

**UNIJUÍ – UNIVERSIDADE REGIONAL DO NOROESTE DO ESTADO DO  
RIO GRANDE DO SUL**

**DCEEng - Departamento de Ciências Exatas e Engenharias**

**TESTE DE SOFTWARE: UMA NECESSIDADE DAS EMPRESAS**

**CRISTIANE CARLINE LUFT**

**Santa Rosa  
2012**

**UNIJUÍ – UNIVERSIDADE REGIONAL DO NOROESTE DO ESTADO DO  
RIO GRANDE DO SUL**

**DCEEng - Departamento de Ciências Exatas e Engenharias**

**TESTE DE SOFTWARE: UMA NECESSIDADE DAS EMPRESAS**

**CRISTIANE CARLINE LUFT**

Trabalho de Conclusão de Curso apresentado ao curso de Ciência da Computação, do Departamento de Ciências Exatas e Engenharias, da Universidade Regional do Noroeste do Estado do Rio Grande do Sul, como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Msc. Marcos Ronaldo Melo Cavalleiro

**Santa Rosa  
2012**

# **TESTE DE SOFTWARE: UMA NECESSIDADE DAS EMPRESAS**

**CRISTIANE CARLINE LUFT**

Trabalho de Conclusão de Curso apresentado ao curso de Ciência da Computação, do Departamento de Ciências Exatas e Engenharias, da Universidade Regional do Noroeste do Estado do Rio Grande do Sul, como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação.

---

Orientador: Prof. Msc. Marcos Ronaldo Melo Cavalheiro

**BANCA EXAMINADORA**

---

Prof. Msc. Romário Lopes Alcântara

**Santa Rosa  
2012**

Dedico este trabalho

aos meus pais e ao meu namorado

pelo apoio dado durante a elaboração deste trabalho!

## **AGRADECIMENTOS**

Primeiramente quero agradecer a Deus, por ter me dado à vida, por ter me dado forças e iluminado meu caminho para que pudesse concluir mais uma etapa da minha vida.

Aos meus pais, Milton e Noeli, que me proporcionaram condições de estudar e sempre me apoiaram e incentivaram neste momento tão importante da minha vida.

Ao meu namorado Jhonathan, que sempre esteve ao meu lado me dando força, carinho, amor e apoio. Te Amo!

Aos meus amigos por compreenderem nos momentos que não pude estar presente.

Aos meus colegas que tive o prazer de conhecer e compartilhar grandes momentos de alegria e aprendizado. Em especial aos que se tornaram verdadeiros amigos nessa caminhada.

Ao meu orientador Marcos Cavalheiro pelo auxílio e apoio dados neste trabalho.

Aos meus professores pelos ensinamentos transmitidos durante o curso.

A UNIJUI pela oportunidade concedida para a realização deste curso.

Muito Obrigada!

“Nenhuma técnica é boa em todos os lugares,  
mas uma técnica pode ser boa em pelo menos um lugar,  
a questão é saber quando ela irá funcionar ou não.”

(Autor Desconhecido)

## **RESUMO**

O teste de software é muito importante para a garantia da qualidade de um software. No entanto, a atividade de teste não é uma tarefa simples, ela exige um bom planejamento durante a execução para ser bem sucedida. E a falta de conhecimento dos programadores ou analistas sobre o processo de teste de software pode fazer com que muitos erros não sejam encontrados, resultando em um software sem qualidade, e gerando assim a insatisfação dos clientes. Desta forma, faz-se necessário o melhor conhecimento das atividades de teste de software, como as técnicas de teste e seus critérios, os níveis (fases) de teste e os vários tipos de teste de software para que sejam produzidos softwares seguros, confiáveis e de maior qualidade.

**Palavras chave:** Teste de Software, Qualidade de Software.

## **ABSTRACT**

Software testing is very important for ensuring the quality of software. However, the test activity is not a simple task, it requires good planning during execution to be successful. And the lack of knowledge of the programmers or analysts about the process of software testing can cause many errors are not found, resulting in a quality free software, and generating customer dissatisfaction. Thus, it is necessary to the best knowledge of the activities of software testing, such as testing techniques and criteria levels (stages) test and the various types of software testing software to be produced safe, reliable and higher quality.

**Keywords:** Software Testing, Software Quality.



## **LISTA DE SIGLAS**

ISO	- International Organization for Standardization
IEC	- International Electrotechnical Commission
IEEE	- Institute of Electrical and Electronics Engineers
XP	- Extreme Programming
TI	- Tecnologia da Informação
SEPRORGS	- Sindicato das Empresas de Informática do Rio Grande do Sul

## LISTA DE FIGURAS

Figura 1: Testes de softwares mais usados .....	60
Figura 2: Gráfico de como os testes são realizados.....	61
Figura 3: Documentos mais utilizados pelas empresas .....	61
Figura 4: Diminuição dos custos após a implantação da atividade de teste .....	62
Figura 5: Diminuição dos erros após a implantação da atividade de teste.....	62
Figura 6: Testes de softwares mais utilizados.....	62
Figura 7: Gráfico de como são realizado os testes .....	63
Figura 8: Documentos mais utilizados.....	63
Figura 9: Diminuição dos erros após a implantação da atividade de teste.....	64

## **LISTA DE TABELAS**

Tabela 1: Testes de Softwares realizados pelas empresas .....	65
Tabela 2: Documentação utilizada pelas empresas .....	66

## SUMÁRIO

<b>INTRODUÇÃO .....</b>	<b>13</b>
<b>1 ASPECTOS GERAIS DA QUALIDADE E TESTE DE SOFTWARE .....</b>	<b>15</b>
1.1 IMPORTÂNCIA DO TESTE DE SOFTWARE .....	17
1.2 ATIVIDADE DE TESTES.....	18
1.3 DOCUMENTAÇÃO DA ATIVIDADE DE TESTES.....	19
1.4 TÉCNICAS DE TESTE DE SOFTWARE.....	22
1.4.1 <i>Teste Funcional (caixa preta)</i> .....	22
1.4.2 <i>Teste Estrutural (caixa branca)</i> .....	25
1.5 FASES DE TESTE DE SOFTWARE .....	26
1.5.1 <i>Teste de Unidade</i> .....	27
1.5.2 <i>Teste de Integração</i> .....	28
1.5.3 <i>Teste de Sistema</i> .....	29
1.5.4 <i>Teste de Aceitação</i> .....	30
1.5.5 <i>Teste de Regressão</i> .....	31
<b>2 OUTROS TIPOS DE TESTE DE SOFTWARE .....</b>	<b>32</b>
2.1 TESTE DE CARGA .....	32
2.2 TESTE DE CONFIGURAÇÃO.....	32
2.3 TESTE DE DESEMPENHO/PERFORMANCE.....	33
2.4 TESTE DE INSTALAÇÃO .....	33
2.5 TESTE DE RECUPERAÇÃO .....	34
2.6 TESTE DE SEGURANÇA .....	34
2.7 TESTE DE USABILIDADE .....	35
2.8 TESTE DE VOLUME.....	36
2.9 TESTE PARALELO .....	36

2.10 TESTES AUTOMATIZADOS.....	36
2.11 METODOLOGIAS ÁGEIS .....	38
2.11.1 <i>Extreme Programming</i> .....	39
2.11.2 <i>SCRUM</i> .....	41
2.11.3 <i>Teste de Software nas Metodologias Ágeis</i> .....	42
<b>3 ESTUDO DE CASO</b> .....	44
3.1 ANÁLISE DAS RESPOSTAS DAS EMPRESAS QUE DESENVOLVEM SOFTWARES.....	59
3.2 ANÁLISE DAS RESPOSTAS DAS EMPRESAS QUE PRESTAM SERVIÇO DE TESTES DE SOFTWARE PARA AS EMPRESAS.....	62
3.3 ANÁLISE DE TODAS AS EMPRESAS .....	64
<b>CONCLUSÃO</b> .....	67
<b>REFERÊNCIAS</b> .....	69
<b>APÊNDICE A – Questionário</b> .....	74
<b>APÊNDICE B – Respostas Do Questionário</b> .....	82

## INTRODUÇÃO

Atualmente existe uma demanda muito grande por software de alta qualidade e sem erros. Com isso, empresas estão procurando aprimorar e aperfeiçoar o processo de desenvolvimento de softwares através de várias atividades de garantia de qualidade, fazendo com que o usuário tenha uma maior confiança e segurança na utilização de um sistema.

A qualidade é um requisito que deve estar associado a todos os tipos de serviços prestados, mais ainda quando se trata do desenvolvimento de um software, pois um único sistema pode estar integrado a todos os departamentos de uma empresa.

Garantir que todos os programas funcionem corretamente não é uma tarefa fácil, ainda mais quando o tamanho e a complexidade do projeto a ser desenvolvido são muito grandes.

Softwares com defeitos e sem qualidade resultam na insatisfação do cliente, imensos gastos com manutenção e, além disso, pode comprometer a imagem da empresa.

O teste é uma das atividades de garantia de qualidade mais utilizada e de grande importância no desenvolvimento de um software. Através dele é possível encontrar falhas nos sistemas, antes que o mesmo seja entregue ao cliente.

O teste de software é o processo de executar o sistema e observar seu comportamento em relação aos requisitos acordados em contrato.

O teste pode ser feito de forma manual ou automatizada, através de diferentes tipos, técnicas e níveis de teste de software, onde estes fornecem o suporte necessário para garantir casos de testes adequados para cada fase do desenvolvimento de um software, pois cada etapa pode gerar diferentes tipos de falhas. Por isso o teste, além de estar presente em todas as fases do desenvolvimento, precisa ser adequado a cada uma destas etapas, a fim de estabelecer um nível de qualidade na detecção de erros. Desta forma fica clara a

necessidade de se conhecer melhor as atividades de teste de software para que sejam produzidos softwares seguros, confiáveis e de maior qualidade.

O objetivo deste trabalho de conclusão de curso é verificar os métodos de testes existentes nas empresas de software na área de tecnologia da informação, através da elaboração de um questionário que será enviado para empresas de desenvolvimento.

Este trabalho está estruturado em três capítulos. O capítulo 1 apresenta o conceito de qualidade de software e mostra quais os atributos necessários para se ter um software de qualidade. Apresenta o conceito, o objetivo e a importância dos testes de software. Também apresenta aspectos referentes à atividade de teste e a documentação e descreve as principais técnicas e níveis de teste de software. O capítulo 2 descreve os tipos de teste de software e apresenta o conceito de teste automatizado e metodologias ágeis. O capítulo 3 descreve os resultados obtidos com a pesquisa realizada nas empresas de desenvolvimento de software. Por fim, são apresentadas as conclusões, as dificuldades obtidas e as sugestões para trabalhos futuros.

## 1. ASPECTOS GERAIS DA QUALIDADE E TESTE DE SOFTWARE

Empresas de desenvolvimento de software estão cada vez mais preocupadas em garantir a qualidade de seus produtos, para que os mesmos sejam competitivos e conquistem a satisfação e confiança dos clientes.

As organizações estão buscando eficiência para conseguir sobreviver em um ambiente cada vez mais hostil – o de um mercado cada vez mais competitivo. As empresas estão buscando a tecnologia para reduzir custos e ampliar sua forma de atuação. Estão sofisticando seus sistemas para tomar decisões cada vez mais complexas, com a intenção de ganhar eficiência e controle (BARTIÉ, 2002).

Para Sommerville (2003), atingir um alto nível de qualidade de produto ou serviço deve ser atualmente o maior objetivo das organizações, pois o mercado não admite mais que se entregue produtos de baixa qualidade.

Bartié (2002) define qualidade de software como “um processo sistemático que focaliza todas as etapas e artefatos produzidos com o objetivo de garantir a conformidade de processos e produtos, prevenindo e eliminando defeitos”.

