CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA"
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Especificação de uma Técnica para Estimativa de Elaboração e Execução de Testes Funcionais

HELENA MONICI CABRINI

Marília, 2012

CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA"
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Especificação de uma Técnica para Estimativa de Elaboração e Execução de Testes Funcionais

Monografia apresentada ao Centro Universitário Eurípides de Marília como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.
Orientador: Prof. Fábio Lucio Meira

Marília, 2012

“A palavra impossível só existe no dicionário dos perdedores”.

Napoleão

“Você pode encarar um erro como uma besteira a ser esquecida, ou como um resultado que aponta uma nova direção”.

Desconhecido

Resumo

Desenvolver software com qualidade e o uso de testes de software são ações que se tornaram primordiais em desenvolvimento de um software, para garantir que a aplicação realizará suas funções conforme o que foi solicitado. Além da complexidade de uma determinada tarefa, é de grande importância também estimar seu tempo de execução, podendo assim o alocar recursos conforme o necessário, cumprindo metas, prazos de clientes e entregando um projeto ou software adequado, possibilitando melhor gerenciamento do projeto.

Análise de Ponto de Função e Análise de Ponto de Caso de Uso são técnicas utilizadas para estimar o número de horas/homem necessárias para o desenvolvimento de um projeto e seu custo estimado.

É importante desenvolver uma técnica para avaliar a complexidade apenas dos testes funcionais aplicados em componentes desenvolvidos utilizando-se as seguintes técnicas: programação orientada a objetos, definição de requisitos funcionais utilizando modelo de caso de uso e definição de métricas utilizando análise de ponto de caso de uso.

Assim, foi desenvolvida uma técnica para estimar a elaboração e execução dos testes funcionais feito de forma manual.

Com a aplicação da técnica sobre os componentes gerados, espera-se comprovar previamente a validade da proposição. Testes futuros e massivos poderão comprovar o resultado apresentado nesse trabalho.

Palavras-chave:qualidade de software, métricas, análise de ponto de caso de uso, complexidade de teste, testes, estimativa.

Abstract

Develop quality software and the use of software testing are actions that have become an essential activity in software development, to ensure that the application will perform its functions as what was requested. Beyond the complexity of a given task, it is also of great importance to estimate its running time, and can allocate resources as needed, meeting objectives, deadlines for clients and delivering a project or appropriate software, enabling better management of the project.

Analysis of Function Point and Analysis of Use Case Point techniques are used to estimate the number of man / hours needed for the development of a project and its estimated cost.

It is important to develop a technique to evaluate the complexity of functional testing only applied to components developed using the following techniques: Object Oriented Programming, Defining Functional Requirements using Use Case Model and Defining metrics using Point Analysis Case Use.

Thus, we developed a technique for estimating the development and execution of functional tests done manually.

With the application of the methodology on the generated components, previously expected to prove the validity of the proposition. Tests and massive future may prove the results presented in this work.

Keywords: software quality, metrics, analysis point of use case, test complexity, testing, estimate.

Sumário

Capítulo 1. Introdução 7
1.1 Objetivos 9
1.2 Justificativa 9
Capítulo 2. Teste Funcional 9
2.1 Conceitos Gerais 9
2.2 Atividades de Teste Funcional 11
2.3 Documentações 12
Capítulo 3. Métricas de Software 14
3.1 Conceitos Gerais 14
3.2 Análises de Ponto de Caso de Uso 15
3.3 Análise de Ponto de Função e CoCoMo II 16
Capítulo 4. Testes Unitários e JUnit 17
4.1 Testes Unitários 17
4.2 JUnit 18
Capítulo 5. Definição da Técnica Desenvolvida 19
5.1 Técnica desenvolvida 19
5.1.1 Introdução à técnica 19
5.1.2 Atividades da técnica 20
5.2 Aplicações da técnica desenvolvida em amostras 21
5.2.1 Primeira Amostra: Sistema de Provas Online 21
5.2.2 Segunda Amostra: Sistema de Gerenciamento de Compras Coletivas 30
5.2.3 Terceira Amostra: Sistema de Gerenciamento de senhas 43
5.2.4 Quarta Amostra: Sistema de Agenda Telefônica 49
5.3 Resultado e definições da técnica 54
Capítulo 6. Conclusões 57
Referências Bibliográficas 57

Capítulo 1. Introdução

Os projetos de desenvolvimento de software não são tão simples e fáceis de serem realizados. Por conta disso, há grande risco de serem encontrados problemas. Nesses projetos, os testes de software não eram valorizados, tornando uma atividade pouco elaborada, sem recursos e sempre ocorria no final do desenvolvimento, sendo o seu objetivo apenas encontrar erros.

Com o avanço da tecnologia, da inovação, do aumento de concorrentes, cresce também a procura da rapidez e eficiência nos programas ou sistemas. Assim, com a complexidade de garantir a qualidade do software e menores riscos, alguns desenvolvedores e algumas empresas estão adotando as atividades de verificação e validação do software como estratégia para melhor oferecer seu produto ou serviço.

Existem várias definições para teste de software:

• É o processo de executar um programa ou sistema com a intenção de encontrar defeitos (Myers, 1979).

• Qualquer atividade que a partir da avaliação de um atributo ou capacidade de um sistema seja possível determinar se ele alcança os resultados desejados (Hetzel, 1988).

• O processo de operação de um sistema ou componente em específicas condições, observando ou registrando os resultados, e fazendo uma avaliação em alguns aspectos do sistema ou componente (IEEE, 1990).

O processo de teste e qualidade de software se tornou uma atividade primordial em desenvolvimento de um software, para garantir que a aplicação se comportará de forma coerente com o que foi solicitado.

Apesar de não ser possível, através de testes, provar que um programa está correto, os testes, se conduzidos sistemática e criteriosamente, contribuem para aumentar a confiança de que o software desempenha as funções especificadas e evidenciar algumas características mínimas do ponto de vista da qualidade do produto (Barbosa, 2002).

Para melhor dividi-los, os testes podem ser considerados em alguns tipos: teste de unidade, teste de integração, teste de sistemas e teste de regressão. Dentro de algumas destas, existem as etapas que são: planejamento, elaboração de casos de teste, execução e análise.

Para que o desenvolvimento de um software seja bem controlado, mensurado e definido, o gerenciamento de projeto é de fundamental importância ao projeto, dividindo suas

atividades, para melhor coordená-lo. Assim, durante o processo de concretização do projeto, o escopo, o esforço, os recursos, os riscos, o custo, as tarefas, são mais bem analisadas.

Mesmo com extenso planejamento, durante o projeto pode haver diversas mudanças, tanto a pedido do cliente quanto mudança na tecnologia utilizada, deixando mais complexo o gerenciamento. Para atender essas necessidades, um exemplo de processo é o RUP: Um processo de engenharia de software cujo objetivo é garantir a produção de software de qualidade, que atenda aos requisitos estabelecidos pelo cliente (escopo), respeitando um orçamento (custo) e um cronograma (prazo) previamente definidos. (Martins, 2006).

O planejamento do trabalho é muito complexo, deixando assim algumas dúvidas quanto ao tempo, ao prazo, aos recursos humanos, custo, entre outros.

Para controlar o projeto de maneira efetiva, a solução proposta é retirar métricas para a entrega do produto.

As métricas são utilizadas para prever retorno do investimento, a produtividade do processo, validar a qualidade do software, definir a gerência de projetos, melhorar as estimativas. Sendo assim, durante todo processo de desenvolvimento deve haver uma seleção contínua de dados de produtividade e resultados, tanto do produto quanto do processo, para que as métricas sejam recolhidas.

As métricas de software são diretas, indiretas, orientada ao tamanho, orientada a função.

O custo da qualidade inclui todos os custos decorrentes da busca da qualidade ou da execução das atividades relacionadas à qualidade. (Pressman, 2002).

O planejamento do tempo pode ter benefícios determinando sensores para avaliar quanto os projetos e processos estão fora de controle e passar a controlá-los, não perder prazos e metas de qualidade e verificar os impactos de melhoramentos de tecnologia.

O tamanho do projeto a ser desenvolvido e a quantidade de pessoas envolvidas no processo aumentam ainda mais a sua complexidade.

É muito importante que seja mensurado o tempo estimado de um projeto de desenvolvimento de software, dada a complexidade da tarefa, pois determina assim as dependências e a duração das atividades.

As estimativas fornecem dados que permitem prever a quantidade de pessoas que serão necessárias, o tempo necessário e os custos do projeto. Não é possível elaborar cronograma e orçamento sem o uso de estimativas.

1.1 Objetivos

Algumas técnicas definem a complexidade do desenvolvimento de testes funcionais, porém, para os Gerentes de Projetos, é importante também que, além da complexidade de uma determinada tarefa, seja possível estimar seu tempo de execução, podendo assim o mesmo alocar recursos conforme o necessário.

Assim, o objetivo é desenvolver uma técnica que permita avaliar a complexidade e o tempo de execução dos testes funcionais aplicados em componentes desenvolvidos, utilizando-se as seguintes técnicas: Programação Orientada a Objetos; Definição de Requisitos Funcionais utilizando modelo de Caso de Uso e Definição de métricas utilizando Análise de Ponto de Caso de Uso.

Avaliada a complexidade dos testes, espera-se que seja possível estabelecer o tempo de desenvolvimento do teste para cada funcionalidade. A técnica deverá identificar o período de teste para: Escrita de casos de teste; Escrita de cenários de teste, Escrita de S*cript* de teste; Execução e *Log* de teste.

1.2 Justificativa

Em geral, técnicas como Análise de Ponto de Função e Análise de Ponto de Caso de Uso são técnicas utilizadas para estimar o número de horas/homem necessárias para o desenvolvimento de um projeto, e, por consequência, seu custo estimado.

Algumas técnicas definem a complexidade do desenvolvimento de testes funcionais, porém, para os Gerentes de Projetos, é importante também que, além da complexidade de uma determinada tarefa, seja possível estimar seu tempo de execução, podendo assim o mesmo alocar recursos conforme o necessário.

Capítulo 2. Teste Funcional

2.1 Conceitos Gerais

O teste funcional tem seus casos de teste planejados exclusivamente pelas especificações dos componentes de implementação, tendo como principal objetivo validar se a funcionalidade desenvolvida está de acordo com as regras especificadas, confrontando seus

resultados esperados, ou seja, informar dados de entrada, processar, obter um resultado e compará-lo ao esperado, validando assim apenas resultados e não o comportamento interno do sistema.

Teste funcional é uma técnica utilizada para se projetarem casos de teste na qual o programa ou sistema é considerado uma caixa preta e, para testá-lo, são fornecidas entradas e avaliadas as saídas geradas para verificar se estão em conformidade com os objetivos especificados (Delamaro, Maldonado, Jino, 2007).

Conhecido Teste de Caixa Preta, pois seus valores internos são desconhecidos. A visão clara é apenas do lado externo (dados de entrada e dados de saída), diferente do Teste Estrutural que verifica a estrutura interna do sistema e a maneira em que os dados estão sendo executados.

