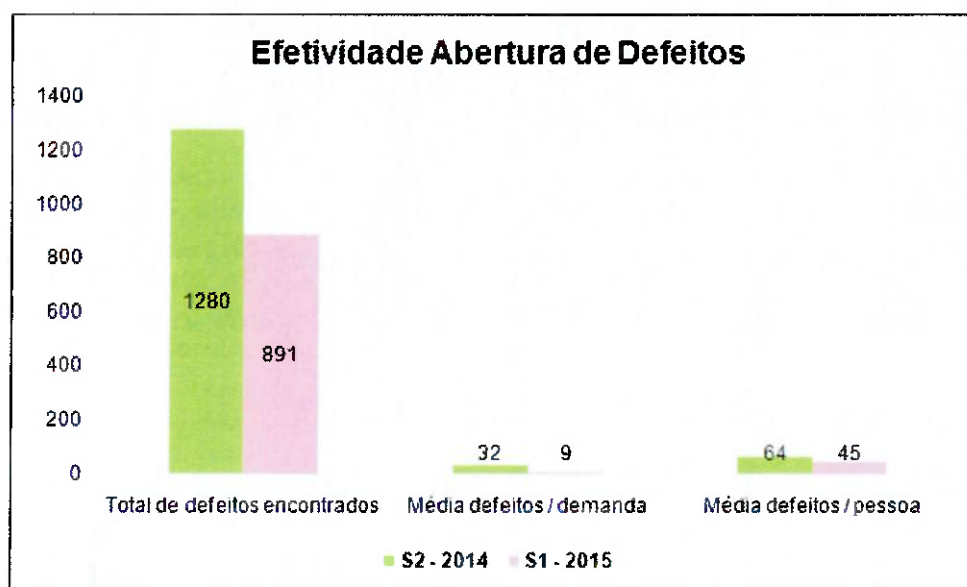


Gráfico 3 – Redução do total de defeitos encontrados e média de defeitos abertos por analista de teste

Na Tabela 6 a seguir, a capacidade de atendimento das demandas foi medida e pode-se constatar que houve redução da média de horas por demanda.

Tabela 6 – Quadro comparativo na capacidade de atendimento de demandas por colaborador e hora – segundo semestre de 2014 e primeiro semestre de 2015

<b>Categoria</b>	<b>Item</b>	<b>S2 - 2014</b>	<b>S1 - 2015</b>
Capacidade	Quantidade de pessoas	20	20
Homem / Hora	Total de horas realizadas	21120	21120
	Média de horas por demanda	528	203

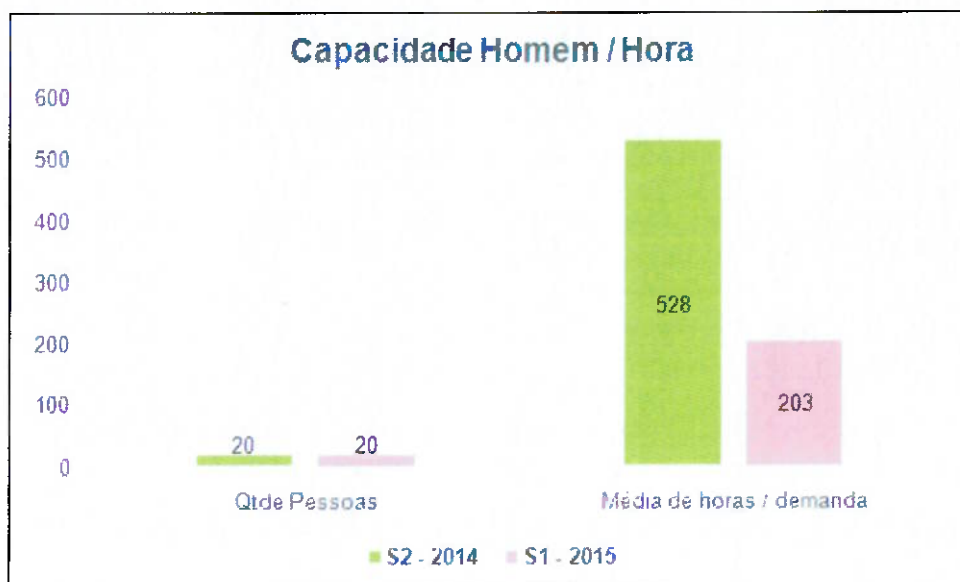


Gráfico 4 - Aumento da capacidade produtiva por pessoa

A Tabela 7 a seguir mostra que, com um aumento de 0,12% no investimento de um semestre para o outro, o custo médio por demanda foi drasticamente reduzido.