Para que a qualidade seja garantida, é necessário definir e estabelecer uma estrutura de procedimentos e de padrões organizacionais que devem ser utilizados durante o desenvolvimento do software que conduzem ao produto final de alta qualidade.

Existe uma ampla gama de atributos de qualidade de software em potencial que devem ser considerados durante o processo de planejamento de qualidade. Em geral, não é possível para qualquer sistema ser otimizado em relação a todos esses atributos, e, assim, uma parte importante do planejamento de qualidade é selecionar os principais atributos de qualidade e planejar como eles podem ser obtidos (SOMMERVILLE, 2003).

A norma ISO/IEC 9126 identifica seis atributos-chave que um software de qualidade deve possuir (PRESSMAN, 2006 apud PINTO JÚNIOR, 2009):

- **Funcionalidade** – o software deve funcionar adequadamente e atender as necessidades especificadas;



- **Confiabilidade** – o software deve funcionar por determinado período de tempo livre de erros e falhas;
- **Usabilidade** – o software deve ser compreendido e utilizado facilmente pelo usuário;
- **Eficiência** – o software deve atingir seu melhor desempenho utilizando seus recursos da melhor forma;
- **Manutenibilidade** – o software deve realizar modificações e reparos sem grandes dificuldades;
- **Portabilidade** – o software deve ser transferido facilmente de um ambiente (plataforma) para outro.

Outros atributos que contribuem para se ter um software de qualidade são “profissionais experientes e bem treinados, metodologias e ferramentas adequadas, participação constante dos usuários finais, bom entendimento do problema e modelagem da solução flexível ao longo prazo” (BARTIÉ, 2002).

A qualidade deve ser garantida em todas as etapas do processo de desenvolvimento do software, para que não seja iniciada uma nova fase sem que a anterior tenha sido concluída adequadamente.

Segundo Bartié (2002), os erros ocorrem em todo o processo de desenvolvimento de um sistema, porém, a maior incidência dos erros está concentrada nas fases iniciais, devido à má especificação e entendimento dos objetivos a serem alcançados.

Com o surgimento de novas tecnologias, os softwares estão se tornando cada vez mais complexos e, em função disso, o mercado de teste de software também está evoluindo de forma a buscar a qualidade dos sistemas.

O teste é uma das fases do processo de desenvolvimento de um software que mais contribui para a garantia da qualidade do software.

O principal objetivo do teste é encontrar erros no software antes que o mesmo seja entregue ao cliente. Conforme Myers (1979, apud BARTIÉ, 2002),

[...] se o objetivo do teste fosse apenas provar a boa funcionalidade de um aplicativo, seriam encontrados poucos defeitos, uma vez que toda a energia do processo de testes seria direcionada apenas na comprovação desse fato. Porém, se o objetivo for identificar erros, um maior número de problemas será encontrado, uma vez que os profissionais de qualidade buscarão vários cenários para avaliar o comportamento do software.

Para Inthurn (2001, apud ZIMMERMANN, 2006), o “teste de software tem como objetivo aprimorar a produtividade e fornecer evidências de confiabilidade e da qualidade do software”.

Através dos testes pode se observar se o sistema em desenvolvimento está sendo feito de maneira correta e conforme os requisitos especificados pelo usuário.

De acordo com Tomelin (2001), os testes avaliam a qualidade do software em todas as etapas do processo de desenvolvimento, desde a análise de requisitos até a fase de manutenção do software.

## 1.1 IMPORTÂNCIA DO TESTE DE SOFTWARE

Os softwares estão tornando-se cada vez mais parte do nosso dia-a-dia, desde produtos comerciais até produtos de consumo. O software se tornou igual a qualquer outro produto comercializado.

Com a exigência cada vez maior do mercado consumidor, é imprescindível que se garanta a qualidade nos produtos de software comercializados no mercado. Diante dessa necessidade muitas empresas de desenvolvimento têm aderido ao teste de software.

O teste de software cumpre um papel fundamental, pois pode contribuir para melhorar a qualidade, diminuir os custos, evitar o retrabalho, aumentar a segurança e a confiabilidade, antecipar a descoberta de falhas e incompatibilidades.

## 1.2 ATIVIDADE DE TESTES

Conforme Myers (1979, apud JORGE, 2009),

[...] a atividade de teste é o processo de executar um programa com a intenção de encontrar um erro. Um bom caso de teste é aquele que tem alta probabilidade de revelar erros e um teste bem sucedido é aquele que detecta a existência de um erro ainda não descoberto.

Segundo Pressman (1995), “a atividade de teste não pode mostrar ausência de bugs, ela só pode mostrar se defeitos estão presentes”. Caso o processo de teste não detecta erros, não quer dizer que o programa está correto e sim que o teste pode não ter sido suficientemente completo para revelar os erros existentes (KOLM, 2001).

A atividade de teste deve ser iniciada nas primeiras etapas do processo de desenvolvimento do software e serem realizados simultaneamente, até a entrega do software ao usuário, para que os resultados de qualidade sejam atingidos (RIOS, 2003).

“Os objetos testados são todos aqueles que aparecem durante o processo de desenvolvimento do software. Isto inclui módulos de código, especificações de requisitos e projeto, estruturas de dados” (NEVES, 1999).

[...] o teste não é uma tarefa trivial. A dificuldade em testar software é caracterizada por alguns pontos importantes como: o teste de software é um processo caro; existe uma falta de conhecimento sobre a relação custo/benefício do teste; há falta de profissionais especializados na área de teste; existem dificuldades em implantar um processo de teste; há o desconhecimento de um procedimento de teste adequado; há o desconhecimento de técnicas de teste adequadas; há o desconhecimento sobre como planejar a atividade de teste; e finalmente, a preocupação com a atividade de teste somente na fase final do projeto (CRESPO et al, 2004).

Conforme Paula Filho (2005), os fatores que dificultam a atividade de teste são as restrições de cronograma e orçamento. Portanto, os testes precisam ser “bem planejados e desenhados, para conseguir-se o melhor proveito possível dos recursos alocados para eles” (PAULA FILHO, 2005).

“Durante e após a realização dos testes, os resultados de cada teste devem ser minuciosamente inspecionados, comparando-se resultados previstos e obtidos” (PAULA FILHO, 2005).

No planejamento dos testes, deve ser identificado “o menor conjunto de cenários que cubra o maior número de situações possíveis. É fundamental identificar os casos de testes que tenham a maior probabilidade de revelar defeitos e eliminar os casos redundantes” (BARTIÉ, 2002).

Iniciar os testes nas fases iniciais do processo de desenvolvimento faz com que os custos do projeto fiquem mais acessíveis, pois, segundo Rios (2003), “os defeitos detectados quando o sistema ainda está nos seus estágios iniciais de desenvolvimento, podem ser mais baratos do que aqueles corrigidos quando o sistema já está em produção”.

“Pela Regra 10 de Myers, o custo de correção dos defeitos cresce exponencialmente à medida que o processo de desenvolvimento avança dentro do ciclo de vida do sistema” (RIOS, 2003).

### 1.3 DOCUMENTAÇÃO DA ATIVIDADE DE TESTES

A documentação é uma parte importante da atividade de testes. Através dela é possível ter um controle do que está sendo testado, o que já foi testado e o que ainda falta testar. Também é uma forma de comunicação entre as equipes de testes e fornece ao cliente evidências da qualidade do software (CORRÊA, 2007).

Robson Corrêa (2007) cita também outras vantagens da documentação, são elas: facilidade na manutenção do software, redução da duplicação de esforços depois de ter uma base de documentação já criada, facilidade no acompanhamento e na gerência do teste e melhora na qualidade das atividades de teste.

Com base na importância da documentação, o IEEE (Institute of Electrical and Electronics Engineers) criou a norma IEEE 829 para padronizar os documentos de teste de software.

Segundo Rios (2007), “nenhum processo de teste pode funcionar adequadamente se não possuir documentos padronizados para serem usados pelos profissionais envolvidos”.

Por exemplo, se cada líder de projeto de teste de software usasse um tipo de Plano de Teste, os planos não estariam necessariamente errados, mas elaborados de forma diferente. Com o decorrer do tempo existiriam centenas de planos de teste escritos de forma diferente dentro de uma mesma organização. Quem fosse, no futuro, usar um desses planos, com toda certeza teria muita dificuldade de buscar as informações que precisasse, pela falta de padrão, pois os dados estariam em cada plano colocados em locais diferentes e usando terminologias diversas para definições semelhantes (RIOS, 2007).

Essa norma propõe o uso de oito documentos para que o processo de teste funcione corretamente, sendo que um documento é para o planejamento, três são para a especificação e quatro são para os relatórios. São eles:

- **Plano de Teste:** é um documento usado para gerenciar o projeto de teste. O Plano de Teste deve possuir: identificador do plano de teste, introdução, itens de teste, funcionalidades a serem testadas, funcionalidades a não serem testadas, abordagem do teste, critérios de liberação/falha dos itens, requisitos de suspensão e retomada, entregas do teste, tarefas do teste, necessidades de ambientes, responsabilidades, necessidades de equipe e de treinamento, cronograma, riscos e aprovações (RIOS, 2007).
- **Especificação de Projeto de Teste:** é usado para refinar o plano de teste, identificando os requisitos e as funcionalidades que serão testadas e quais os tipos de testes devem ser feitos. O Projeto de Teste é composto por: identificador, funcionalidades a serem testadas, abordagem refinada, casos de teste com a sua respectiva identificação, critérios de passagem e falha por funcionalidade (RIOS, 2007).
- **Especificação de Caso de Teste:** define uma unidade de teste que será executada pelo testador. Este documento deve possuir: identificador, itens

de teste, especificações de entrada e de saída, necessidades de ambiente, requisitos ou procedimentos especiais e dependências entre casos de teste (RIOS, 2007).

- **Especificação de Procedimento de Teste:** especifica os passos necessários para executar um conjunto de teste. O Procedimento de Teste deve conter: identificador, propósito, requisitos especiais e passos do procedimento (RIOS, 2007).
- **Diário de Teste:** apresenta um registro cronológico das ocorrências de todo o processo de execução dos testes. O Diário de Teste deve ter os seguintes campos: identificador, descrição, entradas das atividades e eventos (RIOS, 2007).
- **Relatório de Incidente de Teste:** documenta qualquer evento que ocorra durante a execução dos testes que requer uma análise posterior. O Relatório de Incidente de Teste é composto por: identificador, sumário, descrição do incidente e impacto (RIOS, 2007).
- **Relatório-Resumo de Teste:** apresenta um resumo dos resultados alcançados no projeto de teste. O Relatório-Resumo de Teste deve possuir: identificador, sumário, variações, avaliação funcional, sumário dos resultados, avaliação dos testes, sumário das atividades e aprovações (RIOS, 2007).
- **Relatório de Encaminhamento de Itens de Teste:** identifica os itens encaminhados para o teste no caso de equipes distintas serem responsáveis pelas tarefas de desenvolvimento e de teste. Ele deve ter os seguintes campos: identificador, itens passados, localização, estado e aprovações (RIOS, 2007).

Conforme o IEEE, esses documentos podem ser aplicados na execução do teste de qualquer tipo de software. Também podem ser usados em qualquer metodologia, projeto e processo de teste (RIOS, 2007).

## 1.4 TÉCNICAS DE TESTE DE SOFTWARE

As técnicas de teste de software são: teste funcional e teste estrutural. Cada uma dessas técnicas de teste mencionadas possui diversos critérios de teste.

Se uma das duas técnicas for aplicada, não que dizer que a outra não precisa ser feita, pois,

[...] em algumas situações o teste funcional não é capaz de detectar alguns defeitos, pois preocupa somente em comparar se as saídas produzidas pelo sistema estão de acordo com o esperado, não levando em consideração a funcionalidade interna do software. Desta forma, podem existir procedimentos internos que não afetam as respectivas saídas e que não serão detectados se for aplicado apenas o teste funcional (FURLAN, 2009).