Com o Teste Funcional é possível descobrir erros de desempenho, erro de inicialização e término, erros de interface, funções incorretas ou inexistentes, erro na estrutura de dados, erro no acesso ao bando de dados, entre outros, diminuindo visivelmente grande parte de incidentes que podem ocorrer após a implementação.

O que distingue essencialmente as três técnicas de teste – Funcional, Estrutural e Baseada em Erros – é a fonte utilizada para definir os requisitos de teste. Além disso, cada critério de teste procura explorar determinados tipos de defeitos, estabelecendo requisitos de teste para os quais valores específicos do domínio de entrada do programa devem ser definidos com o intuito de exercitá-los. (Delamaro, Maldonado , Jino, 2007).

O objetivo do teste funcional não é manter foco no código do sistema ou na sua implementação, como nos testes unitários, e sim realizar uma simulação do sistema interagindo com o usuário final.

Existem alguns critérios para execução dos testes funcionais, onde são baseados apenas em especificações de requisitos funcionais, sendo aplicados em sistema com qualquer paradigma de programação. São eles: Particionamento de Equivalência, Análise do Valor Limite e Grafo Causa-Efeito.

Particionamento de Equivalência: Técnica utilizada para que os testes sejam projetados em partes, dividindo em classes que serão testadas a partir de um caso de uso específico. Uma classe de cada partição será representada ao menos uma vez durante os testes. O objetivo dessa técnica é eliminar os casos de testes redundantes.

Análise do Valor Limite: Complementar ao Particionamento de Equivalência, porém os dados de teste são escolhidos rigorosamente com limites nas condições de entrada. A saída

do programa também é particionada e são exigidos casos de teste que produzam resultados nos limites dessas classes de saída.

Grafo Causa-Efeito: Diferente dos outros, este critério explora combinações das condições de entrada. São apontadas condições de entrada (causas) e as ações (efeitos) do programa, depois é elaborado um grafo relacionando as causas e efeitos, podendo ser transformado em tabela de decisão para elaboração dos casos de teste (Delamaro, Maldonado, Jino, 2007).

2.2 Atividades de Teste Funcional

A realização da atividade de teste de software não é somente procurar erros. Outras tarefas devem ser executadas, como:

Teste de Requisitos: Verifica se o sistema é executado conforme o que foi especificado. São realizados através da criação de condições de testes e *checklists* de funcionalidades (Checklist é uma lista de verificação que incorporada ao Processo de Teste simplifica e padroniza a execução dos testes (Java Magazine, 1987)).

Teste de Regressão: Testa se algo mudou em relação ao que já estava funcionando corretamente, ou seja, é voltar a testar segmentos já testados após uma mudança em outra parte do software. Os testes de regressão devem ser feitos tanto no software quanto na documentação.

Teste de Tratamento de Erros: Determina a capacidade do software de tratar transações incorretas. Esse tipo de teste requer que o testador pense negativamente e conduza testes como: entrar com dados cadastrais impróprios, tais como preços, salários, etc., para determinar o comportamento do software na gestão desses erros. Produzir um conjunto de transações contendo erros e introduzi-los no sistema para determinar se este administra os problemas.

Teste de Suporte Manual: Verifica se os procedimentos de suporte manual estão documentados e completos, determina se as responsabilidades pelo suporte manual foram estabelecidas.

Teste de Interconexão: Garante que a interconexão entre os softwares de aplicação funcione corretamente. Pois, softwares de aplicação costumam estar conectados com outros softwares de mesmo tipo.

Teste de Controle: Assegura que o processamento seja realizado conforme sua intenção. Entre os controles estão a validação de dados, a integridade dos arquivos, as trilhas de auditoria, o backup e a recuperação, a documentação, entre outros.

Teste Paralelo: Comparar os resultados do sistema atual com a versão anterior determinando se os resultados do novo sistema são consistentes com o processamento do antigo sistema ou da antiga versão. O teste paralelo exige que os mesmos dados de entrada rodem em duas versões da mesma aplicação. Por exemplo: caso a versão mude e os requisitos não, os dados de saída das duas versões devem ser iguais (Bastos, 2007).

As técnicas de testes estruturais buscam garantir que o produto seja estruturalmente sólido e que funcione corretamente, o foco dos testes é averiguar o comportamento do sistema em determinadas situações. Já as técnicas de Testes Funcionais objetivam garantir que os requisitos e as especificações do sistema tenham sido atendidos, o foco dos testes é justamente a comparação do que foi planejado com o que foi produzido.

2.3 Documentações

A atividade de teste de software não é uma atividade simples, por si só. Sua complexidade demanda um *framework* de desenvolvimento de testes, o qual deve ser devidamente planejado e documentado.

Realizar uma documentação de testes de software de qualidade e bem elaborada, devem ser seguidas etapas como: planejamentos de todos os testes, especificar e refinar as atividades de teste e mostrar os resultados obtidos após a execução dos testes.

Para programadores, empresas, organizações, a documentação pode elevar muito o controle dos testes, melhores resultados, ótima organização e capacidade de gestão de teste, se feita de maneira adequada.

O IEEE criou o “IEEE 829” para documentação de teste de software, devendo assim absorver qual é o documento e os itens mais úteis para testar e validar o projeto, havendo oito etapas:

- Especificação:

    Plano de Teste

    Especificação de Projeto de Teste

    Especificação de Caso de Teste

    Especificação de Procedimento de Teste

- Relatório:

Log de Teste

Relatório de Incidente de Teste

Relatório de Sumário de Teste

Relatório de Encaminhamento de Item de Teste

O Plano de Teste é o documento mais importante. É elaborado conforme as especificações de requisitos, e o desenvolvimento do software é utilizado como base para todos os outros documentos. Especifica as tarefas, os riscos, os itens e as funcionalidades a serem testadas.

- **Especificação de Projeto de Teste**: Refina a abordagem apresentada no Plano de Teste e identifica as funcionalidades e características a serem testadas pelo projeto e por seus testes associados.
- **Especificação de Caso de Teste**: Define os casos de teste, incluindo dados de entrada, resultados esperados, ações e condições gerais para a execução do teste.
- **Especificação de Procedimento de Teste**: Especifica os passos para executar um conjunto de casos de teste.

Os relatórios de teste são cobertos por quatro documentos:

- **Diário de Teste**: Apresenta registros cronológicos dos detalhes relevantes relacionados com a execução dos testes.
- **Relatório de Incidente de Teste**: Documenta qualquer evento que ocorra durante a atividade de teste e que requeira análise posterior.
- **Relatório-Resumo de Teste**: Apresenta de forma resumida os resultados das atividades de teste associadas com uma ou mais especificações de projeto de teste e provê avaliações baseadas nesses resultados.
- **Relatório de Encaminhamento de Item de Teste**: Identifica os itens encaminhados para teste no caso de equipes distintas serem responsáveis pelas tarefas de desenvolvimento e de teste. (IEEE 829).

É de extrema importância a documentação dos testes para controlar atividades e o que está sendo testado, controlar o que já foi testado, depois de criada a base de documentação permite o reaproveitamento das mesmas, melhor manutenção do software, evidencia a qualidade e melhora a qualidade das atividades de teste.

As documentações têm foco nos principais itens do processo de teste de software, com o objetivo de aumentar a percepção da qualidade na visão do usuário final.

Capítulo 3. Métricas de Software

3.1 Conceitos Gerais

As métricas de software são responsáveis, principalmente, por colher informações sobre o desenvolvimento do software e oferecer dados para auxiliar no planejamento do projeto. O uso de métricas auxilia ainda na avaliação da qualidade do produto, controle de qualidade, produtividade do processo, definir técnicas de qualidade do produto, prever retorno do investimento, definir a gerência de projetos, melhorarem as estimativas, e não apenas basear-se em avaliações subjetivas.

Assim, durante todo processo de desenvolvimento deve haver uma seleção contínua de dados de produtividade e resultados, tanto do produto quanto do processo, para que as métricas sejam recolhidas.

Basicamente, gerar medição implica em utilizar os dados coletados e analisados pelos Gerentes de Projeto, comparando-os com os anteriores, tendo como resultado se houveram melhorias no projeto ou não. Caso haja áreas com problemas, podem ser tomadas medidas de solução (Pressman, 2002).

As métricas de software são:

- Diretas: medida realizada em atributos observados. Ex: custo, esforço, linhas de código, capacidade de memória, páginas, diagramas.
- Indiretas: Complexidade, eficiência, confiabilidade, facilidade de manutenção.
- Orientada ao Tamanho: Medida diretamente relacionada ao desenvolvimento, linha de código, tokens, memória.
- Orientada a função: Medida a ponto de vista do usuário, relacionada a funcionalidades, usabilidade.

O custo da qualidade inclui todos os custos decorrentes da busca da qualidade ou da execução das atividades relacionadas à qualidade (Pressman, 2002).

Alguns exemplos do que se podem ser medidos e comparados são os números de casos de teste efetuados em um projeto, o número de erros, defeitos, falhas encontrados pelos

analistas de teste em ambiente de homologação, análise de pontos de função, quantidade de mudança nos documentos de requisitos.

Apesar de ser uma ótima atividade para auxiliar no gerenciamento de um projeto, para as métricas de software não há um padrão específico, sendo subjetivas algumas vezes.

3.2 Análises de Ponto de Caso de Uso

As medidas de tamanho de software surgiram com o objetivo de estimar o esforço (número de pessoas-hora) e o prazo associados ao desenvolvimento dos programas e sistemas. Durante bastante tempo a principal medida utilizada foi a quantidade de linhas de código-fonte (*SLOC* – de *SourceLinesOfCode*), (Aguiar, 2003).

A *SLOC* aborda a medição física, sendo bastante limitada e realizada normalmente no final do desenvolvimento. Para melhores resultados, foram criadas novas formas para medir a funcionalidade do software, principalmente no início do projeto.

Surgiu então a Análise de Ponto de Função para medir o software, independente de linguagem ou tecnologia, abordando a funcionalidade do software na visão do usuário.

Em 1993 Gustav Karner criou uma variação dos Pontos de Função específica para a medição da funcionalidade contida em casos de uso. Nasceram então os *Use Case Points* (ou Pontos de Caso de Uso) (Aguiar, 2003).

A Análise de Ponto de Caso de Uso é uma metodologia utilizada para realizar estimativas de esforço, custo e tempo do desenvolvimento de um software, porém com a condição de que os documentos de requisitos sejam feitos como Caso de Uso. Um Caso deUso pode ser definido como sendo a especificação de um conjunto de ações executadas por um sistema e que apresenta um resultado observável e de valor para todo aquele que interage diretamente com o sistema (Meira, 2010, B).

Algumas fases no projeto de desenvolvimento de software devem ser seguidas para manter um controle do planejamento, por exemplo: análise de requisito e *design*, implementação, teste, implantação, manutenção.

A Análise de Requisitos é uma das mais importantes, por conter todo conteúdo que servirá de auxílio e consulta para as próximas fases. Nesta etapa, devem ser especificados os objetivos, interfaces, regras, escopo, para os desenvolvedores, analistas de teste e outras pessoas envolvidas diretamente com o sistema, terem uma visão clara do que deve ser implementado e do que o cliente necessita.