Tabela 7 – Quadro comparativo de custos - segundo semestre de 2014 e primeiro semestre de 2015

<b>Categoria</b>	<b>Item</b>	<b>S2 - 2014</b>	<b>S1 - 2015</b>
Custo	Custo operacional	R\$1.726.000,00	R\$1.728.000,00
	Custo por pessoa	R\$86.300,00	R\$86.400,00
	Custo por hora	R\$81,72	R\$81,82
	Média de custo por demanda	R\$43.150,00	R\$16.615,38

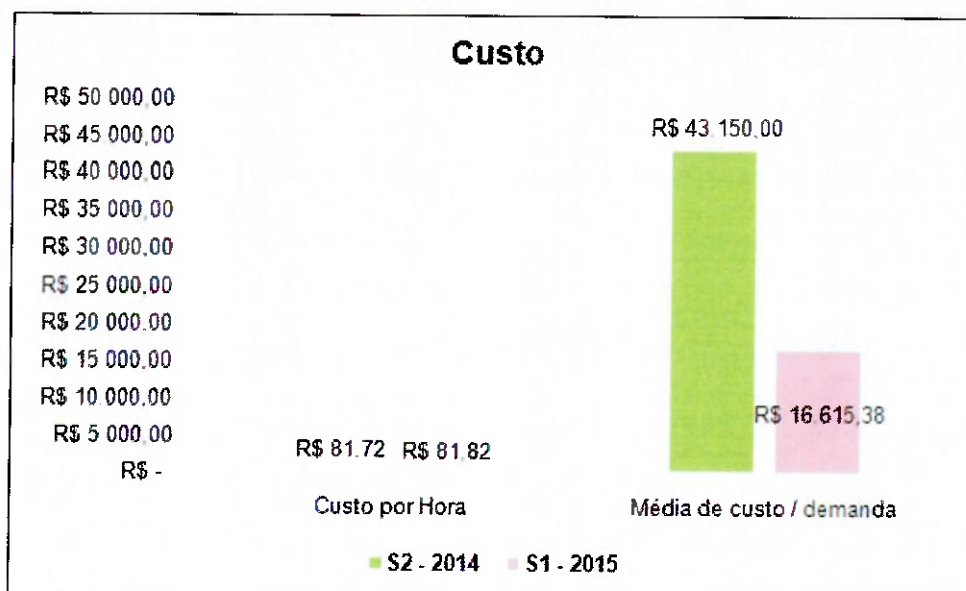


Gráfico 5 – Apresentação de custos por hora de trabalho e por demanda

## 4 CONCLUSÃO

Através do uso de algumas das ferramentas da qualidade, foi possível identificar, coletar e analisar dados que apontaram os pontos críticos de falhas no processo de controle de qualidade de *software* da **Amil Health Service Provider**. Com isso, foi possível priorizar as ações de correções que trariam melhorias nas entregas sem gerar impactos de mudanças no processo formal da organização.

Os resultados obtidos através dos dados compilados foram capazes de reafirmar que o aumento dos projetos e demandas de desenvolvimento de *software* é cada vez maior, e que os fatores qualidade, prazo e custo são determinantes para algumas decisões estratégicas da organização, inclusive na continuidade da relação de contratação dos fornecedores de serviços de desenvolvimento e controle de qualidade de *software*.

As ações sugeridas para resolução em curto prazo (seis meses) dos principais problemas foram implantadas por todas as equipes envolvidas na operação e os resultados foram medidos e compilados pelos líderes de tais equipes.

Os dados apresentados neste trabalho, através das tabelas comparativas e gráficos comprovam que, de fato, as ações de correções das fraquezas observadas na matriz FOFA contribuíram para o aumento na qualidade dos produtos de *software* entregues pelo fornecedor de desenvolvimento, pois houve uma redução relevante no número de defeitos encontrados pela equipe de controle de qualidade de *software*. Também contribuíram para a redução dos riscos das ameaças como por exemplo, a internalização das atividades de controle de qualidade de *software*.

Finalmente, pode-se observar que a diminuição dos problemas nos sistemas desenvolvidos e o aumento da capacidade produtiva comprovaram que a melhoria do serviço prestado pelo fornecedor de controle de qualidade de *software* reduziu o prejuízo da organização gerado por problemas de sistema que ocorrem no ambiente produtivo.

### 4.1 Sugestões para trabalhos futuros

Como continuidade deste trabalho, sugere-se que seja feito um novo alinhamento com o corpo diretivo da organização a fim de entender as novas expectativas do

cliente para então realizar nova medição do processo e atualizar a matriz FOFA com o objetivo de reavaliar os principais problemas.

A automatização de processos e reutilização de artefatos devem ser atividades desenvolvidas em um próximo momento, pois reduzem consideravelmente o esforço de planejamento, modelagem e execução dos testes, além de diminuir a necessidade de treinamento dos analistas de teste e reduzir os prazos finais de projetos.