### 1.4.1 Teste Funcional (caixa preta)

O teste funcional também é conhecido como teste de caixa preta, “pelo fato de tratar o software como uma caixa cujo conteúdo é desconhecido e só é possível visualizar o lado externo” (MEIRELLES, 2008). Ele utiliza métodos “para garantir que os requisitos do sistema sejam plenamente atendidos pelo software que foi construído” (BARTIÉ, 2002).

O teste funcional concentra-se nos requisitos funcionais do software (PRESSMAN, 1995). Através dele torna-se possível verificar as entradas e saídas de cada unidade. Segundo Pressman (1995), o teste de caixa preta descobre “funções incorretas ou ausentes, erros de interface, erros nas estruturas de dados ou no acesso a banco de dados externos, erros de desempenho, e erros de iniciação e término”.

Este tipo de teste não se preocupa em “verificar como ocorrem internamente os processamentos no software, mas se o algoritmo inserido no software produz os resultados esperados” (BARTIÉ, 2002).

Segundo Sommerville (2007, apud JORGE, 2009), o teste funcional envolve dois passos principais: identificar as funções que o software deve realizar e criar casos de teste capazes de checar se essas funções estão sendo realizadas pelo software. As funções que o software deve possuir são identificadas a partir de sua especificação. Assim, uma especificação bem elaborada e de acordo com os requisitos do usuário é essencial para esse tipo de teste.

O profissional responsável por realizar esse tipo de teste deve ter “conhecimento dos requisitos, suas características e comportamentos esperados, para que seja possível avaliar o software através dos resultados gerados pela aplicação” (BARTIÉ, 2002).

“O uso dessa técnica reduz o número de casos de teste necessários para se executar um teste confiável e produz casos de teste que indicam a presença ou ausência de classes de erros” (MYERS, 1979, apud VILLAS BOAS, 2003).

Inthurn (2001, apud ZIMMERMANN, 2006) apresenta as seguintes vantagens do teste funcional: alta capacidade de detectar erros tanto de aplicação quanto do processo de negócio; os resultados são avaliados visualmente na aplicação; testes mais simples de serem implementados; e para a realização de testes não requer conhecimento da tecnologia empregada. A mesma autora também apresenta as seguintes desvantagens dessa técnica: convencimento das áreas de homologação já existentes a seguir um procedimento definido de teste; e requer maior manutenção dos artefatos de teste gerados.

Os principais critérios associados a essa técnica são: particionamento de equivalência, análise de valor limite e grafo de causa-efeito.

#### *a. Particionamento de Equivalência*

Esse critério divide o domínio de entrada de dados em classes ou partições de equivalência. Cada classe representa um possível erro a ser identificado (BARTIÉ, 2002).



Segundo o mesmo autor cada entrada deve ser analisada com o objetivo de identificar um conjunto de valores válidos e inválidos. Desse conjunto são extraídas as classes e obtido os casos de testes.

Por exemplo, em um sistema de gestão de contratos, a idade válida dos clientes varia entre 18 e 120 anos. As classes identificadas são os valores abaixo de 18, valores entre 18 e 120, valores acima de 120. Para cada uma dessas classes, existem um conjunto de valores que têm, potencialmente, a mesma capacidade de detectar erros, sendo indispensável a execução de vários testes para valores pertencentes à mesma classe de equivalência (BARTIÉ, 2012).

Ou seja, se um caso de teste de determinada classe revela um erro, todos os demais elementos daquela mesma classe também revelam o mesmo erro.

#### *b. Análise de Valor Limite*

Esse critério é complementar ao particionamento de equivalência. De acordo com Bartié (2002), a análise de valor limite difere do particionamento de equivalência em dois aspectos:

- Ao invés de selecionar qualquer elemento em uma classe de equivalência, devem ser selecionados os casos de teste das extremidades de cada classe, ou seja, os valores-limite. Pois, segundo o autor, o software está mais susceptível ao erro nas fronteiras dos domínios de dados do que propriamente nas regiões centrais;
- Além de considerar as condições de entrada, os casos de teste devem considerar o conjunto de saída.

#### *c. Grafo de Causa-Efeito*

Segundo Pressman (1995), esse critério “oferece uma representação concisa das condições lógicas e das ações correspondentes”.

Primeiramente, são levantadas as possíveis condições de entrada (causas) e as possíveis ações (efeitos) do programa. A seguir é construído um grafo relacionando as causas e efeitos levantados. Esse grafo é convertido em uma tabela de decisão e então as regras desta tabela são usadas para gerar os casos de teste (PRESSMAN, 1995).

#### *1.4.2 Teste Estrutural (caixa branca)*

O teste estrutural, também conhecido como teste de caixa branca, é baseado na arquitetura interna do software. Ele emprega métodos para identificar erros nas estruturas internas dos programas através da simulação de situações que exercitem adequadamente todas as estruturas usadas na codificação (BARTIÉ, 2002).

O teste de caixa branca “baseia-se num minucioso exame dos detalhes procedimentais. Os caminhos lógicos através do software são testados, fornecendo-se casos de teste que põem à prova conjuntos específicos de condições e laços” (SILVA, 2005).

Essa técnica de teste visa detectar “comandos incorretos, estruturas incorretas, variáveis não definidas e erros de inicialização e finalização de loops” (FURLAN, 2009).

Usando métodos de teste de caixa branca, podem-se derivar os casos de teste que garantam que todos os caminhos independentes dentro de um módulo tenham sido exercitados pelo menos uma vez; exercitem todas as decisões lógicas para valores falsos ou verdadeiros; executem todos os laços em suas fronteiras e dentro de seus limites operacionais; e exercitem as estruturas de dados internas para garantir a sua validade (PRESSMAN, 1995).

O profissional responsável por realizar esse tipo de teste deve conhecer a tecnologia empregada pelo software e possuir conhecimento sobre a arquitetura interna da solução, para isso, ele deverá ter acesso a fontes e estrutura de banco de dados (BARTIÉ, 2002).

Bartié (2002) apresenta as seguintes vantagens dos testes estruturais: alta eficiência na detecção de erros e podem ser modelados e estruturados pelo próprio desenvolvedor. O mesmo autor também apresenta as seguintes desvantagens dessa técnica: são difíceis de implementar e sem a contratação de uma equipe de testes, os resultados podem ser demorados e pouco significativos.

Dentre os critérios estruturais podem-se destacar: critérios baseados em fluxo de controle, critérios baseados em fluxo de dados e critérios baseados na complexidade.

Os critérios baseados em fluxo de controle utilizam somente características de controle da execução do programa, tais como comandos ou desvios, para derivar os requisitos de teste. Os critérios baseados em fluxo de dados utilizam informações do fluxo de dados do programa para derivar os requisitos de teste. E os critérios baseados na complexidade derivam os requisitos de teste a partir da complexidade do programa (MALDONADO et al, 1998).

## 1.5 FASES DE TESTE DE SOFTWARE

Além da utilização de técnicas e critérios para aumentar o nível de confiabilidade do software testado, quando o tamanho e a complexidade do projeto de teste aumentam, é necessário dividir a atividade de teste em várias fases. Com esta divisão, os testes podem focar em diferentes aspectos do software testado, assim como diferentes tipos de erros.

As fases de teste de software são: teste de unidade, teste de integração, teste de sistema e teste de aceitação. Além dessas cinco, tem o teste de regressão que é uma fase especial de teste de software.

A seguir são apresentadas as descrições de cada uma das fases de maneira detalhada.

### 1.5.1 Teste de Unidade

O teste de unidade é a fase do processo de teste em que se testam os menores componentes do código. Segundo Pressman (1995), “o teste de unidade concentra-se no esforço de verificação da menor unidade de projeto de software - o módulo”.

Os testes de unidades têm por objetivo verificar cada unidade que compõe o software, isoladamente, para determinar se cada uma delas realiza o que foi especificado.

Segundo Bartié (2002), nessa fase de teste, é exercitado toda a estrutura interna de um componente do software, como os desvios condicionais, os laços de processamento e todos os possíveis caminhos alternativos de execução.

Para Inthurn (2001), o teste de unidade preocupa-se em exercitar a menor unidade de projeto de software com o intuito de verificar se as informações que entram e saem são consistentes; as condições de limites garantem a execução correta da unidade; todos os caminhos básicos estão corretos; e os caminhos de tratamento de erros sejam devidamente tratados.

“Durante esta fase utiliza-se muito a técnica de teste estrutural que requer a execução de elementos específicos de estrutura de controle de cada unidade, com o objetivo de garantir uma completa cobertura e máxima detecção de erros” (MERLIN, 2006).

O teste de unidade é o tipo mais importante de teste para a grande maioria das situações, já que é ele que deve testar se um algoritmo faz o que deveria ser feito e garantir que o código encapsulado por uma unidade deve produzir o efeito colateral esperado. Outro aspecto importante é que o teste de unidade é focalizado em um trecho específico do código, desta forma os erros encontrados são facilmente localizados, diminuindo o tempo gasto com depuração (BERNARDO E KON, 2008).

### 1.5.2 Teste de Integração

Após testar separadamente cada módulo, inicia-se a fase de integração, onde estes módulos são agrupados para compor os subsistemas, conforme a arquitetura do sistema definida no projeto preliminar. O objetivo desta fase é encontrar falhas de integração entre as unidades (FURLAN, 2009).

De acordo com Pressman (1995), o teste de integração é a fase para a construção da estrutura do programa e para descobrir erros associados a interfaces.

O teste de integração pode ter sua execução iniciada assim que alguns componentes ficarem prontos, ou seja, quando um componente estiver funcionando e tiver atingido seus objetivos (PFLEEGER, 2004 apud FURLAN, 2009).

[...] os tipos de erros geralmente revelados com o teste de integração elevam em muito o custo da atividade de teste se forem detectados somente nos estágios mais avançados, principalmente se a correção do erro forçar modificações em unidades previamente testadas. Desse modo, a realização dos testes de integração é de fundamental importância para assegurar uma melhor qualidade do software sendo construído (ZANOTTO, 2002).

A integração pode ser incremental ou não-incremental. Na integração não-incremental as unidades são testadas individualmente e depois integradas de uma só vez. Isso torna o teste menos eficaz, pois torna-se difícil isolar um erro e, quando esses são corrigidos, novos erros são gerados na estrutura do programa. Já na integração incremental, o programa construído é testado em pequenos segmentos, tornando o teste mais eficiente, pois os erros são mais fáceis de serem isolados e corrigidos e as interfaces tem maior probabilidade de serem testadas corretamente (PRESSMAN, 1995).

O teste de integração é seguido por dois métodos de interação: top-down e bottom-up.

Na integração top-down,

[...] as unidades de maior nível hierárquico são criadas inicialmente, sofrendo um processo de refinamento e decomposição sucessivos, até que seja alcançado o menor nível estrutural do projeto e todas as unidades tenham sido criadas. À medida que as unidades são construídas e alteradas, os testes de integração avaliam se as interfaces com outras unidades continuam compatíveis (BARTIÉ, 2002).

Na integração bottom-up,

[...] os componentes de níveis mais básicos da arquitetura do aplicativo são conjuntamente testados por controladores construídos para simular as interações dinâmicas e suas interfaces. À medida que se desenvolve o processo de integração, componentes reais substituem os controladores anteriores e novos controladores de testes são criados para simular as interfaces de um maior nível da arquitetura. Esses níveis de integração seguem até que não existam mais níveis a serem alcançados, atingindo o nível máximo de integração que os componentes podem chegar (BARTIÉ, 2002).

A escolha de um método de integração depende das características do software e do cronograma do projeto. O ideal seria usar o método top-down para os níveis superiores da estrutura do programa acoplada com o método bottom-up para os níveis subordinados (PRESSMAN, 1995).