Utilizando a Análise de Ponto de Caso de Uso, o modelo da Análise Requisito deve ser o Modelo de Caso de Uso, segundo RUP, contém um modelo do comportamento de um sistema que suporta o processo de negócio. É de fácil compreensão, ilustrando realmente de que o sistema necessita, porém deve ser utilizado apenas quando há interação entre usuário e sistema. Ele é composto por dois tipos de documentos: o Diagrama de Caso de Uso e as Especificações de Caso de Uso (Meira, 2010, B).

Para a medição da aplicação como um todo, na Análise de Ponto de Caso de Uso são utilizados cálculos de Pontos de Caso de Uso, mensurando a complexidade do software, aplicando as seguintes avaliações: Cálculo do Peso dos Atores do sistema, Cálculo do Peso dos Casos de Uso, Cálculo dos Fatores de Ajuste (Técnicos e Ambientais), Cálculo do Porte do Sistema.

Profissionais capacitados aplicando, corretamente e com qualidade, a Análise de Ponto de Caso de Uso como métrica, a estimativa fica mais precisa, aumentando a confiança dos envolvidos no projeto, perante seus resultados, mesmo sabendo que as métricas não são totalmente exatas.

3.3 Análise de Ponto de Função e CoCoMo II

Estimar custo e tempo do projeto de desenvolvimento de software pode ser realizado com base em vários tipos de métricas de software, onde duas delas são: Análise de Ponto de Função e CoCoMo II (*ConstructiveCostModel*).

A Análise de Ponto de Função, por sua vez tem como objetivos analisar o nível de produtividade da equipe, o esforço de desenvolvimento de software, o custo de software, a taxa de produção e de manutenção de software. É uma técnica pode ser aplicada tanto no dimensionamento de projetos de aplicações já implantadas, quanto no dimensionamento de projetos de desenvolvimento ou manutenção de aplicações. (Vazques, 2008).

Esta métrica de software é independente de linguagem ou tecnologia, abordando a funcionalidade do software na visão do usuário, utilizável desde o início do projeto.

A contagem de ponto de função baseia-se em sete passos: Determinar o tipo de contagem de pontos de função, Identificar o escopo de contagem e a fronteira da aplicação, Determinar a contagem de pontos de função não ajustados, Contagem das funções de dados, Contagem das funções transacionais, Determinar o valor do fator de ajuste e Calcular os pontos de função ajustados, como exibido no seguinte diagrama:

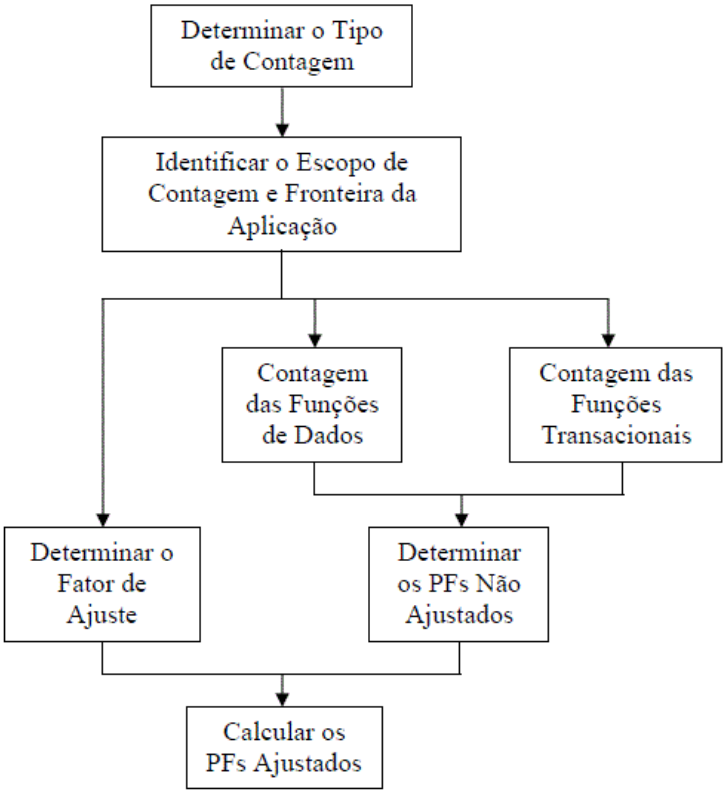

Figura 1 - O Procedimento de Contagem de Pontos de Função (Hazan, 2001).

A outra métrica é a CoCoMo, um modelo paramétrico de estimativa de custo de software que assume a existência de uma relação matemática entre tamanho, esforço e prazo.

Tal relação é afetada por parâmetros de performance. Os relacionamentos são baseados em suposições teóricas e/ou dados históricos (NASA, 2010).

O CoCoMo II surgiu após ser realizada uma análise de que o CoCoMo, onde o mesmo estava baseado em projetos muito antigos e não havia possibilidade de lidar com componentes e ciclos de vida iterativos.

Este modelo deriva dados a partir do tamanho funcional da aplicação, cobrindo áreas como negócios, controle, científica, suporte e sistema operacional. Tem como prioridade custo, tempo e qualidade.

Capítulo 4. Testes Unitários e JUnit

4.1 Testes Unitários

O teste unitário faz uso intensivo de técnicas de teste para exercitar caminhos específicos na estrutura de controle de determinado componente. Visa garantir a mais completa cobertura e a máxima detecção de erros.

Podemos defini-lo como: Estágio de teste onde o comportamento do componente de software é analisado isoladamente, restringindo o campo de atuação do teste evita-se a influência de outros componentes que fazem parte da aplicação (Celepar, 2009).

O teste unitário tem como objetivo manter foco no código do sistema ou na sua implementação.

Este teste baseia-se em validar separadamente as menores partes de código possíveis, verificando se o resultado esperado é igual ao que foi retornado, onde cada teste valida o seu respectivo método de forma independente.

Os testes unitários devem ser realizados antes ou durante o desenvolvimento do sistema, sendo realizado pelo programador. Este teste aumenta a cobertura de teste, previne teste de regressão, incentivam o refectoring, e podem agilizar o desenvolvimento.

Para serem realizados testes somente de um método que faz uso de recursos externos, devem ser utilizados Stubs e/ou Mocks, onde Mocks são utilizados para testar interação, simulando comportamentos de outros componentes e realizando comunicação entre objetos, e Stub para esboçar estado, remover e controlar dependências sobre outros serviços, em outras palavras: stubs providenciam respostas pré-configuradas para as chamadas feitas durante os testes, normalmente não respondem a nada que não esteja programado para o teste. Stubs também podem gravar informações sobre as chamadas, como um gateway que lembra as mensagens que 'enviou', ou talvez apenas quantas mensagens 'enviou'. Mocks são objetos pré-programados com informações que formam uma especificação das chamadas que esperam receber (Fowler, 2009).

Com o intuito de facilitar os testes unitários, existem ferramentas que realizam comparações mais complexas, testes com arrays, comparações de arquivos, tipos de dados, utilização de mocks, testes de métodos que acessam o banco de dados, entre outros. Uma dessas ferramentas é o JUnit.

4.2 JUnit

O JUnit é uma ferramenta de desenvolvimento utilizada para execução de testes unitários, a qual oferece a geração de relatórios relacionados a complexidade do sistema, abrangência dos testes, criação de *mocks*, detectar códigos duplicados, etc.

JUnit é um *framework* (conjunto de classes que colaboram para realizar uma responsabilidade para um domínio de um subsistema da aplicação (Fayad e Schmidt, 1997)) open-source que possibilita a criação de testes unitários em Java. A maioria das IDEs do mercado incorporam o JUnit dentro de seu ambiente de desenvolvimento, facilitando assim o uso desse *framework*.

Possibilita a criação das classes de testes, que contêm um ou mais métodos para que sejam realizados os testes, podendo ser organizados de forma hierárquica, de forma que o sistema seja testado em partes separadas, algumas integradas ou até mesmo todas de uma só vez. Além disso, este framework tem como objetivo, facilitar a criação de casos de teste, além de permitir escrever testes que retenham seu valor ao longo do tempo, ou seja, que possam ser reutilizáveis (JUnit, 2012).

O teste de unidade é importante na construção de métodos, pois permite ao programador testá-los durante a construção do sistema garantindo com isso a implementação de métodos livres de erros lógicos, que ocorrem com muita frequência. Por possibilitar os testes antes da conclusão do sistema, o programador pode testar seus métodos separadamente assim que eles estejam prontos. Isso evita que o programador fique percorrendo o código inteiro para descobrir os erros lógicos que apareciam quando o programa já estava pronto.

Capítulo 5. Definição da Técnica Desenvolvida

Serão explicadas e apresentadas, neste capítulo, as métricas e atividades realizadas para o desenvolvimento e especificação da técnica que permite avaliar uma estimativa de elaboração, execução e complexidade dos testes funcionais.

Depois de ser apresentada serão descritas amostras de sistemas, utilizados para auxiliar na definição da técnica, para comprovarmos o resultado esperado.

5.1 Técnica desenvolvida

5.1.1 Introdução à técnica

A técnica definida identificará o período de teste para escrita de casos de teste; escrita de cenários de teste, escrita de Script de teste; execução e log de teste.

Para isso, tem como principal objetivo avaliar uma estimativa de elaboração, execução e complexidade dos testes funcionais, estabelecendo o tempo de desenvolvimento do teste para cada funcionalidade do sistema.

Abaixo, seguem as atividades que serão desenvolvidas para definir a técnica e obter os resultados esperados.

5.1.2 Atividades da técnica

Para conseguir mensurar e avaliar a complexidade e o tempo de execução dos testes será necessário seguir algumas atividades que fornecerá resultados mais detalhados. São elas:

1º - Identificação dos Métodos que serão testados: Levantamento de todos os métodos do sistema que serão testados, para melhor analisá-los isoladamente.

2º - Número de Stub ou Mocks: Para realização de teste unitário, se houver mocks ou stubs, pode aumentar a complexidade do teste, portanto listar a quantidade dos mesmos.

3º - Identificação dos Cenários: Cada combinação a ser testada para cada método é considerado como um Cenário, assim, quanto mais cenários, o teste é mais complexo. Nesta atividade, verificar quantos cenários serão testados.

4º - Número de Integrações: Após testes unitários, são testadas integrações entre os métodos. Quanto mais integrações, mais complexo é o teste.

5º - Escrita do script para o cenário: Para cada cenário, realizar uma combinação dos dados de entrada, dados de saída e o resultado esperado, para realização do teste funcional manual.

6º - Aplicação do teste: Utilizar o script dos cenários, e aplicar no sistema de forma manual.

7º - Estudo do Resultado: Analisar se o resultado final foi igual ao resultado esperado.

Para realizar a mensuração, cada atividade e seu tempo serão classificados em minutos cheios: As atividades deverão ser classificadas como “Muito Simples”, “Simples”, “Média”, “Complexa” ou “Muito Complexa”, depois, a cada uma delas será atribuído um peso, conforme os resultados obtidos,para analisarmos a complexidade do teste. E para complementar, também será mensurado o tempo de cada atividade, para definirmos um tempo final de execução.

5.2 Aplicações da técnica desenvolvida em amostras

5.2.1 Primeira Amostra: Sistema de Provas Online

Nesta primeira amostra serão realizados testes funcionais em um Sistema de Provas Online chamado “TestNet Sistema Gerenciador de Testes Online”, aplicando as atividades citadas anteriormente.