## 5 REFERÊNCIAS BIBLIOGRÁFICAS

[BSTQB] Brazilian Software Testing Qualifications Board - syllabus\_ctal\_tm\_2012br  
– Corpo de conhecimento para certificação em qualidade de software – nível  
avançado – gerenciamento - versão 2012 BR – Outubro de 2012 – São Paulo.  
Disponível em: <http://www.bstqb.org.br/?q=download>.

[PRESSMAN] R.S. Pressman (2011).

Software Engineering: A Practitioner's Approach. 7<sup>th</sup> Edition. The McGraw Hill  
Companies Inc. ISBN: 0073375977.

## 6 REFERÊNCIAS COMPLEMENTARES

[IEEE829] IEEE Standard for Software and System Test Documentation. Capítulos 2 e 4.

[FEWSTER AND GRAHAM] M. Fewster and D. Graham (1999).  
Software Test Automation, Effective use of test execution tools. Addison-Wesley.  
ISBN 0-201-33140-3.

[IEEE 610] IEEE 610.12:1990. Standard Glossary of Software Engineering Terminology.

[Fenton] N. Fenton (1991).  
Software Metrics: a Rigorous Approach. Chapman & Hall.  
ISBN 0-53249-425-1.

[ISO 9000] ISO 9000:2005. Quality Management Systems – Fundamentals and Vocabulary.

[IEEE 829] IEEE 829:1998. Standard for Software Test Documentation.

[ISO 9126] ISO/IEC 9126-1:2001. Software Engineering – Software Product Quality – Part 1: Quality characteristics and sub-characteristics.

## 7 GLOSSÁRIO

As definições a seguir relacionadas foram extraídas de Dicionário Aurélio, 5ª edição e Glossário de Termos do ISTQB - Versão 2.4br.

- Artefato** - Produtos produzidos durante o processo de teste e requeridos para planejar, projetar e executar testes, entre eles documentação, roteiros, entradas/inputs, resultados esperados, procedimentos de preparação e de limpeza, arquivos, bancos de dados, ambiente e qualquer software adicional ou utilitários utilizados no teste [Fewster e Graham].
- Ciclo de teste** - Execução do processo de teste contra um único release identificável do objeto de teste.
- Critério de aceite** - Os critérios de saída que um componente ou sistema deve satisfazer a fim de ser aceito por um usuário, cliente ou outra entidade autorizada. [IEEE 610].
- Critério de inicio** - Conjunto de condições genéricas e específicas que permite um processo avançar com uma determinada tarefa, por exemplo: fase de teste. A finalidade dos critérios de entrada é evitar que uma tarefa implique em mais esforços (desperdício) em comparação com o esforço necessário.
- Defeito** - Falha em um componente ou sistema que pode fazer com que o componente ou sistema falhe ao desempenhar sua função, por exemplo, uma sentença incorreta ou uma definição de dados incorreta. Um defeito, se descoberto durante a execução, pode levar a falha do componente ou do sistema.
- Demanda** - Solicitações de mudanças no sistema oriundas da área usuária do sistema
- Eficácia** - Força latente que têm as substâncias para produzir determinados efeitos.

	Virtude de tornar efetivo ou real. Força (de produzir efeitos).
Eficiência	- Qualidade do que é eficiente. Capacidade para produzir realmente um efeito. Qualidade de algo ou alguém que produz com o mínimo de erros ou de meios.
Entregável	- Qualquer produto (trabalho) que deva ser entregue a alguém que não o autor do produto (trabalho).
Escopo	- As funções que serão testadas
Estimativa	- Aproximação calculada de um resultado relacionado com vários aspectos do teste (por exemplo, esforço despendido, data de conclusão, custos envolvidos, número de casos de teste, etc), que é utilizável mesmo se os dados de entrada sejam incompletos, incertos ou incompreensíveis.
Falha	- Desvio do componente ou sistema da entrega, resultado ou serviço esperado [Fenton].
Marco	- Determinado ponto de um projeto no qual os "entregáveis" definidos (intermediários) e os resultados devem estar prontos.
Problema	- Ver defeito.
Projeto	- Conjunto único de atividades coordenadas e controladas com datas de início e fim, comprometidas a atingir um objetivo conforme requisitos específicos, incluindo as limitações de tempo, custo e recursos [ISO 9000].
Protótipo	- Desenho das telas do sistema
Reteste	- Teste que executa os casos de teste que falharam na última vez que foram executados, a fim de verificar o sucesso das ações corretivas.
Software	- Programas de computador, procedimentos e possível documentação associada e dados relativos à operação de um sistema de computador. [IEEE 610].
Teste	- Conjunto de um ou mais casos de teste [IEEE 829].