### *1.5.3 Teste de Sistema*

Após a integração do software, aplica-se o teste de sistema, sendo que é esta a etapa em que o software é testado por completo.

O teste de sistema avalia o software em busca de falhas por meio da utilização do mesmo, como se fosse um usuário final. Dessa forma, os testes são executados nos mesmos ambientes, com as mesmas condições e com os mesmos dados de entrada que um usuário utilizaria no seu dia-a-dia [...] (DIAS NETO, 2006).

Esse teste “tem como objetivo determinar se o software e demais elementos que compõem o sistema, tais como hardware e banco de dados, combinam-se adequadamente e se atendem aos requisitos especificados” (FURLAN, 2009).

“Planejar os testes de sistema não é uma tarefa fácil, pois exige o perfeito entendimento dos requisitos não funcionais de um software, que muitas vezes não são claramente descritos ou representam cenário complexos” (BARTIÉ, 2002).

#### *1.5.4 Teste de Aceitação*

O teste de aceitação é a etapa em que o teste é conduzido por usuários finais do sistema. Essa fase serve para o cliente julgar se o produto é, ou não, aceitável de acordo com os critérios previamente acordados, e da maneira especificada em contrato.

Nesta fase o software é disponibilizado ao cliente para que este interaja com o sistema verificando se as funcionalidades requisitadas no início do projeto estão corretas (BARTIÉ, 2002).

Segundo o mesmo autor, se essa fase apresentar muitas falhas, é sinal de que os processos de detecção de erros anteriormente executados não estão sendo implementados corretamente (BARTIÉ, 2002).

O teste de aceitação pode ser:

- **Teste Alfa:** onde o teste de aceitação é executado pelo cliente nas instalações do desenvolvedor. O software é utilizado num ambiente real com a presença do desenvolvedor que registra erros e problemas de uso (PRESSMAN, 1995);
- **Teste Beta:** onde o teste de aceitação é realizado pelo usuário final no ambiente de produção, sem a presença ou controle do desenvolvedor. O usuário registra os problemas encontrados e relata-os ao desenvolvedor (PRESSMAN, 1995).

### *1.5.5 Teste de Regressão*

O teste de regressão é uma categoria especial de teste, que visa assegurar que alterações feitas em parte do produto não afetem as partes já testadas. A cada nova versão do software, são aplicados todos os testes que foram feitos nas versões anteriores do sistema.

Conforme Paula Filho (2000), as alterações feitas durante a correção dos erros detectados podem gerar problemas em segmentos do código previamente testados. Os testes de regressão verificam novamente esses segmentos, para checar se eles continuam funcionando corretamente.

Durante a realização desse tipo de teste devem ser verificados se a documentação do sistema está consistente com o comportamento atual e se os dados e as condições de teste continuam válidos (PAULA FILHO, 2000).

Testes de regressão feitos com base na bateria de testes de aceitação são particularmente importantes durante a manutenção, pois nesta fase é mais frequente que as alterações realizadas afetem outras porções do código. São usados também durante a integração, para confirmar a estabilidade dos módulos já integrados anteriormente (PAULA FILHO, 2000).



## 2. OUTROS TIPOS DE TESTE DE SOFTWARE

A seguir são apresentados os tipos de testes referentes às características do software.

### 2.1 TESTE DE CARGA

O teste de carga é responsável por testar se o software funciona como o esperado em diversas condições de carga de trabalho.

Esses testes podem ser executados da seguinte forma (BARTIÉ, 2002):

- Elevando e reduzindo sucessivamente o número de transações simultâneas;
- Aumentando e reduzindo o tráfego de rede;
- Aumentando o número de usuários simultâneos;
- Combinando todos esses elementos.

### 2.2 TESTE DE CONFIGURAÇÃO

O teste de configuração é responsável por testar se o software funciona corretamente nas configurações de hardware e software exigidos.

Esses testes podem ser executados da seguinte forma (BARTIÉ, 2002):

- Variando os sistemas operacionais;
- Variando os browsers;

- Variando os hardwares que irão interagir com a solução;
- Combinando todos esses elementos.

## 2.3 TESTE DE DESEMPENHO/PERFORMANCE

O teste de desempenho é responsável por testar se o desempenho do sistema integrado é adequado de acordo com os requisitos do sistema.

Testes de desempenho podem ser realizados ao longo de todos os passos do processo de teste, mas somente quando todos os componentes do sistema estão plenamente integrados é que se pode ter uma medida real do desempenho do sistema (PRESSMAN, 1995).

Esses testes podem ser executados da seguinte forma (BARTIÉ, 2002):

- Validar todos os requisitos de desempenho identificados;
- Simular n usuários acessando simultaneamente a mesma informação;
- Simular n usuários processando simultaneamente a mesma transação;
- Simular n% de tráfego de rede;
- Combinar todos esses elementos.

## 2.4 TESTE DE INSTALAÇÃO

O teste de instalação é responsável por testar se o software foi instalado corretamente em diferentes configurações de hardware exigido.

Esses testes podem ser executados da seguinte forma (BARTIÉ, 2002):

- Realizar a primeira instalação do software;

- Realizar a instalação de um software já instalado;
- Realizar a instalação de atualização de um software;
- Realizar a instalação de software em vários ambientes distintos;
- Realizar todas as alternativas de instalação;
- Validar pré-requisitos de instalação de software.

## 2.5 TESTE DE RECUPERAÇÃO

O teste de recuperação é responsável por testar se o sistema continua suas operações após a perda de integridade do mesmo.

Este teste faz o software falhar de várias formas para verificar se este consegue se recuperar adequadamente (PRESSMAN, 1995).

Esses testes podem ser executados da seguinte forma (BARTIÉ, 2002):

- Interromper o acesso a rede (por alguns instantes, por um longo período);
- Interromper o processamento (desligar o micro, desligar o servidor);
- Gerar arquivos, cancelar o processamento e avaliar se existem arquivos gerados.

## 2.6 TESTE DE SEGURANÇA

O teste de segurança é responsável por testar se os mecanismos de proteção de um sistema realmente o protegerão de acessos não autorizados (PRESSMAN, 1995).

Esses testes podem ser executados da seguinte forma (BARTIÉ, 2002):

- Validar todos os requisitos de segurança identificados;
- Tentar acessar funcionalidades e informações que requerem perfil avançado;
- Tentar invadir/derrubar o servidor de dados/internet;
- Tentar extrair backups de informações sigilosas;
- Tentar descobrir senhas e quebrar protocolos de segurança;
- Tentar processar transações geradas de fontes inexistentes;
- Tentar simular comportamento/infecção por vírus.

## 2.7 TESTE DE USABILIDADE

O teste de usabilidade é responsável por testar se o software está de fácil entendimento e manuseio pelos usuários.

Esses testes podem ser executados da seguinte forma (BARTIÉ, 2002):

- Entrar em cada tela e avaliar a facilidade de navegação entre elas;
- Realizar n operações e depois desfazê-las;
- Realizar procedimentos críticos e avaliar mensagem de alerta;
- Avaliar número de passos para realizar as principais operações;
- Avaliar a existência de ajuda em todas as telas;
- Realizar n buscas no manual de ajuda e validar os procedimentos sugeridos.

## 2.8 TESTE DE VOLUME

O teste de volume é responsável por testar se o software consegue lidar com grandes volumes de dados.

Esses testes podem ser executados da seguinte forma (BARTIÉ, 2002):

- Aumentando sucessivamente o volume de transações;
- Aumentando sucessivamente o volume de consultas;
- Aumentando sucessivamente o tamanho de arquivos a serem processados.

## 2.9 TESTE PARALELO

O teste paralelo é responsável por testar se os resultados do sistema atual são consistentes com o processamento do antigo sistema.

Para realizar este teste podem-se rodar os mesmos dados em duas versões de aplicações, para ver se os dados de saída são iguais.

## 2.10 TESTES AUTOMATIZADOS

Muitas empresas de pequeno e médio porte fazem os testes manualmente. Porém, esta não é a melhor forma de realizá-los, pois ela é demorada e pode estar sujeita a falhas humanas. O ideal seria utilizar processos automatizados, onde é possível executar uma maior quantidade de testes em menos tempo.

O teste automatizado utiliza ferramentas para avaliar se o software que está sendo desenvolvido funciona como esperado. Através dele é possível validar os requisitos de um sistema apontando os erros existentes.

As vantagens da automação dos testes são (WK, 2003):

- Aumento da produtividade e diminuição do custo destinado às atividades de testes;
- Eliminação do trabalho repetitivo de inserção de dados e observação dos resultados;
- Melhoria da qualidade do processo e produto de software;
- Maior confiança quanto à liberação do produto, a cada nova versão;
- Reusabilidade dos scripts de testes, com grande facilidade para alteração de informações;
- Diminuição do tempo gasto com a execução dos testes (se comparado aos testes manuais);
- Aumento da cobertura (abrangência) dos testes, através do número crescente de implementações de casos de testes.

“Para a realização de testes automatizados, existem ferramentas prontas que auxiliam na criação, desenvolvimento e manutenção desses testes. Elas permitem que scripts com testes automatizados sejam criados, alterados e executados” (MACHADO, 2009).

A utilização de ferramentas de teste é fundamental para o sucesso do projeto, no entanto, a tarefa de definição de scripts de teste eficazes necessita de profissional qualificado. Além de qualificação para definição do domínio dos testes, é necessário ainda domínio da ferramenta utilizada e tempo para configuração de cada teste, assim como análise dos resultados obtidos versus resultados esperados (LEAL, 2009).

Para o sucesso dos testes automatizados, Carvalho (2010) apresenta alguns pontos importantes, são eles:

- Apresentar metas realísticas e uma maneira de mensurar o realizado versus o planejado;
- A migração da automação em uma organização deve ser realizada de forma gradativa, não é imediata;
- Importância da parceria entre as equipes de testes manuais e testes automáticos;
- Comunicação periódica aos participantes do cronograma atual, bem como, o planejamento futuro para atingir as metas desejadas;
- Estabelecer métodos de verificação dos resultados dos testes;
- Tratar a automação como um projeto de desenvolvimento;
- Utilizar corretamente e de forma completa os recursos disponíveis na ferramenta de automação.

## 2.11 METODOLOGIAS ÁGEIS

As metodologias ágeis são conjuntos de práticas e métodos, criadas com o objetivo de tornar o desenvolvimento do software mais rápido e com custo mais acessível.

Nessas metodologias é dado maior valor aos participantes do projeto e suas interações ao invés de processos e ferramentas de desenvolvimento. Além disso, existe a preocupação de gastar menos tempo com documentação e mais com a implementação (SOARES, 2004).

Outra característica dos métodos ágeis é que é oferecido ao desenvolvedor total flexibilidade. Com isso, é possível se adaptar facilmente a novas mudanças, ao invés de procurar analisar previamente tudo o que pode acontecer no decorrer do desenvolvimento (SOARES, 2004).

Também são entregues constantemente partes operacionais do software. Assim, o cliente não precisa esperar muito para ver o software funcionando, como nas metodologias tradicionais (SOARES, 2004).

As principais metodologias ágeis são a XP e a Scrum, que serão vistos a seguir.

### *2.11.1 Extreme Programming*

“A Extreme Programming (XP) é uma metodologia ágil destinada para equipes pequenas e médias que desenvolvem software baseado em requisitos vagos e que se modificam rapidamente” (BECK, 1999 apud SOARES, 2004).

A XP enfatiza o desenvolvimento rápido do projeto, visa garantir a satisfação do cliente, além de favorecer e priorizar o comprimento das estimativas (SOARES, 2004).