O sistema foi implementado na linguagem PHP5 e com banco de dados MySQL 4, disponibilizado pelo ex-aluno do Centro Universitário Eurípides de Marília – UNIVEM, Fábio Hiromu Dalmazzo Nowaki, onde o desenvolveu como trabalho de conclusão de curso.

1º - Identificação dos Métodos que serão testados

a) Login
b) Cadastro de usuário
c) Adicionar nova instituição
d) Alterar dados
e) Cadastrar um novo curso
f) Cadastrar uma nova disciplina
g) Informações do professor
h) Cadastrar um novo aluno
i) Cadastrar uma nova pergunta
j) Agendar provas
k) Provas realizadas
l) Sair

2º - Número de Stub ou Mocks

- Nenhum

3º - Identificação dos Cenários

a) Login válido
   Login inválido
   Login em branco
   Quantidade de caracteres nos campos

b) Cadastro válido
   Cadastro com dados inválidos
   Cadastro em branco
   Quantidade de caracteres nos campos
   Verificar se foi salvo corretamente

c) Cadastro válido
   Cadastro inválido
   Campos em branco
   Campos obrigatórios
   Validação de Endereço
   Quantidade de caracteres nos campos
   Verificar se foi salvo corretamente

d) Editar deixando os campos em branco
   Editar informando campos inválidos
   Editar os dados e salvar novamente com sucesso
   Editar campos obrigatórios e não obrigatórios
   Deixar campos não obrigatórios em branco
   Verificar se foi salvo corretamente

e) Informar campo inválido

Informar campo em branco
Verificar quantidade de caracteres nos campos
Verificar se foi salvo corretamente

f) Cadastro válido
Cadastro inválido
Campos em branco
Campos obrigatórios
Quantidade de caracteres nos campos
Verificar apresentação nos campos selectbox
Verificar se foi salvo corretamente

g) Cadastro válido
Cadastro inválido
Campos em branco
Campos obrigatórios
Validação de endereço
Quantidade de caracteres nos campos
Validação de senha
Validação de e-mail
Verificar se foi salvo corretamente

h) Cadastro válido
Cadastro inválido
Campos em branco
Campos obrigatórios
Validação de endereço
Quantidade de caracteres nos campos
Validação de senha
Validação de e-mail
Verificar se foi salvo corretamente

i) Cadastro válido

Cadastro inválido

Campos em branco

Campos obrigatórios

Quantidade de caracteres nos campos

Verificar os selectbox se adiciona respectivos dados

Verificar se foi salvo corretamente

j) *Automática*:

Verificar curso e disciplina se são os mesmos cadastrados

Cadastro válido

Cadastro inválido

Campos em branco

Campos obrigatórios

Quantidade de caracteres nos campos

Máscara dos campos com data e hora

Verificar se a prova foi agendada automática

Verificar se as perguntas serão selecionadas aleatória e automaticamente

Manual:

Verificar se curso e disciplina são os mesmos cadastrados

Cadastro válido

Cadastro inválido

Campos em branco

Campos obrigatórios

Quantidade de caracteres nos campos

Máscara dos campos com data e hora

Verificar se é possível selecionar as perguntas

Perguntas não serão selecionadas aleatoriamente e automaticamente

k) Verificar relatório de provas realizadas

l) Clicar no link Sair

4º - Número de Integrações

a) Login e Cadastro de usuário
b) Curso e Disciplina
c) Curso e Pergunta
d) Disciplina, Professor e Curso
e) Pergunta, Curso e Disciplina
f) Prova, Curso e Disciplina
g) Disciplina e Aluno

5º - Escrita do script para o cenário

a) Login

	Login	**Senha**	**Resultado esperado**
1	Helena	123456	Login efetuado com sucesso
2	Naoexiste	Qualquercoisa	Login não efetuado e mensagem de erro
3	!@#$¨%	“@#$	Login não efetuado e mensagem de erro
4	-	-	Login não efetuado e mensagem de erro
5	Ultrapassar limite	Ultrapassar limite	Login não efetuado e mensagem de erro

b) Cadastro de usuário

	Digite seu nome completo	**Qual seu nível de ensino**	**Qual sua área de atuação**	**E-mail**	**Escolha seu login**	**Senha**	**Nome da Instit.**	**Result. esperado**
1	Ana Mariz	Superior	Computação	ana@teste.com	aninha	12345	Instit I	Cadastro realizado
2	“!@#$¨¨	Nenhum	Física	aaa.teste.com	“!@#$	0000	“!@#$%	Mensagem de erro
3								Mensagem de erro
4	Passa limite			Passa limite	Passa limite	Passa limite	Passa limite	Mensagem de erro
5	Tais Cayu	Médio	Outras	tais@teste.com	tais.cayu	112233	Instit II	Cadastro salvo

	Nome	**Endereço**	**Nº**	**Bairro**	**Compl**	**Cidade**	**Estado**	**Resultado esperado**
1	Instituição I	Rua ABC	23	Solares	Apto 12	Marília	SP	Cadastro Realizado
2	“!@#$%¨	“!@#$%¨	Aa	“!@#$%	‘#$&%	“!@#$%¨&	RRRR	Mensagem de erro

3								Mensagem de erro
4	Instituição II							Cadastro Realizado
5	Instituição III	00000	Cc	00-00	00-00	000000000	TTTTT	Mensagem de erro
6	passa limite	passa limite		passa limite	passa limite	passa limite	passa limite	Mensagem de erro
7	Instituição IV	Rua DEF	40	Montanha		São Paulo	SP	Cadastro Salvo

c) Adicionar nova instituição

d) Alterar dados

	Nome	**Endereço**	**Nº**	**Bairro**	**Compl**	**Cidade**	**Estado**	**Resultado esperado**
1								Mensagem de erro
2	“!@#$%¨*	“!@#$%	!!	“!@#$%¨	“!@#$%¨	(¨%	#$%	Mensagem de erro
3	Instituição VI	Rua GHI	50	Bairro	Compl	São Paulo	SP	Editado e Salvo
4	Instituição VII							Editado e Salvo

e) Cadastrar um novo curso

	Nome	**Resultado esperado**
1	“!@#$%¨&*	Mensagem de erro
2		Mensagem de erro
3	Ultrapassar limite	
4	Computação	Salvo com sucesso

f) Cadastrar uma nova disciplina

	Nome	**Curso**	**Professor**	**Resultado esperado**
1	Cálculo	Computação	José Ricardo	Cadastrado com sucesso
2	“!@#$%¨	Não informar	Não informar	Mensagem de erro
3				Mensagem de erro
4	-	-	-	Mensagem de erro
5	Cálculo II	Verificar cursos cadastrados	Verificar profs cadastrados	Devem ser recuperados dados já cadastrado do Curso e do Professor

g)Informações do professor

	Nome	E-mail	Login	Senha	Ativo/Inativo
1	Maria Alini	maria@teste.com	maria.alini	123456	Ativo
2	“!@#$%¨*	maria.com	0000	0000	Nenhum
3					
4	Ana Lucia	ana@teste.com	ana.lucia	67890	Inativo
5	Ultrapassar Limite	000111aaa	Ultrapassar Limite	Ultrapassar Limite	Nenhum

End	Nº	Bairro	Compl	Cidade	Estado	Resultado Esperado
Rua Flores	40	Bairro	Compl	Cidade	SP	Cadastro realizado com sucesso
000	Aa	0000				Mensagem de erro
						Mensagem de erro
						Cadastro realizado com sucesso
Ultrapassar Limite	...	Ultrapassar Limite	Ultrapassar Limite			Mensagem de erro

h)Cadastrar um novo aluno

	Nome	RA	Senha	E-mail	Ativo/Inativo
1	João Pedro	98765	Abc1234	joao@teste.com	Ativo
2	“!@#$%¨*	00000	0000	Joao.com	Nenhum
3					
4	Ana Lucia	6789	Abcd	ana@teste.com	Inativo
5	Ultrapassar Limite	000111aaa	Ultrapassar Limite	Ultrapassar Limite	Nenhum
6	Paulo José	112233	1111111111	paulo@teste.com	Ativo

End	Nº	Bairro	Compl	Cidade	Estado	Resultado Esperado
Rua Flores	40	Bairro	Compl	Cidade	SP	Cadastro realizado com sucesso
000	Aa	0000				Mensagem de erro
						Mensagem de erro
						Cadastro realizado com sucesso
Ultrapassar Limite	...	Ultrapassar Limite	Ultrapassar Limite			Mensagem de erro
						Mensagem de erro de senha

i)Cadastrar uma nova pergunta

	Curso	Disciplina	Tópico	Tipo Perg	Dificuldade
1	Escolher conforme já cadastrado	Escolher conforme já cadastrado	Escolher conforme já cadastrado	Dissertativa	Fácil
2	-	-	-	-	-
3					
4				Relacione	Difícil
5	Escolher conforme já cadastrado	Escolher conforme já cadastrado	Escolher conforme já cadastrado	Verdadeiro ou Falso	Médio

Ativa/ Inativa	Fonte	Título Pergunta	Respostas	Resultado Esperado
Ativa	Livro X	O que significa HD?	Hard Disk	Mensagem de sucesso e salvo com sucesso
-	“!@#$%¨&	“!@#$%¨&*()_	“!@#$%	Mensagem de erro
				Mensagem de erro
Inativa				Mensagem de erro
Ativa	Ultrapassar Limite	Ultrapassar Limite	Ultrapassar Limite	Mensagem de erro

j)Agendar provas

Automática:

	Curso	Disciplina	Título da Prova
1	Escolher curso já cadastrado	Escolher disciplina já cadastrada	Cálculo II
2	-	-	“!@#$%¨&*(
3	-	-	
4			Teste
5	Verificar curso já cadastrado	Verificar disciplina já cadastrada	passar limite
6	Escolher curso já cadastrado	Escolher curso já cadastrado	Álgebra

Nº de Perguntas	Dificuldade da Prova	Início	Término	Resultado esperado
10	Fácil	01/01/2012 13:30	01/01/2012 18:30	Mensagem de sucesso e agendada
Abc	-	99/99/9999 99:99	00/00/0000 00:00	Mensagem de erro
				Mensagem de erro
				Mensagem de erro
Ultrapassar limite	Ultrapassar limite	-	-	Mensagem de erro
5	Médio	Verificar Máscara	Verificar Máscara	Mensagem de erro

Verificar se as perguntas serão selecionadas aleatória e automaticamente

Manual:

	Curso	Disciplina	Título da Prova
1	Escolher curso já cadastrado	Escolher disciplina já cadastrada	Cálculo II
2	-	-	"!@#$%¨&*(
3	-	-	
4			Teste
5	Verificar curso já cadastrado	Verificar disciplina já cadastrada	Ultrapassar limite
6	Escolher curso já cadastrado	Escolher curso já cadastrado	Álgebra

Início	Término	Resultado esperado
01/01/2012 13:30	01/01/2012 18:30	Mensagem de sucesso e agendada
99/99/9999 99:99	00/00/0000 00:00	Mensagem de erro
		Mensagem de erro
		Mensagem de erro
-	-	Mensagem de erro
Verificar Máscara	Verificar Máscara	Mensagem de erro

k)Provas realizadas: Clicar em "Provas" > "Provas Realizadas": Verificar relatório de provas realizadas.

l)Sair: Clicar no link "Sair" e o logout é realizado.