[...] a XP não deve ser aplicado a qualquer tipo de projeto. O grupo de desenvolvedores deve formar uma equipe de 2 a 10 integrantes, que devem estar por dentro de todas as fases do desenvolvimento. É necessário realizar vários testes, às vezes, alterar o projeto em decorrência destes. A equipe tem de ser bastante interessada e pró-ativa, para assegurar a alta produtividade, e o cliente deve estar sempre disponível para tirar dúvidas e tomar decisões em relação ao projeto (SOUZA, 2007).

Segundo Beck (1999), a XP possui quatro valores: comunicação, simplicidade, feedback e coragem (BECK, 1999 apud SOARES, 2004).

A comunicação tem o objetivo de manter o melhor relacionamento possível entre clientes e desenvolvedores, preferindo conversas pessoais ao invés de outros meios de comunicação. A simplicidade visa permitir a criação de código simples e procura implementar apenas requisitos atuais (SOARES, 2004).

O feedback tem como objetivo testar constantemente o software, para que o programador tenha informações constantes do código. E o cliente deve sugerir



constantemente novas características, para que o produto final esteja de acordo com as necessidades reais do cliente (SOARES, 2004).

É necessário coragem para implementar os três valores anteriores, pois nem todos os profissionais da área tem facilidade de comunicação, nem todos conseguem produzir programas simples e o feedback não é uma tarefa fácil por parte de certos profissionais (SOARES, 2004).

Conforme Beck (1999), a XP possui 12 práticas que devem ser seguidas na modelagem de um sistema. São elas (BECK, 1999 apud SOARES, 2004):

- **Planejamento:** deve ser decidido o que é necessário ser feito e o que pode ser adiado no projeto, pois a XP baseia-se em requisitos atuais e não em requisitos futuros.
- **Entregas frequentes:** a construção do software deve ser simples, e conforme os requisitos surgem, o mesmo deve ser atualizado. Cada versão entregue deve ter o menor tamanho possível, contendo os requisitos mais importantes.
- **Metáfora:** o software não deve ser descrito com o uso de termos técnicos.
- **Projeto simples:** o programa desenvolvido deve ser simples e satisfazer os requisitos atuais.
- **Testes:** o projeto deve ser validado durante todo o processo de desenvolvimento. Os programadores desenvolvem o software criando primeiramente os testes.
- **Programação em pares:** a implementação do código deve ser feita em dupla. Enquanto um implementa o código, o outro observa continuamente o trabalho que está sendo feito, procurando identificar erros sintáticos e semânticos e pensando em como melhorar o código.
- **Refatoração:** é o aperfeiçoamento do projeto do software. A refatoração deve ser feita apenas quando é possível simplificar o módulo atual sem perder nenhuma funcionalidade.

- **Propriedade coletiva:** o código deve ser desenvolvido por toda a equipe de desenvolvimento. Qualquer pessoa pode incluir algo para melhorar uma funcionalidade, desde que faça a bateria de testes necessária.
- **Integração contínua:** o sistema deve ser construído várias vezes por dia, mantendo os programadores em sintonia, além de possibilitar processos rápidos. Ao integrar apenas um conjunto de modificações de cada vez fica evidente quem deve fazer as correções quando os testes falham.
- **40 horas de trabalho semanal:** não deve ser feito horas extras constantemente. Para isto deve-se melhorar o planejamento ao invés de sobrecarregar as pessoas.
- **Cliente presente:** o cliente deve estar sempre disponível para tirar as dúvidas de requisitos, para que não tenha atrasos e construções erradas.
- **Código padrão:** o código deve ser padronizado para que este possa ser compartilhado entre todos os participantes da equipe.

### 2.11.2 SCRUM

A Scrum é uma metodologia ágil que é definida como “um processo de desenvolvimento iterativo e incremental que pode ser aplicado a qualquer tipo de produto (software) ou no gerenciamento de qualquer atividade complexa” (BISSI, 2007).

A metodologia Scrum estabelece um conjunto de regras e práticas de gestão que devem ser adotadas para garantir o sucesso de um projeto. Centrado no trabalho em equipe, melhora a comunicação e maximiza a cooperação, permitindo que cada um faça o seu melhor e se sinta bem com o que faz, o que mais tarde se reflete num aumento de produtividade. Englobando processos de engenharia, esse método não requer nem fornece qualquer técnica ou método específico para a fase de desenvolvimento de software (BISSI, 2007).

A Scrum é baseada em pequenos ciclos de tempo, normalmente 30 dias, chamados Sprints, onde se trabalha para alcançar objetivos bem definidos (SANCHEZ, 2007).

Os principais papéis da Scrum são a Equipe (geralmente entre 5 e 9 pessoas) que são responsáveis por entregar o projeto e trabalhar de forma auto-gerenciada. O Product Owner, responsável pela visão de negócio do projeto. É quem define as prioridades. E também o Scrum Master que é mistura de gerente, facilitador e mediador. Deve identificar e remover os obstáculos e assegurar que as práticas Scrum serão executadas (SANCHEZ, 2007).

Na metodologia Scrum são realizadas reuniões diárias de 30 minutos e cada integrante da equipe precisa responder as questões: o que foi feito ontem? O que planeja fazer para amanhã? Tem algum impedimento? (MATSUDO, 2009).

O objetivo da reunião é que todos os integrantes da equipe saibam o que cada um está fazendo e se existe algum impedimento que o Scrum Master deverá solucionar (MATSUDO, 2009).

Ao final de cada Sprint é realizada uma reunião para que o Product Owner verifique se o desenvolvimento foi feito corretamente. Também é feito uma reunião para encontrar os pontos bons e ruins que aconteceram durante o Sprint e levar como aprendizado, melhorando o processo, o entrosamento da equipe e o projeto em desenvolvimento (MATSUDO, 2009).

### *2.11.3 Teste de Software nas Metodologias Ágeis*

A maneira de como o teste de software é realizada nas metodologias ágeis é o mesmo do que o das metodologias tradicionais. Porém, nas metodologias ágeis é realizado mais testes, devido a pouca documentação existente.

Nos processos guiados por planos, existe uma série extensa de artefatos documentais resultantes de uma análise profunda sobre o sistema a ser desenvolvido. Os testes são utilizados nesse caso apenas como mais um

artefato produzido pelo processo visando garantir a detecção de erros antes da liberação da versão final (LEAL, 2009).

Já nos processos ágeis, não existe todo esse aparato documental, uma vez que processos ágeis priorizam código executável ao invés de produção extensa de documentação. O papel dos testes é muito importante para o sucesso da metodologia ágil, é um ponto essencial para garantia de que exista fidelidade das implementações em relação aos requisitos (LEAL, 2009).

### 3. ESTUDO DE CASO

Com base no conhecimento obtido através das leituras realizadas, foi elaborado um questionário de teste de software, com o objetivo de verificar os métodos de testes existentes nas empresas de software na área de TI.

O questionário foi criado no Google Docs, conforme o APÊNDICE A, e para a realização da pesquisa, foi encaminhado este questionário por e-mail para 267 empresas de desenvolvimento de software, sendo que 245 são empresas associadas no Sindicato das Empresas de Informática do Rio Grande do Sul (SEPRORGS).

A coleta de dados da pesquisa durou aproximadamente um mês e destas 267 empresas, 14 empresas responderam o questionário, sendo 12 empresas de desenvolvimento de software e 2 empresas que fornecem serviços de testes de software para as organizações.

Comparando com o trabalho realizado por Vanessa Schuster (2009), obteve-se um melhor resultado, pois, Schuster (2009) encaminhou o questionário para 284 empresas e apenas duas empresas responderam.

A seguir estão descritos detalhadamente as respostas de cada empresa, porém ocultado o nome das mesmas. Os gráficos das respostas estão no APÊNDICE B. As figuras (gráficos) e as tabelas apresentadas neste trabalho são de autoria própria.

#### **a) Empresa A**

A empresa A atua no mercado a mais de 10 anos, possui de 10 a 20 funcionários e presta serviços para empresas regionais.

Essa empresa realiza a atividade de teste desde que começou a desenvolver softwares, realizando teste de caixa branca e teste de sistema. Para os testes de caixa branca utiliza os critérios baseados em fluxo de dados.

Os responsáveis por realizar os testes são os próprios desenvolvedores do sistema e uma equipe de testes da empresa, sendo que a execução dos testes é iniciada na fase de implementação do software.

A maior incidência dos erros está concentrada no meio do processo de desenvolvimento do software e são os erros de funcionalidade os mais comuns de serem encontrados nos sistemas desenvolvidos.

A dificuldade encontrada pela empresa em realizar os testes é o fato de a atividade de teste ser limitada por restrições de cronograma.

A empresa realiza os testes manualmente e a atividade de teste não é documentada.

O percentual de tempo que é consumido com a atividade de teste é de até 5%. Os testes apresentam grande contribuição, sendo que houve diminuição dos custos e dos erros encontrados após a implantação da atividade de teste.

## **b) Empresa B**

A empresa B atua no mercado a mais de 10 anos, possui mais de 50 funcionários e presta serviços para empresas regionais.

A atividade de teste de software foi implantada na empresa há menos de 1 ano, sendo que é realizado o teste de caixa branca, de integração, de configuração, de desempenho/performance e teste de migração de banco de dados. Para os testes de caixa branca utiliza os critérios baseados em fluxo de controle e os critérios baseados em fluxo de dados.

Os responsáveis por realizar os testes são os próprios desenvolvedores do sistema e uma equipe de testes da empresa, sendo que a execução dos testes é iniciada na fase de implementação do software.

A maior incidência dos erros está concentrada no meio do processo de desenvolvimento do software e são os erros de interpretação de análise de

requisitos e os erros de código os mais comuns de serem encontrados nos sistemas desenvolvidos.

As dificuldades encontradas pela empresa em realizar os testes são pela falta de profissionais especializados na área de testes e pelo desconhecimento de um procedimento de teste adequado.

A empresa realiza os testes manualmente e de forma automatizada. Segundo a empresa, as vantagens dos testes automatizados em relação aos testes feitos de forma manual são a eliminação do trabalho repetitivo de inserção de dados e observação dos resultados; a melhora da qualidade do processo e produto de software e o aumento da produtividade e diminuição do custo destinado às atividades de teste.

Os documentos utilizados pela empresa são a especificação do projeto de teste, a especificação de caso de teste, o relatório de incidente de teste e o relatório de encaminhamento de itens de teste.

O percentual de tempo que é consumido com a atividade de teste é de 5% a 10%. Os testes apresentam pequena contribuição, sendo que houve aumento nos custos e diminuição dos erros encontrados após a implantação da atividade de teste.

### **c) Empresa C**

A empresa C atua no mercado de 7 a 10 anos, possui de 30 a 40 funcionários e presta serviços para empresas de médio e grande porte sem restrições geográficas.

Essa empresa realiza a atividade de teste desde que começou a desenvolver softwares, realizando teste de caixa preta, de caixa branca, de integração, de sistema, de carga, de configuração, de desempenho/performance, de instalação, de recuperação, de segurança e teste de usabilidade.

Para os testes de caixa preta utiliza os critérios de particionamento de equivalência e análise de valor limite. E para os testes de caixa branca utiliza os critérios baseados em fluxo de dados e os critérios baseados na complexidade.

Os responsáveis por realizar os testes são os próprios desenvolvedores do sistema, sendo que a execução dos testes é iniciada na fase de elaboração do projeto do software.

A maior incidência dos erros está concentrada nas fases iniciais do processo de desenvolvimento do software e são os erros de interpretação de análise de requisitos e os erros de funcionalidade os mais comuns de serem encontrados nos sistemas desenvolvidos.

A empresa não possui dificuldades em realizar a atividade de teste.

Os testes são feitos manualmente e os documentos utilizados pela empresa são o plano de teste, a especificação do projeto de teste, a especificação de procedimento de teste, o diário de teste, o relatório de incidente de teste, o relatório-resumo de teste e o relatório de encaminhamento de itens de teste.