6º - Aplicação do teste

Todos os cenários descritos na 5º atividade foram aplicados no sistema, realizando o teste funcional manual do mesmo. Essa atividade teve duração de 233 minutos, classificando-se como muito complexa.

7º - Estudo do Resultado

Para analisar o resultado do teste realizado nesta primeira amostra, são informados: o tempo e a quantidade de itens de cada atividade e classificá-las como MUITO SIMPLES, SIMPLES, MÉDIA, COMPLEXA, MUITO COMPLEXA, depois, a cada uma delas é atribuído um peso conforme os resultados obtidos, para definirmos a complexidade do teste.

Atividade	Itens	Tempo	Classificação
1º - Identificação dos métodos que serão testados	12	25m	MÉDIA
2º - Número de Stub ou Mocks	0	6m	SIMPLES
3º - Identificação dos Cenários	78	57m	COMPLEXA
4º - Número de Integrações	7	11m	MÉDIA
5º - Escrita do script para o cenário	88	166m	MUITO COMPLEXA
6º - Aplicação do teste	88	233m	MUITO COMPLEXA

O tempo total gasto para execução do teste funcional manual do sistema de provas online chamado “TestNet Sistema Gerenciador de Testes Online”, foi de 498 minutos o equivalente há aproximadamente 9 horas.

5.2.2 Segunda Amostra: Sistema de Gerenciamento de Compras Coletivas

Nesta segunda amostra serão realizados testes funcionais em um sistema de gerenciamento de compras coletivas chamado “GroupBuy”, aplicando as atividades citadas anteriormente.

O sistema foi implementado na linguagem PHP e com banco de dados MySQL, disponibilizado no site http://groupbuy.sourceforge.net/, e registrado no SourceForge.net em 22 de janeiro de 2007.

1º - Identificação dos Métodos que serão testados

a) Login do usuário
b) Recuperação de senha
c) Cadastro do usuário
d) Relatório de oferta atual
e) Relatório de ofertas válidas
f) Relatório que ofertas que falharam
g) Cadastrar nova oferta
h) Relatório de pedido atual
i) Relatório de pedido pago
j) Relatório de pedido não pago
k) Relatório de cupom não utilizado
l) Relatório de cupom usado
m) Relatório de cupom expirado
n) Relatório de ticket
o) Novo ticket
p) Lista de parceiros
q) Cadastrar novo parceiro
r) Cadastrar cidades da categoria
s) Cadastrar ofertas da categoria
t) Cadastrar usuários da categoria

2º - Número de Stub ou Mocks

- Nenhum

3º - Identificação dos Cenários

a) Login válido
Login inválido
Login em branco
Login com e-mail e sem senha
Login com senha e sem e-mail

b) Recuperação válida
   Recuperação com e-mail em branco
   Recuperação com e-mail inválido
   Recuperação com e-mail não existente no sistema

c) Cadastro válido
   Cadastro com e-mail inválido
   Cadastro com usuário fora do limite de caracteres
   Cadastro com senha inválida
   Cadastro com confirmação de senha inválida
   Obrigatoriedade dos campos
   Cadastro com todos os campos em branco

d) Exibição em branco
   Exibição correta dos campos do relatório
   Exibição dos detalhes
   Edição do item
   Apagar o item

e) Exibição em branco
   Exibição correta dos campos do relatório
   Exibição dos detalhes
   Edição do item
   Apagar o item

f) Exibição em branco
   Exibição correta dos campos do relatório
   Exibição dos detalhes
   Edição do item
   Apagar o item

g) Cadastro válido

Cadastro inválido

Cadastro com campos em branco

Verificar campos obrigatórios

Verificar os selectbox

Verificar campos com restrições

h) Exibição em branco

Exibição correta dos campos do relatório

Exibição dos detalhes

Edição do item

Apagar o item

Filtrar por e-mail

i) Exibição em branco

Exibição correta dos campos do relatório

Exibição dos detalhes

Edição do item

Apagar o item

Filtrar por e-mail

j) Exibição em branco

Exibição correta dos campos do relatório

Exibição dos detalhes

Edição do item

Apagar o item

Filtrar por e-mail

k) Exibição em branco

Exibição correta dos campos do relatório

l) Exibição em branco

Exibição correta dos campos do relatório

m) Exibição em branco
Exibição correta dos campos do relatório

n) Exibição em branco
Exibição correta dos campos do relatório
Exibição dos detalhes
Edição do item
Apagar o item
Validação dos filtros
Realizar download em Excel

o) Cadastro válido
Cadastro inválido
Cadastro com campos em branco
Verificar campos obrigatórios
Verificar campos com datas
Verificar campos com restrições

p) Exibição em branco
Exibição correta dos campos do relatório
Edição do item
Apagar o item

q) Cadastro válido
Cadastro inválido
Cadastro com campos em branco
Verificar campos obrigatórios

r) Exibição em branco
Exibição correta dos campos do relatório
Edição do item
Apagar o item
Cadastro válido

Cadastro inválido
Cadastro com campos em branco
Verificar campos obrigatórios
Clicar em ‘Fechar’

s) Exibição em branco
Exibição correta dos campos do relatório
Edição do item
Apagar o item
Cadastro válido
Cadastro inválido
Cadastro com campos em branco
Verificar campos obrigatórios
Clicar em ‘Fechar’

t) Exibição em branco
Exibição correta dos campos do relatório
Edição do item
Apagar o item
Cadastro válido
Cadastro inválido
Cadastro com campos em branco
Verificar campos obrigatórios
Clicar em ‘Fechar’

4º - Número de Integrações

a) Oferta e Categoria
b) Oferta e Cidade
c) Oferta e Parceiro
d) Cupom e Parceiro

5º - Escrita do script para o cenário

a) Login do usuário

	E-mail	Senha	Resultado esperado
1	helenamcabrini@gmail.com	123456	Login efetuado com sucesso
2	helenamcabrini@gmail.com	senhainvalida	Mensagem de erro
3	-	-	Os campos obrigatórios são destacados
4	helenamcabrini@gmail.com	-	O campo senha é destacado e não faz login
5	-	123456	O campo e-mail é destacado e não faz login

b) Recuperação de senha

	E-mail	Resultado esperado
1	helenamcabrini@gmail.com	Recuperação realizada com sucesso
2	-	Mensagem de erro
3	testes.teste.com	Mensagem de erro
4	naoexiste@teste.com	Mensagem de erro

c) Cadastro de Usuário

	E-mail	Usuário	Senha	Confirme a Senha	Celular	Resultado esperado
1	teste1@teste.com	testador	123456	123456	99998888	Cadastro realizado
2	teste2.com	Teste	123456	123456	88889999	Campo e-mail destacado
3	teste2@teste.com	A	123456	123456	88887777	Campo usuário destacado
4	teste3@teste.com	Teste3	1	123456	99998888	Campo senha destacado
5	teste3@teste.com	Teste3	123456	1	99998888	Campo confirmação de senha destacado
6	teste4@teste.com	Teste4	123456	123456		Cadastro realizado
7	-	-	-	-	-	Todos campos destacados

d) Relatório de oferta atual

	Campos	Operação	Resultado esperado
1	ID, Item, Categoria, Data, Completa, Preço.	Detalhes, Editar, Deletar.	Todos os campos em branco
2	ID, Item, Categoria, Data, Completa, Preço.	Detalhes, Editar, Deletar.	Campos preenchidos corretamente
3	-	Clicar em Detalhes	Exibidas todas as informações da Oferta Atual
4	-	Clicar em Editar	Exibido o cadastro de oferta com os campos preenchidos conforme item escolhido

5	-	Clicar em Deletar	Mensagem de confirmação e apaga item

e) Relatório de ofertas válidas

	Campos	Operação	Resultado esperado
1	ID, Item, Categoria, Data, Completa, Preço.	Detalhes, Editar, Deletar.	Todos os campos em branco
2	ID, Item, Categoria, Data, Completa, Preço.	Detalhes, Editar, Deletar.	Campos preenchidos corretamente
3	-	Clicar em Detalhes	Exibidas todas as informações da Oferta Atual
4	-	Clicar em Editar	Exibido o cadastro de oferta com os campos preenchidos conforme item escolhido
5	-	Clicar em Deletar	Mensagem de confirmação e apaga item

f) Relatório que ofertas que falharam

	Campos	Operação	Resultado esperado
1	ID, Item, Categoria, Data, Completa, Preço.	Detalhes, Editar, Deletar.	Todos os campos em branco
2	ID, Item, Categoria, Data, Completa, Preço.	Detalhes, Editar, Deletar.	Campos preenchidos corretamente
3	-	Clicar em Detalhes	Exibidas todas as informações da Oferta Atual
4	-	Clicar em Editar	Exibido o cadastro de oferta com os campos preenchidos conforme item escolhido
5	-	Clicar em Deletar	Mensagem de confirmação e apaga item

g) Cadastrar nova oferta

	Cidade&Categoria	Título da Oferta	Preço Normal	Preço Com	Mínimo	Desconto Máximo	Máximo por Pessoa
1	Escolher corretamente	Notebook com 50% de desconto	2000	1000	10	0	1
2	-	-	1000	2000	0	-1	0
3	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-
5	Verificar se são exibidos conforme cadastrados	-	-	-	-	-	-

	Data Início	Data Fim	Validade	Introdução	Regulamento
1	03/11/2012	04/11/2012	03/04/2013	Texto válido	Texto válido

2	04/11/2012	03/11/2012	01/01/2011	Qualquer texto	Qualquer texto
3	-	-	-	-	-
4	-	-	-	-	-
5	-	-	-	-	-

	Empresa Parceira	**Ticket**	**Nome do Item**	**Imagens**	**FLV Vídeo**	**Detalhes da Oferta**	**Coment**	**GB Dicas**
1	Parceiro 1	1000	Item 1	Anexar Imagem	-	Texto válido	Texto válido	Texto válido
2	Não Informar	0	Não informa r	Não anexar	-	Qualquer texto	Qualquer texto	Qualqu er texto
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-
5	Verificar se são exibidos dados cadastrados como Parceiros	-	-	-	-	-	-	-

	Entrega	**Desconto**	**Resultado Esperado**
1	Cupom	10	Cadastro realizado com sucesso
2		aaa	Mensagem de erro e campos em destaque para serem preenchidos
3	-	-	Mensagem de erro e campos em destaque para serem preenchidos
4	-	-	Mensagem de erro. Ir informando os campos e verificando os obrigatórios.
5	-	-	Os selectbox devem ser preenchidos conforme cadastrados.

h) Relatório de pedido atual

	E-mail	**Campos**	**Operação**	**Resultado esperado**
1	-	ID, Item, Usuário, Quantidade, Total, Total não pago, Pagar.	Dinheiro? PagSeguro?	Todos os campos em branco
2	-	ID, Item, Usuário, Quantidade, Total, Total	Dinheiro? PagSeguro?	Campos preenchidos corretamente

		não pago, Pagar.		
3	-	-	Clicar em Dinheiro	Exibe mensagem de confirmação
4	-	-	Clicar PagSeguro	Exibe mensagem de confirmação
5	Informar um e-mail	ID, Item, Usuário, Quantidade, Total, Total não pago, Pagar.	-	Serão exibidos somente itens relacionados ao e-mail informado

i) Relatório de pedido pago

	E-mail	**Campos**	**Operação**	**Resultado esperado**
1	-	ID, Oferta, Usuário, Quantidade, Total, Total não pago, Pagar.	Detalhes	Todos os campos em branco
2	-	ID, Oferta, Usuário, Quantidade, Total, Total não pago, Pagar.	Detalhes	Campos preenchidos corretamente
3	-	-	Clicar em Detalhes	Exibidas todas as informações do Pedido Pago
4	Informar um e-mail	ID, Oferta, Usuário, Quantidade, Total, Total não pago, Pagar.	-	Serão exibidos somente itens relacionados ao e-mail informado

j) Relatório de pedido não pago

	E-mail	**Campos**	**Operação**	**Resultado esperado**
1	-	ID, Oferta, Usuário, Quantidade, Total, Total não pago, Pagar.	Detalhes, Editar, Deletar.	Todos os campos em branco
2	-	ID, Oferta, Usuário, Quantidade, Total, Total não pago, Pagar.	Detalhes, Editar, Deletar.	Campos preenchidos corretamente
3	-	-	Clicar em PagSeguro	Exibida mensagem de confirmação
4	Informar um e-mail	ID, Oferta, Usuário, Quantidade, Total, Total não pago, Pagar.	-	Serão exibidos somente itens relacionados ao e-mail informado

k) Relatório de cupom não utilizado

	Campos	**Resultado esperado**
1	Código, Item, Usuário, Validade.	Todos os campos em branco
2	Código, Item, Usuário, Validade.	Campos preenchidos corretamente

l) Relatório de cupom usado

	Campos	**Resultado esperado**

1	Código, Item, Usuário, Data.	Todos os campos em branco
2	Código, Item, Usuário, Data.	Campos preenchidos corretamente

m) Relatório de cupom expirado

	Campos	Resultado esperado
1	Serial, Item, Usuário, Data.	Todos os campos em branco
2	Serial, Item, Usuário, Data.	Campos preenchidos corretamente

n) Relatório de ticket

	Oferta ID	Parceiro ID	Código	Status	Campos	Operação	Resultado esperado
1	-	-	-	-	ID, Valor ID, Válido, Status, Parceiro.	Deletar	Todos os campos em branco
2	-	-	-	-	ID, Valor ID, Válido, Status, Parceiro.	Deletar	Campos preenchidos corretamente
3	-	-	-	-	-	Clicar em Deletar	Exibida mensagem de confirmação
4	1	-	-	Used	ID, Valor ID, Válido, Status, Parceiro.	-	Serão exibidos somente itens relacionados ao filtro informado
5	-	2	3	Used	ID, Valor ID, Válido, Status, Parceiro.	-	Serão exibidos somente itens relacionados ao filtro informado
6	-	-	-	Unused	ID, Valor ID, Válido, Status, Parceiro.	-	Serão exibidos somente itens relacionados ao filtro informado
7	-	-	-	-	Clicar em Download	-	Aberto Excel com o filtro informado.

o) Novo ticket

	Parceiro ID	Valor do Ticket	Quant	Válido de	Válido até	Código	Resultado esperado
1	1	10	100	01-11-2012	30-11-2012	2103_GM	Cadastro salvo
2	10000	0	3000	30-11-2012	01-11-2012	aaa	Campos inválidos destacados

3	-	-	-	-	-	-	Campos em branco destacados
4							Ir informando um campo de cada vez. Todos são obrigatórios.

p) Lista de parceiros

	Campos	Operação	Resultado esperado
1	ID, Nome, E-mail, Tel., Data.	Editar, Deletar.	Todos os campos em branco
2	ID, Nome, E-mail, Tel., Data.	Editar, Deletar.	Campos preenchidos corretamente
3	-	Clicar em Editar.	Exibido o cadastro de parceiros com os campos preenchidos conforme item escolhido
4	-	Clicar em Deletar	Mensagem de confirmação e apaga item

	Usuário	Senha	Nome	Site	E-mail	Tel.	End	Celular
1	User	123456	User1	www.teste.com	teste@teste.com	33331111	Av AB, 3	99991111
2	"!@#$%¨&*()_+	"!@#$%¨&*()_+	"!@#$%¨&*()_+	"!@#$%¨&*()_+	"!@#$%¨&*()_+	"!@#$%¨&*()_+	"!@#$%¨&*()_+	"!@#$%¨&*()_+
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-

q) Cadastrar novo parceiro

	Localização	Outras Inf.	Nome do Banco	Nome do Cliente	Agência e Conta	Resultado Esperado
1	Qualquer texto	Qualquer texto	Bradesco	Nome Teste	0002 1111	Cadastro efetuado
2	"!@#$%¨&*()_+	"!@#$%¨&*()_+	"!@#$%¨&*()_+	"!@#$%¨&*()_+	"!@#$%¨&*()_+	Mensagem de erro
3	-	-	-	-	-	Campos destacados
4	-	-	-	-	-	Ir informando um campo de cada vez. Todos são obrigatórios.

r) Cadastrar cidades da categoria

	Nome	Minúsculo	Primeira	Categoria	Resultados Esperados

			Letra		
1	Ana	ana	A	Teste	Cadastrado e exibe na listagem
2	"!@#$%¨&*(	"!@#$%¨&*(	12345	Teste	Não deixa cadastrar com mais de uma letra em “Primeira Letra”
3	-	-	-	-	Mensagem de erro
4	Teste	-	-	-	Mensagem de erro
5	-	-	-	-	Clicar em “Fechar” esconde a janela

s) Cadastrar ofertas da categoria

	Nome	**Minúsculo**	**Primeira Letra**	**Categoria**	**Resultados Esperados**
1	Teste	teste	A	Teste	Cadastrado e exibe na listagem
2	"!@#$%¨&*(	"!@#$%¨&*(	12345	Teste	Não deixa cadastrar com mais de uma letra em “Primeira Letra”
3	-	-	-	-	Mensagem de erro
4	Teste	-	-	-	Mensagem de erro
5	-	-	-	-	Clicar em “Fechar” esconde a janela

t) Cadastrar usuários da categoria

	Nome	**Minúsculo**	**Primeira Letra**	**Categoria**	**Resultados Esperados**
1	Teste	teste	A	Teste	Cadastrado e exibe na listagem
2	"!@#$%¨&*(	"!@#$%¨&*(	12345	Teste	Não deixa cadastrar com mais de uma letra em “Primeira Letra”
3	-	-	-	-	Mensagem de erro
4	Teste	-	-	-	Mensagem de erro
5	-	-	-	-	Clicar em “Fechar” esconde a janela

6º - Aplicação do teste

Todos os cenários descritos na 5º atividade foram aplicados no sistema, realizando o teste funcional manual do mesmo. Essa atividade teve duração de 151 minutos, classificando-se como complexa.

7º - Estudo do Resultado

Para analisar o resultado do teste realizado nesta segunda amostra, são informados: o tempo e a quantidade de itens de cada atividade e classificá-las como MUITO SIMPLES, SIMPLES, MÉDIA, COMPLEXA, MUITO COMPLEXA, depois, a cada uma delas é atribuído um peso conforme os resultados obtidos, para definirmos a complexidade do teste.

Atividade	Itens	Tempo	Classificação
1º - Identificação dos métodos que serão testados	20	17m	MÉDIA
2º - Número de Stub ou Mocks	0	4m	SIMPLES
3º - Identificação dos Cenários	108	41m	COMPLEXA
4º - Número de Integrações	4	12m	MÉDIA
5º - Escrita do script para o cenário	108	155m	MUITO COMPLEXA
6º - Aplicação do teste	108	151m	COMPLEXA

O tempo total gasto para execução do teste funcional manual do sistema de gerenciamento de compras coletivas chamado “GroupBuy”, foi de 388 minutos o equivalente há aproximadamente 7 horas.

5.2.3 Terceira Amostra: Sistema de Gerenciamento de senhas

Nesta terceira amostra serão realizados testes funcionais em um sistema de gerenciamento de senhas chamado “*PHP Password Manager*”, aplicando as atividades citadas anteriormente.

O sistema foi implementado na linguagem PHP e com banco de dados MySQL, disponibilizado no site http://sourceforge.net/projects/ppma/ registrado no dia 07 de outubro de 2009.

1º - Identificação dos Métodos que serão testados

a) Login
b) Visualização da senha
c) Criação da senha
d) Editar senha
e) Excluir senha

f) Visualização da tag
g) Criação da tag
h) Editar tag
i) Excluir tag
j) Mudar senha
k) Sair

2º - Número de Stub ou Mocks

- Nenhum

3º - Identificação dos Cenários

a) Login válido
Login inválido
Login em branco
Quantidade de caracteres nos campos

b) Exibição dos itens
Verificação dos campos exibidos

c) Cadastro válido
Cadastro com dados inválidos
Cadastro em branco
Quantidade de caracteres nos campos
Verificar visualização da senha
Verificar se foi salvo corretamente

d) Validar botão de edição
Verificar visualização da senha
Modificar os dados e salvar
Verificar atualização

e) Validar botão de exclusão
   Validar mensagem de confirmação
   Clicar no ok apaga a senha
   Clicar em cancelar não apaga a senha

f) Exibição dos itens
   Verificação dos campos exibidos

g) Cadastro válido
   Cadastro com dados inválidos
   Cadastro em branco
   Quantidade de caracteres no campo
   Verificar se foi salvo corretamente

h) Validar botão de edição
   Modificar os dados e salvar
   Verificar atualização

i) Validar botão de exclusão
   Validar mensagem de confirmação
   Clicar no ok apaga a senha
   Clicar em cancelar não apaga a senha

j) Senha antiga inválida e nova válida
   Senha antiga válida e nova inválida
   Senha antiga válida, nova válida e confirmação da nova divergente
   Campos em branco
   Quantidade de caracteres inválidos

k) Clicar no logout

4º - Número de Integrações

a) Login e Mudança de senha
b) Senhas e Tag

5º - Escrita do script para o cenário

a) Login

	Username	Password	Resultado esperado
1	helenamcabrini@gmail.com	123456	Login efetuado com sucesso
2	helenamcabrini@gmail.com	senhainvalida	Mensagem de erro
3	-	-	Mensagem de erro
4	helenamcabrini@gmail.com	-	Mensagem de erro
5	-	123456	Mensagem de erro

b) Visualização da senha

	Ação	Resultado esperado
1	Verificar quantidade de itens cadastrados	Todos os itens devem ser exibidos
2	Verificar campos	Os campos devem estar preenchidos corretamente conforme indica o cadastro

c) Criação da senha

	Name	Username	Password	URL	Tag	Comment	Resultado esperado
1	Ana Maria	ana.maria	123456	teste.com	Compras	Qualquer Texto	Cadastrado realizado
2	"!@#$%¨ &*(	"!@#$%¨ &*(	.....	teste	----	----	Mensagem de erro
3	-	-	-	-	-	-	Mensagem de erro
4	Ultrapassa limite	Ultrapassa limite	Ultrapassa limite	Ultrapassa limite	Ultrapassa limite	Ultrapassa limite	Mensagem de erro
5			123456				Ao clicar no botão, exibe a senha para leitura

d) Editar senha

	Ação	Resultado esperado
1	Clica no botão para editar	Abre formulário de atualização com os dados preenchidos
2	Clica no botão de visualizar senha	A senha é exibida para leitura