O percentual de tempo que é consumido com a atividade de teste é de 20% a 30%. Os testes apresentam grande contribuição, sendo que houve diminuição dos custos e dos erros encontrados após a implantação da atividade de teste.

#### **d) Empresa D**

A empresa D atua no mercado de 7 a 10 anos, possui menos de 10 funcionários e presta serviços para empresas regionais.

A atividade de teste de software foi implantada na empresa de 1 a 3 anos, sendo que é realizado o teste de caixa preta, o teste de aceitação e o teste de cobertura (code coverage). A empresa desconhece os critérios de caixa preta apresentados no questionário.



Os responsáveis por realizar os testes são os próprios desenvolvedores do sistema e os clientes, sendo que a execução dos testes é iniciada após a conclusão das implementações.

A maior incidência dos erros está concentrada nas fases finais do processo de desenvolvimento do software e são os erros de código os mais comuns de serem encontrados nos sistemas desenvolvidos.

As dificuldades encontradas pela empresa em realizar os testes são pela falta de profissionais especializados na área de testes e pelo fato de a atividade de teste ser limitada por restrições de cronograma.

A empresa realiza os testes manualmente e de forma automatizada, sendo que utiliza a ferramenta Emma para a realização dos testes automatizados.

De acordo com a empresa, as vantagens dos testes automatizados em relação aos testes feitos de forma manual são a maior confiança quanto à liberação do produto, a cada nova versão.

A atividade de teste não é documentada. O percentual de tempo que é consumido com a atividade de teste é de até 5%. Os testes apresentam grande contribuição, sendo que houve diminuição dos custos e dos erros encontrados após a implantação da atividade de teste.

#### **e) Empresa E**

A empresa E atua no mercado de 3 a 5 anos, possui menos de 10 funcionários e presta serviços para diversas empresas.

A atividade de teste de software foi implantada na empresa de 1 a 3 anos, sendo que é realizado o teste de caixa preta e o teste de caixa branca. A empresa desconhece os critérios de caixa preta e de caixa branca apresentados no questionário.

Os responsáveis por realizar os testes são os próprios desenvolvedores do sistema e uma equipe de testes da empresa, sendo que a execução dos testes é iniciada após a implementação do software.

A maior incidência dos erros está concentrada nas fases finais do processo de desenvolvimento do software e são os erros de código e os erros de digitação os mais comuns de serem encontrados nos sistemas desenvolvidos.

A dificuldade encontrada pela empresa em realizar os testes é pelo fato de o teste ser um processo caro.

Os testes são feitos manualmente e o documento utilizado pela empresa é o plano de teste.

O percentual de tempo que é consumido com a atividade de teste é de 5% a 10%. Os testes apresentam grande contribuição, sendo que houve diminuição dos custos e dos erros encontrados após a implantação da atividade de teste.

#### **f) Empresa F**

A empresa F atua no mercado a mais de 10 anos, possui de 10 a 20 funcionários e presta serviços para empresas de outros estados.

A atividade de teste de software foi implantada na empresa de 1 a 3 anos, sendo que é realizado o teste de caixa preta, de caixa branca, de unidade, de integração, de sistema, de aceitação, de instalação e teste de usabilidade.

Para os testes de caixa preta utiliza os critérios de particionamento de equivalência e análise de valor limite. E para os testes de caixa branca utiliza os critérios baseados em fluxo de controle e os critérios baseados em fluxo de dados.

Os responsáveis por realizar os testes são os próprios desenvolvedores do sistema e uma equipe de testes da empresa, sendo que a execução dos testes é iniciada na elaboração do projeto de software.

A maior incidência dos erros está concentrada no meio do processo de desenvolvimento do software e são os erros de interpretação de análise de requisitos e os erros de código os mais comuns de serem encontrados nos sistemas desenvolvidos.

A dificuldade encontrada pela empresa em realizar os testes é pelo fato de o teste ser um processo caro.

Os testes são feitos manualmente e os documentos utilizados pela empresa são o plano de teste, o relatório de incidente de teste e o relatório de encaminhamento de itens de teste.

O porcentual de tempo que é consumido com a atividade de teste é de 10% a 20%. Os testes apresentam grande contribuição, sendo que houve aumento nos custos e diminuição dos erros encontrados após a implantação da atividade de teste.

#### **g) Empresa G**

A empresa G atua no mercado a mais de 10 anos, possui menos de 10 funcionários e presta serviços para empresas regionais.

A atividade de teste de software foi implantada na empresa a mais de 10 anos, sendo que é realizado o teste de caixa preta. A empresa desconhece os critérios de caixa preta apresentados no questionário.

Os responsáveis por realizar os testes são os próprios desenvolvedores do sistema, uma equipe de testes da empresa, os clientes e os usuários, sendo que a execução dos testes é iniciada na fase de codificação.

A maior incidência dos erros está concentrada no meio do processo de desenvolvimento do software e são os erros de interpretação de análise de requisitos, os erros de código e os erros de funcionalidade os mais comuns de serem encontrados nos sistemas desenvolvidos.

As dificuldades encontradas pela empresa em realizar os testes são pela falta de profissionais especializados na área de testes, pela dificuldade em implantar um

processo de teste, pelo fato de o teste ser um processo caro e pelo fato de a atividade de teste ser limitada por restrições de cronograma.

A empresa realiza os testes manualmente e a atividade de teste não é documentada.

O percentual de tempo que é consumido com a atividade de teste é de 20% a 30%. Os testes apresentam média contribuição, sendo que houve diminuição dos custos e dos erros encontrados após a implantação da atividade de teste.

#### **h) Empresa H**

A empresa H atua no mercado a mais de 10 anos, possui mais de 50 funcionários e presta serviços para empresas regionais.

A atividade de teste de software foi implantada na empresa de 3 a 5 anos, sendo que é realizado o teste de caixa preta, de integração, de sistema, de regressão, de carga e teste de desempenho/performance. Para os testes de caixa preta utiliza os critérios de particionamento de equivalência, análise de valor limite e grafo de causa e efeito.

Os responsáveis por realizar os testes são a equipe de testes da empresa, sendo que a execução dos testes é iniciada na especificação dos requisitos.

A maior incidência dos erros está concentrada no meio do processo de desenvolvimento do software e são os erros de interpretação de análise de requisitos, os erros de funcionalidade e os erros de interface os mais comuns de serem encontrados nos sistemas desenvolvidos.

As dificuldades encontradas pela empresa em realizar os testes são pelo desconhecimento de um procedimento de teste adequado, pelo desconhecimento de técnicas de teste adequadas e pelo fato de a atividade de teste ser limitada por restrições de cronograma.

Os testes são feitos manualmente e os documentos utilizados pela empresa são o plano de teste, a especificação de caso de teste e o relatório de incidente de teste.

O percentual de tempo que é consumido com a atividade de teste é de 10% a 20%. Os testes apresentam média contribuição, sendo que houve diminuição dos erros encontrados após a implantação da atividade de teste.

#### **i) Empresa I**

A empresa I atua no mercado a mais de 10 anos, possui mais de 50 funcionários e presta serviços para diversas empresas.

A atividade de teste de software foi implantada na empresa a mais de 10 anos, sendo que é realizado o teste de caixa preta, de caixa branca, de unidade, de integração, de sistema, de aceitação, de regressão, de carga, de configuração, de desempenho/performance, de instalação, de segurança, de usabilidade, e teste de volume.

Para os testes de caixa preta utiliza os critérios de particionamento de equivalência, análise de valor limite e grafo de causa e efeito. E para os testes de caixa branca utiliza os critérios baseados em fluxo de controle, critérios baseados em fluxo de dados e os critérios baseados na complexidade.

Os responsáveis por realizar os testes são a equipe de testes da empresa, sendo que a execução dos testes é iniciada na especificação de requisitos.

A maior incidência dos erros está concentrada no meio do processo de desenvolvimento do software e são os erros de código os mais comuns de serem encontrados nos sistemas desenvolvidos.

A empresa não possui dificuldades em realizar a atividade de teste.

A empresa realiza os testes manualmente e de forma automatizada. Segundo a empresa, as vantagens dos testes automatizados em relação aos testes feitos de forma manual são a eliminação do trabalho repetitivo de inserção de dados e

observação dos resultados; a melhora da qualidade do processo e produto de software; a maior confiança quanto a liberação do produto, a cada nova versão; a reusabilidade dos scripts de testes, com grande facilidade para alteração de informações; diminuição do tempo gasto com a execução dos testes; o aumento da produtividade e diminuição do custo destinado às atividades de teste; e o aumento da cobertura dos testes, através do número crescente de implementações de casos de testes.

Os documentos utilizados pela empresa são o plano de teste, a especificação do projeto de teste, a especificação de caso de teste, a especificação de procedimento de teste, o diário de teste, o relatório de incidente de teste, o relatório-resumo de teste e o relatório de encaminhamento de itens de teste.

O percentual de tempo que é consumido com a atividade de teste é acima de 50%. Os testes apresentam grande contribuição, sendo que houve diminuição dos custos e dos erros encontrados após a implantação da atividade de teste.

#### **j) Empresa J**

A empresa J atua no mercado a mais de 10 anos, possui de 30 a 40 funcionários e presta serviços para o setor público, prefeituras, fundações, autarquias e câmara de vereadores.

A atividade de teste de software foi implantada na empresa de 1 a 3 anos, sendo que é realizado o teste de caixa preta, de sistema, de desempenho/performance, de usabilidade, de volume e teste paralelo. Para os testes de caixa preta utiliza o critério de grafo de causa e efeito.

Os responsáveis por realizar os testes são a equipe de testes da empresa, sendo que a execução dos testes é iniciada na fase de codificação. E são os erros de código, os erros de funcionalidade e os erros de interface os mais comuns de serem encontrados nos sistemas desenvolvidos.

As dificuldades encontradas pela empresa em realizar os testes são pela dificuldade em implantar um processo de teste e pelo fato de a atividade de teste ser limitada por restrições de cronograma.

A empresa realiza os testes manualmente e de forma automatizada, sendo que utiliza a ferramenta Selenium para a realização dos testes automatizados.

De acordo com a empresa, as vantagens dos testes automatizados em relação aos testes feitos de forma manual são a eliminação do trabalho repetitivo de inserção de dados e observação dos resultados; a melhora da qualidade do processo e produto de software; a reusabilidade dos scripts de testes, com grande facilidade para alteração de informações; diminuição do tempo gasto com a execução dos testes; e o aumento da produtividade e diminuição do custo destinado às atividades de teste.

A atividade de teste não é documentada. O percentual de tempo que é consumido com a atividade de teste é de 10% a 20%. Os testes apresentam grande contribuição, sendo que houve aumento nos custos e diminuição dos erros encontrados após a implantação da atividade de teste.

#### **k) Empresa K**

A empresa K atua no mercado a mais de 10 anos, possui mais de 50 funcionários e presta serviços para empresas de outros estados.

A atividade de teste de software foi implantada na empresa de 1 a 3 anos, sendo que é realizado o teste de unidade, de integração, de sistema, de aceitação, de regressão, de carga, de desempenho/performance, de recuperação, de segurança e teste de usabilidade.

Os responsáveis por realizar os testes são a equipe de testes da empresa, sendo que a execução dos testes é iniciada após o software estar pronto com devidas melhorias e correções.

Os erros ocorrem durante todo o processo de desenvolvimento do software e são os erros de interpretação de análise dos requisitos, os erros de código, os erros de funcionalidade, os erros de digitação e os erros de interface os mais comuns de serem encontrados nos sistemas desenvolvidos.

As dificuldades encontradas pela empresa em realizar os testes são pela falta de profissionais especializados na área de testes, pela dificuldade em implantar um processo de teste, e pelo fato de o teste ser um processo caro.

A empresa realiza os testes manualmente e de forma automatizada, sendo que utiliza a ferramenta TestComplete para a realização dos testes automatizados.

De acordo com a empresa, as vantagens dos testes automatizados em relação aos testes feitos de forma manual são a eliminação do trabalho repetitivo de inserção de dados e observação dos resultados; a melhora da qualidade do processo e produto de software; a maior confiança quanto a liberação do produto, a cada nova versão; a reusabilidade dos scripts de testes, com grande facilidade para alteração de informações; diminuição do tempo gasto com a execução dos testes; o aumento da produtividade e diminuição do custo destinado às atividades de teste; e o aumento da cobertura dos testes, através do número crescente de implementações de casos de testes.

Os documentos utilizados pela empresa são o plano de teste, a especificação do projeto de teste, a especificação de caso de teste, a especificação de procedimento de teste, o relatório de incidente de teste e o relatório-resumo de teste.

O percentual de tempo que é consumido com a atividade de teste é de 20% a 30%. Os testes apresentam grande contribuição, sendo que houve aumento nos custos e diminuição dos erros encontrados após a implantação da atividade de teste.

### **I) Empresa L**

A empresa L atua no mercado a mais de 10 anos, possui de 10 a 20 funcionários e presta serviços para empresas regionais.



A atividade de teste de software foi implantada na empresa de 1 a 3 anos, sendo que é realizado o teste de caixa preta, teste de unidade e teste de regressão. A empresa desconhece os critérios de caixa preta apresentados no questionário.

Os responsáveis por realizar os testes são a equipe de testes da empresa e os próprios desenvolvedores do sistema, sendo que a execução dos testes é iniciada na fase de codificação.

A maior incidência dos erros está concentrada no meio do processo de desenvolvimento do software e são os erros de interpretação de análise dos requisitos e os erros de código os mais comuns de serem encontrados nos sistemas desenvolvidos.

As dificuldades encontradas pela empresa em realizar os testes são pela falta de profissionais especializados na área de testes, pela dificuldade em implantar um processo de teste, e pelo desconhecimento de técnicas de teste adequadas.

Os testes são feitos manualmente e os documentos utilizados pela empresa são o plano de teste, a especificação de caso de teste e planilha de teste.

O percentual de tempo que é consumido com a atividade de teste é de 10% a 20%. Os testes apresentam grande contribuição, sendo que houve diminuição dos custos e dos erros encontrados após a implantação da atividade de teste.

#### **m) Empresa M**

A empresa M não desenvolve software, apenas realiza os testes para as empresas. Ela atua no mercado de 3 a 5 anos, possui de 20 a 30 funcionários e presta serviços de testes para empresas locais, regionais, de outros estados e de outros países.

Essa empresa realiza a atividade de teste desde que começou a atuar no mercado, realizando os testes de caixa preta, de sistema, de aceitação, de regressão, de carga, de configuração, de desempenho/performance, de instalação, de recuperação, de segurança, de usabilidade, de volume e teste paralelo. Para os

testes de caixa preta utiliza os critérios de particionamento de equivalência, análise de valor limite e grafo de causa e efeito.

A equipe de testes da empresa inicia a execução dos testes na especificação dos requisitos, sendo que a maior incidência dos erros está concentrada no meio do processo de desenvolvimento.

As dificuldades encontradas pela empresa em realizar os testes são pela falta de profissionais especializados na área de testes, pela dificuldade em implantar um processo de testes em alguns clientes, e pelo fato de a atividade de teste ser limitada por restrições de cronograma.

A empresa realiza os testes manualmente e de forma automatizada. Sendo que para o teste de caixa preta utiliza as ferramentas TestComplete, Sikuli, Rational Robot e Selenium. Para o teste de performance usa o Loadster, JMeter e Weblod e para o teste de segurança utiliza o Netsparker.

De acordo com a empresa, as vantagens dos testes automatizados em relação aos testes feitos de forma manual são a eliminação do trabalho repetitivo de inserção de dados e observação dos resultados; a reusabilidade dos scripts de testes, com grande facilidade para alteração de informações; diminuição do tempo gasto com a execução dos testes; o aumento da produtividade e diminuição do custo destinado às atividades de teste; e o aumento da cobertura dos testes, através do número crescente de implementações de casos de testes.

Os documentos utilizados pela empresa são o plano de teste, a especificação do projeto de teste, a especificação de caso de teste, a especificação de procedimento de teste, o relatório de incidente de teste e o relatório-resumo de teste.

O percentual de tempo que é consumido com a atividade de teste varia conforme o cliente. Os testes apresentam grande contribuição, sendo que houve diminuição dos erros encontrados após a implantação da atividade de teste.

#### **n) Empresa N**

A empresa N também não desenvolve software, apenas realiza os testes para as empresas. Ela atua no mercado a mais de 10 anos, possui mais de 50 funcionários e presta serviços de testes para empresas de outros estados.

Essa empresa realiza a atividade de teste desde que começou a atuar no mercado, realizando teste de caixa preta, de caixa branca, de unidade, de integração, de sistema, de aceitação, de regressão, de carga, de configuração, de desempenho/performance, de instalação, de recuperação, de segurança, de usabilidade, de volume e teste paralelo.

Para os testes de caixa preta utiliza os critérios de particionamento de equivalência, análise de valor limite e grafo de causa e efeito. E para os testes de caixa branca utiliza os critérios baseados em fluxo de controle, os critérios baseados em fluxo de dados e os critérios baseados na complexidade.

A equipe de testes da empresa inicia a execução dos testes dependendo dos clientes, sendo que a maior incidência dos erros está concentrada no final do processo de desenvolvimento.

A dificuldade encontrada pela empresa em realizar os testes é a falta de entendimento da necessidade de testes de software pelos clientes.

A empresa realiza os testes manualmente e de forma automatizada. Sendo que utiliza as ferramentas Selenium, IBM/Rational, HP, MicroFocus e JMeter para a realização dos testes automatizados.

De acordo com a empresa, as vantagens dos testes automatizados em relação aos testes feitos de forma manual são a eliminação do trabalho repetitivo de inserção de dados e observação dos resultados; a melhora da qualidade do processo e produto de software; a maior confiança quanto a liberação do produto, a cada nova versão; a reusabilidade dos scripts de testes, com grande facilidade para alteração de informações; diminuição do tempo gasto com a execução dos testes; o aumento da produtividade e diminuição do custo destinado às atividades de teste; e

o aumento da cobertura dos testes, através do número crescente de implementações de casos de testes.

Os documentos utilizados pela empresa são o plano de teste, a especificação do projeto de teste, a especificação de caso de teste, a especificação de procedimento de teste, o diário de teste, o relatório de incidente de teste, o relatório-resumo de teste e o relatório de encaminhamento de itens de teste.

O percentual de tempo que é consumido com a atividade de teste é acima de 50%. Os testes apresentam grande contribuição, sendo que houve diminuição dos custos e dos erros encontrados após a implantação da atividade de teste.

### 3.1 ANÁLISE DAS RESPOSTAS DAS EMPRESAS QUE DESENVOLVEM SOFTWARES

Através da análise feita das respostas, pode-se perceber que as empresas de desenvolvimento de software com mais de 10 anos de atuação no mercado, começaram a testar os softwares apenas nos últimos anos. De todas as empresas que responderam, apenas 4 iniciaram a atividade de teste de software no momento em que começaram a atuar no mercado.

A figura 1 mostra que os testes mais realizados pelas empresas são o teste de caixa preta, o teste de sistema, o teste de caixa branca, o teste de integração e o teste de desempenho/performance.

#### 6. Quais testes de softwares são feitos?



Figura 1: Testes de Softwares mais usados.

Os principais responsáveis por realizar os testes são as equipes de testes da empresa e os próprios desenvolvedores do sistema e a maioria dos testes são iniciados na fase de codificação.

A maior incidência dos erros está concentrada no meio do processo de desenvolvimento e os erros mais comuns de serem encontrados são os erros de código.

As principais dificuldades encontradas pelas empresas são a falta de profissionais especializados na área de teste e o fato de a atividade ser limitada por restrições de cronograma.

Todas as empresas realizam os testes manualmente, e destas, cinco empresas também realizam os testes de forma automatizada, conforme mostra a figura 2.

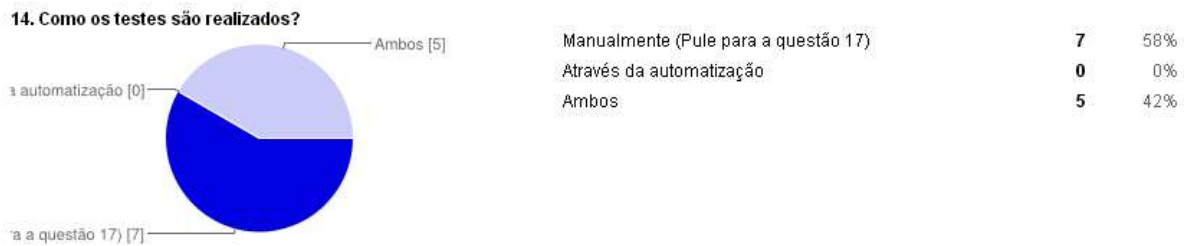


Figura 2: Gráfico de como os testes são realizados.

As principais vantagens dos testes automatizados em relação aos testes feitos manualmente são a eliminação do trabalho repetitivo de inserção de dados e observação dos resultados, a melhoria da qualidade do processo e produto de software e o aumento da produtividade e diminuição do custo destinado às atividades de teste.

A figura 3 mostra que os principais documentos utilizados são o plano de teste e o relatório de incidente de teste.

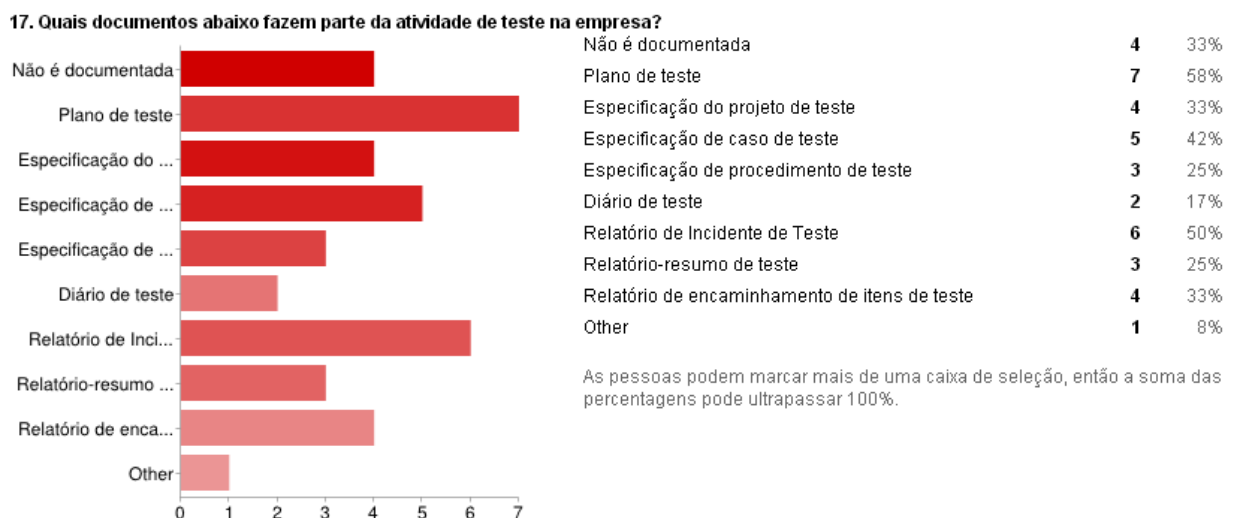


Figura 3: Documentos mais utilizados pelas empresas.

A maioria das empresas gasta em torno de 10% a 20% de tempo com a atividade de teste e para a maioria, os testes de softwares apresentam grande contribuição.