3	Modificar dados e salvar	Os dados são atualizados e salvos

e) Excluir senha

	Ação	Resultado esperado
1	Clicar no botão para apagar	Exibe mensagem de confirmação
2	Clicar em “Cancelar”	Não é apagada a senha
3	Clicar em “Ok”	É excluído o registro da senha

f) Visualização da Tag

	Ação	Resultado esperado
1	Verificar quantidade de itens cadastrados	Todos os itens devem ser exibidos
2	Verificar campos	Os campos devem estar preenchidos corretamente conforme indica o cadastro

g) Criação da Tag

	Nome	Resultado esperado
1	Nome da Tag	Cadastro realizado
2	“!@#$%¨&*()	Mensagem de erro
3	-	Mensagem de erro
4	Ultrapassar limite de caracter	Mensagem de erro
5	Nome da Tag2	Cadastro realizado e gravado registro

h) Editar Tag

	Ação	Resultado esperado
1	Clica no botão para editar	Abre formulário de atualização com os dados preenchidos
2	Modificar dados e salvar	Os dados são atualizados e salvos

i) Excluir Tag

	Ação	Resultado esperado
1	Clicar no botão para apagar	Exibe mensagem de confirmação
2	Clicar em “Cancelar”	Não é apagada a tag
3	Clicar em “Ok”	É excluído o registro da tag

j) Mudar senha

	OldPassword	New Password	RepeatPassword	Resultado esperado
1	Senhainvalida	123456	123456	Não troca senha
2	123456	invalida	Invalida	Não troca senha
3	123456	78901	1234	Não troca senha
4	-	-	-	Não troca senha
5	123456	67890	67890	Troca senha

k) Sair

Ao clicar em “Logout” o sistema volta para tela de login.

6º - Aplicação do teste

Todos os cenários descritos na 5º atividade foram aplicados no sistema, realizando o teste funcional manual do mesmo. Essa atividade teve duração de 25 minutos, classificando-se como simples.

7º - Estudo do Resultado

Para analisar o resultado do teste realizado nesta terceira amostra, são informados: o tempo e a quantidade de itens de cada atividade e classifica-las como MUITO SIMPLES, SIMPLES, MÉDIA, COMPLEXA, MUITO COMPLEXA, depois, a cada uma delas é atribuído um peso conforme os resultados obtidos, para definirmos a complexidade do teste.

Atividade	Itens	Tempo	Classificação
1º - Identificação dos métodos que serão testados	11	6m	SIMPLES
2º - Número de Stub ou Mocks	0	3m	SIMPLES
3º - Identificação dos Cenários	40	18m	SIMPLES
4º - Número de Integrações	2	4m	SIMPLES
5º - Escrita do script para o cenário	36	29m	SIMPLES
6º - Aplicação do teste	36	25m	SIMPLES

O tempo total gasto para execução do teste funcional manual do sistema de gerenciamento de senhas “PHP Password Manager” foi de 85 minutos o equivalente há aproximadamente 2 horas.

5.2.4 Quarta Amostra: Sistema de Agenda Telefônica

Nesta quarta amostra serão realizados testes funcionais em um sistema de agenda telefônica chamado “Address Book”, aplicando as atividades citadas anteriormente.

O sistema foi implementado na linguagem PHP e com banco de dados MySQL, disponibilizado no site http://sourceforge.net/projects/php-addressbook/ registrado no dia 19 de janeiro de 2006.

1º - Identificação dos Métodos que serão testados

a) Login
b) Criar conta
c) Adicionar endereço
d) Editar e deletar endereço
e) Adicionar grupos
f) Próximos aniversários
g) Impressão dos endereços
h) Impressão de telefones
i) Logout

2º - Número de Stub ou Mocks

- Nenhum

3º - Identificação dos Cenários

a) Login válido
   Login inválido
   Login em branco

Quantidade de caracteres nos campos

b) Cadastro válido
   Cadastro inválido
   Cadastro com campos em branco
   Verificar campos obrigatórios

c) Cadastro válido
   Cadastro inválido
   Cadastro com campos em branco
   Verificar campos obrigatórios
   Verificar campos com datas
   Verificar upload de imagem

d) Validar botão Editar
   Validar botão Deletar

e) Cadastro válido
   Cadastro inválido
   Cadastro com campos em branco
   Verificar campos obrigatórios
   Deletar grupo
   Editar grupo

f) Adicionar endereço e verificar relatório de aniversários

g) Verificar tela de impressão

h) Verificar exibição de todos os telefones

i) Clicar em Logout, volta na tela de login

4º - Número de Integrações

a) Endereço e Grupo

5º - Escrita do script para o cenário

a) Login

	User	Password	Resultado esperado
1	helenamcabrini@gmail.com	123456	Login efetuado com sucesso
2	helenamcabrini@gmail.com	senhainvalida	Mensagem de erro
3	-	-	Mensagem de erro
4	helenamcabrini@gmail.com	-	Mensagem de erro
5	-	123456	Mensagem de erro

b) Criar conta

	User	Password	Resultado esperado
1	helenamcabrini@gmail.com	123456	Conta criada com sucesso
2	helenamcabrini@gmail.com	senhainvalida	Mensagem de erro
3	-	-	Mensagem de erro
4	helenamcabrini@gmail.com	-	Mensagem de erro
5	-	123456	Mensagem de erro

c) Adicionar endereço

	Address	Firstname	Middlename	Lastname	Nickname	Photo
1	Rua das Flores1	Helena	Monici	Cabrini	Lena	Adicionar
2	!#$%¨&*()_	“!@#$%¨&*()	“!@#$%	¨”!@#$%¨*	“!@#$%¨&	-
3	-	-	-	-	-	-
4	Rua Teste 4	Ana	Maria	Souza	Aninha	Adicionar
5	Rua Teste 2	José		Oliveira	Zé	Verificar

	Title	Company	Homepage	e-mail
1	Teste	Empresa	www.teste.com	helena@teste.com
2	“!@#$%	“!@#$%	Teste	Teste
3	-	-	-	-
4	Teste	Teste	www.teste.com	teste@teste.com

5	-	-	-	-

	Anniversary	**Group**	**Notes**	**Resultados Esperados**
1	01/01/1990	Família	Teste	Cadastro salvo com sucesso
2	aa/bb/cc	-	-	Mensagem de erro
3	-	-	-	Mensagem de erro
4	01/01/2030	-	-	Mensagem derro
5	-	-	-	Verificar upload de imagem

d) Editar e deletar endereço

	Ação	**Resultado esperado**
1	Clica no botão para editar	Abre formulário de atualização com os dados preenchidos
2	Modificar dados e salvar	Os dados são atualizados e salvos
3	Selecionar um endereço e clicar em deletar	Os dados são apagados

e) Adicionar grupos

	Groupname	**Parentgroup**	**Group header**	**Groupfooter**	**Resultado esperado**
1	Família	Teste	Grupo Família	Grupo Família	Cadastro efetuado
2	“!@#$%	¨-	“!@#$%¨*	“!@#$%	Mensagem de erro
3	-	-	-	-	Mensagem de erro
4	Amigos	Teste	Grupo Amigos	Grupo Amigos	Cadastro efetuado, selecionar e clicar em deletar, o mesmo é apagado
5	Trabalho	Teste	Grupo Trabalho	Grupo Trabalho	Cadastro efetuado, clicar em editar. Abre janela com campos preenchidos, alterá-los e salvar com sucesso a alteração.

f) Próximos Aniversários

	Ação	**Resultado esperado**
1	Cadastrar um aniversário em Janeiro	Cadastro efetuado
2	Cadastrar um aniversário em Março	Cadastro efetuado
3	Cadastrar um aniversário em Dezembro	Cadastro efetuado
4	Clicar nos Próximos Aniversários	Exibido relatório com os aniversários em ordem crescente

g) Impressão dos endereços

	Ação	**Resultado esperado**

1	Clicar na impressão dos endereços	São exibidos todos os endereços com os campos sem possibilidade de edição, para imprimir

h) Impressão de telefones

	Ação	Resultado esperado
1	Clicar na impressão dos telefones	São exibidos todos os telefones com os campos sem possibilidade de edição, para imprimir

i) Logout

Ao clicar em "Logout" o sistema volta para tela de login

6º - Aplicação do teste

Todos os cenários descritos na 5º atividade foram aplicados no sistema, realizando o teste funcional manual do mesmo. Essa atividade teve duração de 41 minutos, classificando-se como média.

7º - Estudo do Resultado

Para analisar o resultado do teste realizado nesta quarta amostra, são informados: o tempo e a quantidade de itens de cada atividade e classificá-las como MUITO SIMPLES, SIMPLES, MÉDIA, COMPLEXA, MUITO COMPLEXA, depois, a cada uma delas é atribuído um peso conforme os resultados obtidos, para definirmos a complexidade do teste.

Atividade	Itens	Tempo	Classificação
1º - Identificação dos métodos que serão testados	9	6m	SIMPLES
2º - Número de Stub ou Mocks	0	3m	MUITO SIMPLES
3º - Identificação dos Cenários	26	12m	SIMPLES
4º - Número de Integrações	1	2m	MUITO SIMPLES
5º - Escrita do script para o cenário	29	36m	MÉDIA
6º - Aplicação do teste	29	41m	MÉDIA

O tempo total gasto para execução do teste funcional manual do sistema de agenda telefônica chamado “Address Book” foi de 100 minutos o equivalente há aproximadamente 2 horas.

5.3 Resultado e definições da técnica

A técnica criada para estimar a elaboração e execução de testes funcionais é composta pelas atividades definidas nos itens anteriores, onde as mesmas devem ser aplicadas ao realizar teste funcional em um sistema. Somente será válida para teste funcional manual e não para testes automatizados, pois o tempo de execução e classificação é divergente.

Quatro sistemas utilizados como amostras ajudaram determinar o tempo, estimativa e classificação de cada atividade, desconsiderando os mocks e stubs, pois nenhum dos sistemas tinham essa implementação.

Uma média em minutos, de cada atividade, foi calculada para cada amostra, utilizando o tempo de execução dividido pela quantidade de itens que foram listados.
Exemplo: na atividade “Identificação dos métodos que serão testados” da primeira amostra, o tempo de execução foi 25 minutos. Esse tempo foi dividido pelos itens listados, que foram 12 métodos, para calcular a média em minutos e resultou em 2 minutos.

Segue abaixo as tabelas das amostras:

Tabela 1 - Dados da primeira amostra.

Atividade	Itens	Tempo	Classificação	Média
1º - Identificação dos métodos que serão testados	12	25m	MÉDIA	2m
2º - Número de Stub ou Mocks	0	6m	SIMPLES	
3º - Identificação dos Cenários	78	57m	COMPLEXA	1m
4º - Número de Integrações	7	11m	MÉDIA	2m
6º - Escrita do script para o cenário	88	166m	MUITO COMPLEXA	2m
7º - Aplicação do teste	88	233m	MUITO COMPLEXA	3m

Tabela 2 - Dados da segunda amostra.

Atividade	Itens	Tempo	Classificação	Média
1º - Identificação dos métodos que serão testados	20	17m	MÉDIA	1m
2º - Número de Stub ou Mocks	0	4m	SIMPLES	
3º - Identificação dos Cenários	108	41m	COMPLEXA	0.5m
4º - Número de Integrações	4	12m	MÉDIA	3m
6º - Escrita do script para o cenário	108	155m	MUITO COMPLEXA	1.5m
7º - Aplicação do teste	108	151m	COMPLEXA	1.5m

Tabela 3 - Dados da terceira amostra.

Atividade	Itens	Tempo	Classificação	Média
1º - Identificação dos métodos que serão testados	11	6m	SIMPLES	0.5m
2º - Número de Stub ou Mocks	0	3m	SIMPLES	
3º - Identificação dos Cenários	40	18m	SIMPLES	0.5m
4º - Número de Integrações	2	4m	SIMPLES	2m
6º - Escrita do script para o cenário	36	29m	SIMPLES	1m
7º - Aplicação do teste	36	25m	SIMPLES	1m

Tabela 4 - Dados da quarta amostra.

Atividade	Itens	Tempo	Classificação	Média
1º - Identificação dos métodos que serão testados	9	6m	SIMPLES	1m
2º - Número de Stub ou Mocks	0	3m	MUITO SIMPLES	
3º - Identificação dos Cenários	26	12m	SIMPLES	0.5m
4º - Número de Integrações	1	2m	MUITO SIMPLES	2m
6º - Escrita do script para o cenário	29	36m	MÉDIA	1m
7º - Aplicação do teste	29	41m	MÉDIA	1.5m

Depois de calcular a média em minutos por amostra, segue a Tabela 5 com os cálculos das médias entre as amostras para cada atividade, definindo assim um peso final por

atividade. Segundo Pressman e Vazques, para cada média é considerado uma margem de erro de aproximadamente 15%.

Exemplo: Para a atividade "Identificação dos métodos que serão testados" foram somadas as médias anteriores (2 + 1 + 0.5 + 1 = 4,5) e divididas pela sua quantidade que é 4 (quatro). O resultado 1,125 é arredondado, finalizando em 1 (um) peso por item.

Tabela 5 - Média de cada atividade.

Atividade	Médias anteriores	Média	Peso por item
1º - Identificação dos métodos que serão testados	2, 1, 0.5, 1	1.12	1
2º - Número de Stub ou Mocks	-	-	-
3º - Identificação dos Cenários	1, 0.5, 0.5, 0.5	0.6	0.5
4º - Número de Integrações	2, 3, 2, 2	2.25	2
6º - Escrita do script para o cenário	2, 1.5, 1, 1	1.37	1.5
7º - Aplicação do teste	3, 1.5, 1, 1.5	1.75	1.5

Obtendo o peso por item, considera-se que cada ponto é o equivalente a um minuto, portanto temos como exemplo: se um cenário vale 0.5 ponto cada, ao termos 10 cenários significa que essa atividade vale 5 pontos.

Com a estimativa em peso, podemos aplicar as atividades da técnica definida em diversos sistemas, e conseguir mensurar o tempo de elaboração e execução de testes funcionais de forma mais precisa.

Ao executar as atividades nas amostras, foi observado que o número de dados de entrada, dados de saída e a comparação dos resultados são muito relevantes no momento de contabilizar o tempo de execução e consequentemente a complexidade do teste, sendo mais longos e complexos os testes que contém muitos dados a serem inseridos. Da mesma forma, amostras com muitos erros/bugs encontrados, demandam mais tempo para realizar o teste do que as amostras que estavam com menos erros, ou seja, o custo dos bugs encontrados na etapa de desenvolvimento é menor do que quando encontrados no etapa de teste.

Para próxima versão deste trabalho, alguns pontos importantíssimos na técnica para a estimativa dos testes devem ser considerados: As ponderações que não foram avaliadas e após

finalizar as definições, aplicar a técnica nos sistemas desenvolvidos para a validação real da mesma.

Capítulo 6. Conclusões

Este trabalho apresentou a definição de uma técnica de estimativa de software capaz de mensurar o tempo de elaboração e execução de testes funcionais, realizados de forma manual, estabelecendo o tempo de desenvolvimento do teste para cada funcionalidade do sistema e identificando o período de teste para: escrita de casos de teste; escrita de cenários de teste, escrita de s*cript* de teste; execução e *log* de teste.

Quatro sistemas já desenvolvidos foram utilizados como amostra para definir a técnica de forma adequada, possibilitando encontrar as variáveis e atividades a serem elaboradas, sendo fundamental para comparar os resultados.

Em relação aos resultados obtidos, foi possível aplicar uma média em minutos nas amostras testadas, chegando a um peso por item da técnica, podendo aplicá-la em outros sistemas e ser hábil para realizar estimativas do teste funcional.

Para continuação este trabalho, na próxima versão, além de algumas ponderações que não foram avaliadas, por exemplo: o número de cenários, número de dados em um cenário, tamanho da massa de dados, número de bugs encontrados, etc., deve ser realizada uma aplicação da técnica em outros sistemas, para comprovar a capacidade da mesma de mensurar o tempo de elaboração e execução de testes funcionais manuais.

Referências Bibliográficas

Aguiar, M. **Pontos de Função ou Pontos por Caso de Uso? Como Estimar Projetos Orientados a Objetos**. disponível em http://www.bfpug.com.br/Artigos/UCP/Aguiar-Pontos_de_Funcao_ou_Pontos_por_Caso_de_Uso.pdf, 2003.

Barbosa, E. F. et. All. **Introdução ao Teste de Software**; Universidade de São Paulo - ICMC/USP, 2002.

Bastos, A.; Rios, E.;Cristalli, R.; Moreira, T. **Base de conhecimento em teste de software.** São Paulo, Martins Fontes, 2007.

Campos, F. **Introdução ao teste de software. Uma abordagem prática**. disponível em http://www.slideshare.net/FabricioFFC/introduo-ao-teste-de-software-uma-abordagem-prtica.

Carvalho, A. D; Brandão, D.G; Tavares, T.A. **Colaboke: Um gerenciador de conteúdo multimídia com conceitos de framework**. 2011. Disponível em http://www.lbd.dcc.ufmg.br/colecoes/epiwpg/2011/s03_carvalho.pdf.

Celepar, **Documento de normatização do uso de UCP no Processo de Desenvolvimento de Software na Companhia de Informática do Paraná** (CELEPAR); 2006.

Celepar. **Guia Técnicas de Teste Metodologia**. Celepar, Ariel Bolzan Witczak. 2009.

Eliza, R; Lagares, V. **Adotando checklists no teste de software**. disponível em http://www.devmedia.com.br/websys.4/webreader.asp?cat=6&revista=javamagazine_87#a-3191.

Delamaro, M.; Maldonado,J.;Ji-No,M. **Introdução ao teste de software**. Rio de Janeiro: Campus, 2007.

Dias, A. C.; Lima, G. M. P. S.; Travassos, G. H. PESC – Programa de Engenharia de Sistemas e Computação. **Teste de Software Projeto Real**. UFRJ, 2006.

Dias, R. **Análise por Pontos de Função: Uma Técnica para Dimensionamento de Sistemas de Informação**. disponível em: www.presidentekennedy.br/resi/edicao03/artigo02.pdf.

Farias, P. P. M. **Framework functest: Aplicado padrões de software na automação de testes funcionais**. Fortaleza, 2007. < http://pt.scribd.com/doc/55531852/23/Particionamento-em-Classes-de-Equivalencia> Acesso em 19 de outubro de 2012.

Fowler, M. **Mocks não são Stubs**. disponível em http://www.infoq.com/br/articles/mocks-Arent-Stubs.

Guarizzo, K. **Métricas de Software**. Jaguariúna, 2008. disponível em http://bibdig.poliseducacional.com.br/document/?view=184.

Hazan, C. **Análise de Pontos por Função** – agosto , 2001 . disponível em http://www.inf.ufes.br/~falbo/download/aulas/es-g/2005-1/APF.pdf.

Heimberg,V.; Grahl .A.E.**Estudo de Caso de Aplicação da Métrica de Pontos de Casos de Uso numa Empresa de Software**. (FURB/DSC). disponível em http://www.inf.furb.br/seminco/2005/artigos/130-vf.pdf.

Hetzel, B. **The Growth of Software Testing**, 1988.

IEEE standard glossary of software engineering terminology.Standard 610.12, IEEE Press, 1990.

Martins, J. C. C. **Gerenciando Projetos de desenvolvimento de software com PMI, RUP e UML** - 3 ed. rev. e ampl. - Rio de Janeiro: Brasport, 2006.

Mastering Requiriments Management With Use Case; IBM; 2005.

Meira, F. L. **Processo Unificado Aberto: Um processo ágil de desenvolvimento de software. .**NET Magazine 75. 2010. A.

Meira, F. L. **Especificação de Requisitos;Análise de Ponto de Caso de Uso.** NET Magazine 75. 2010. B.

Molinari, L. **Teste de Software. Produzindo Sistemas Melhores e Mais Confiáveis**. 1. ed. São Paulo: Érica, 2003.

Murillo, R. **A evolução do teste de software**. disponível em http://imasters.uol.com.br/artigo/9369/des_de_software/a_evolucao_do_teste_de_software/, 2008.

Myers, G. J. **The Art of Software Testing**. John Wiley ans Sons. ISBN 0-471-04328-1, 1979.

Nogueira, E. **Como documentar seus testes.** Disponível em http://sembugs.blogspot.com.br/2007/03/como-documentar-seus-testes.html.

IBM, **Mastering Requiriments Managements With Use Case**. IBM® Software Group; 2006.

IEEE 829-1998 **Standard for Software Test Documentation**. disponível em http://www.ieee.org/index.html.

ISO/IEC 9126-1:2001- **Sotfware engineering – Product quality – Part 1**: **Qualitymodel**

Oliveira, R. B. **Framework Functest: Aplicando Padrões de Software na Automação de Testes Funcionais**; Fortaleza, CE – Brasil; Dezembro / 2007.

Pressman, R.S. Ph.D. **Software Engineering - A Practitioner's Approach**, 2002.

RationalUnifiedProcess; IBM; 2007.

Revista Eletrônica de Sistema de Informação ISSN 1677-3071, 2009.
Vazques, C. E. Et. Al. **Análise de Pontos de Função: Medição, Estimativas e Gerência de Projetos**. 9ª Edição; Editora Érica. 2008.

Witczak, A. B. **Metodologia de Desenvolvimento** - CELEPAR. GuiaTécnicas de Teste; MetodologiaCelepar.Agosto, 2009.

Advanced Software Testing - Vol. 1: Guide to the ISTQB Advanced Certification as an Advanced Test Analyst (Rockynook Computing).