A maioria das empresas responderam que após a implantação da atividade de teste houve diminuição dos custos e para todas as empresas houve diminuição dos erros encontrados após a entrega do produto ao cliente, conforme mostra a figura 4 e 5, respectivamente.

20. Com relação aos custos, após a implantação da atividade de teste observou-se que:

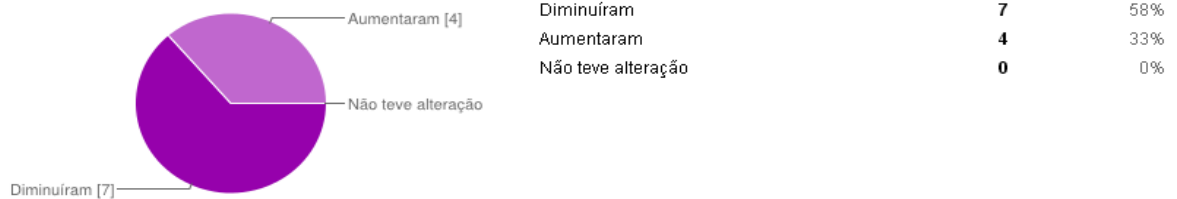


Figura 4: Diminuição dos custos após a implantação da atividade de teste.

21. Depois da implantação da atividade de teste de software, os erros encontrados após a entrega do produto ao cliente:

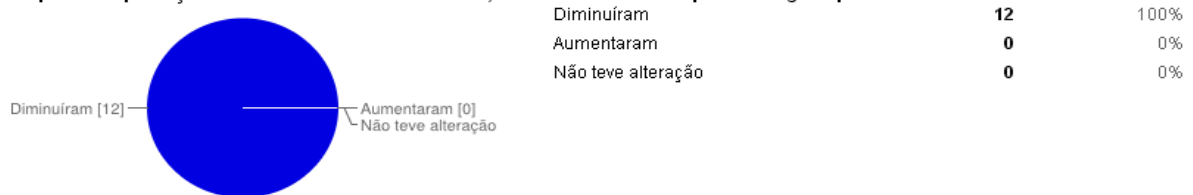


Figura 5: Diminuição dos erros após a implantação da atividade de teste.

### 3.2 ANÁLISE DAS RESPOSTAS DAS EMPRESAS QUE PRESTAM SERVIÇO DE TESTES DE SOFTWARE PARA AS EMPRESAS

Comparando as respostas das duas empresas que prestam serviço de teste de software, pode-se perceber que apenas uma das empresas não realiza os testes de caixa branca, o teste de unidade e o teste de integração, como mostra a figura 6.

6. Quais testes de softwares são feitos?

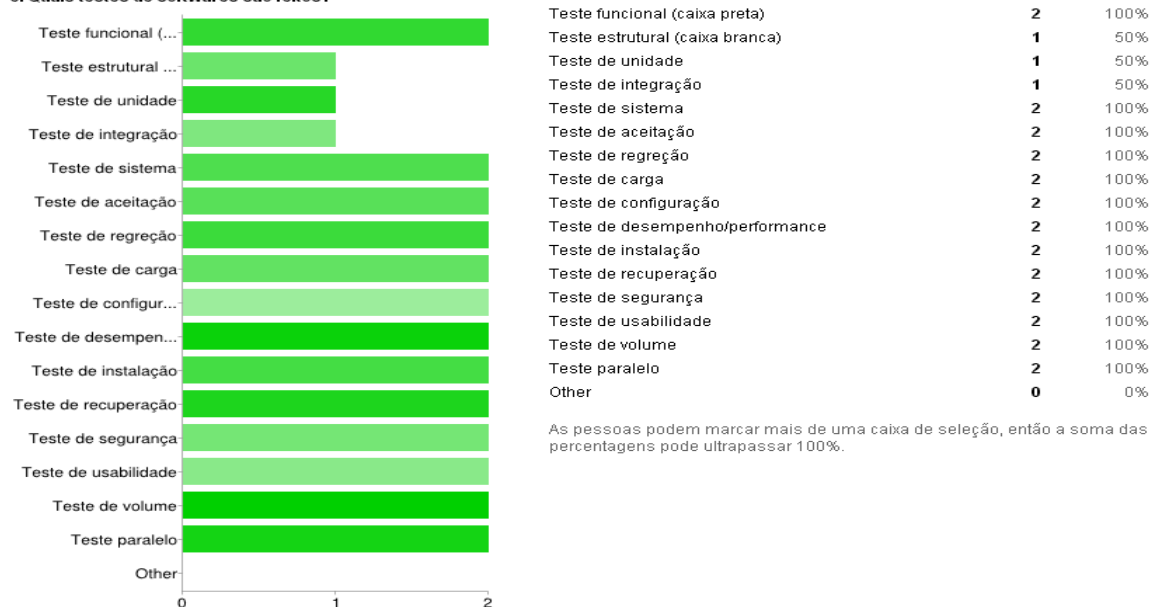
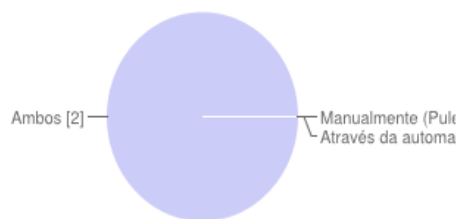


Figura 6: Testes de softwares mais utilizados.

A figura 7 mostra que as duas empresas realizam os testes manualmente e de forma automatizada.

**14. Como os testes são realizados?**



Manualmente (Pule para a questão 17)	0	0%
Através da automa	0	0%
Ambos	2	100%

Figura 7: Gráfico de como são realizado os testes.

Ambas responderam que as principais vantagens dos testes automatizados em relação aos testes feitos manualmente são a eliminação do trabalho repetitivo de inserção de dados e observação dos resultados, a reusabilidade dos scripts de testes, com grande facilidade para alteração de informações, a diminuição do tempo gasto com a execução dos testes, o aumento da produtividade e diminuição do custo destinado às atividades de teste, e o aumento da cobertura (abrangência) dos testes, através do número crescente de implementações de casos de testes.

Os documentos usados pelas duas empresas são o plano de teste, a especificação do projeto de teste, a especificação de caso de teste, a especificação de procedimento de teste, o relatório de incidente de teste e o relatório-resumo de teste, como mostrado a seguir.

**17. Quais documentos abaixo fazem parte da atividade de teste na empresa?**

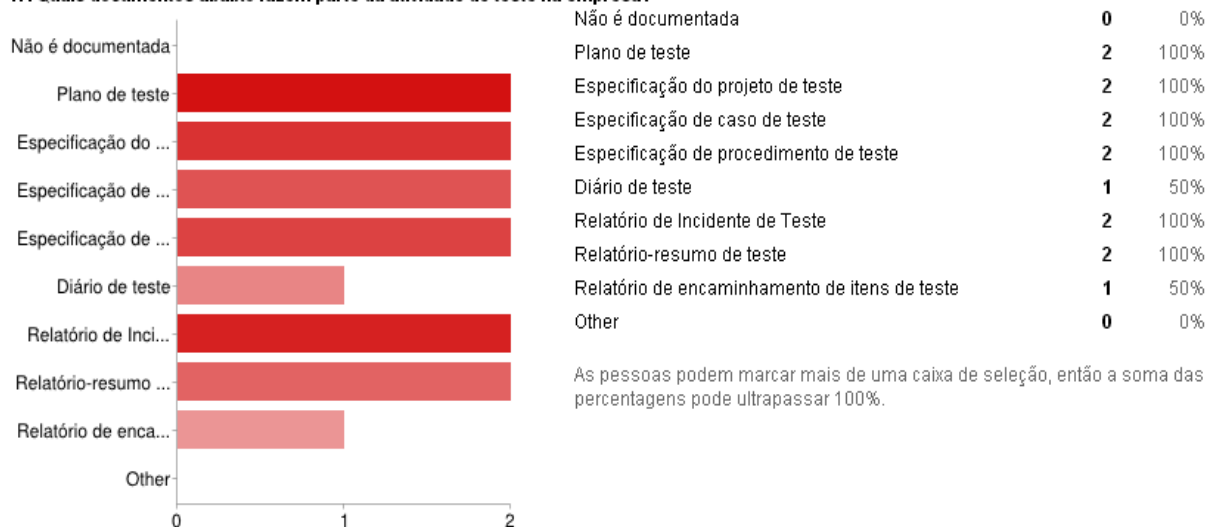


Figura 8: Documentos mais utilizados.



A figura 9 mostra que para as duas empresas houve diminuição dos erros após a implantação da atividade de teste.

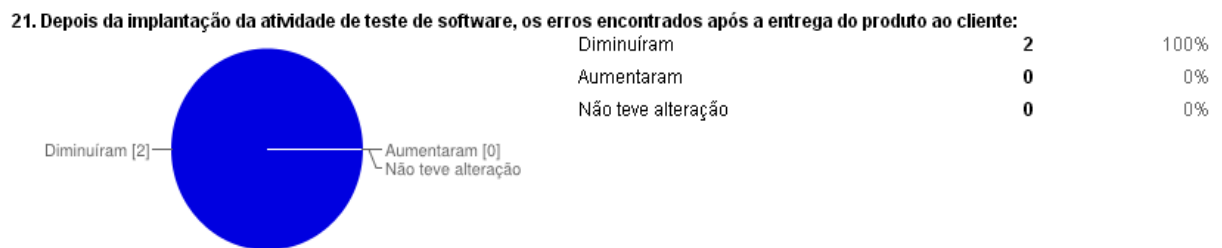


Figura 9: Diminuição dos erros após a implantação da atividade de teste.

### 3.3 ANÁLISE DE TODAS AS EMPRESAS

A partir da análise feita das respostas de todas as empresas que responderam o questionário, pode-se verificar que em relação aos testes de softwares (conforme a tabela 1):

- Apenas uma empresa realiza todos os testes apresentados no questionário, sendo esta uma empresa que presta serviço de teste de software;
- Uma empresa não realiza o teste de caixa preta nem o teste de caixa branca;
- Cinco empresas realizam as duas técnicas de teste (caixa preta e caixa branca);
- Três empresas realizam as cinco fases de teste (unidade, integração, sistema, aceitação e regressão);
- Duas empresas não realizam nenhuma das fases de teste de software;
- O teste de caixa preta é o mais realizado pelas empresas, seguido pelo teste de sistema, desempenho, caixa branca, integração, usabilidade, aceitação, regressão, carga, unidade, configuração, instalação, segurança, recuperação, volume e por último o teste paralelo.

Tabela 1: Testes de Softwares realizados pelas empresas.

Teste Empresa	Caixa preta	Caixa branca	Unidade	Integração	Sistema	Aceitação	Regressão	Carga	Configuração	Desempenho	Instalação	Recuperação	Segurança	Usabilidade	Volume	Paralelo
A		*			*											
B		*		*					*	*						
C	*	*		*	*			*	*	*	*	*	*	*		
D	*					*										
E	*	*														
F	*	*	*	*	*	*					*			*		
G	*															
H	*			*	*		*	*		*						
I	*	*	*	*	*	*	*	*	*	*	*		*	*	*	
J	*				*					*				*	*	*
K			*	*	*	*	*	*		*		*	*	*		
L	*		*				*									
M	*				*	*	*	*	*	*	*	*	*	*	*	*
N	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
TOTAL	11	7	5	7	9	6	6	6	5	8	5	4	5	7	4	3

Outras conclusões foram que, das 14 empresas:

- Duas empresas não possuem dificuldades em implantar o processo de teste de software;
- Duas utilizam todos os documentos apresentados no questionário, sendo uma empresa que desenvolve software e a outra que presta serviço de testes para as empresas (conforme a tabela 2);
- Quatro empresas não documentam a atividade de teste (conforme a tabela 2);
- O documento mais utilizado é o plano de teste, seguido pelo relatório de incidente de teste, especificação de caso de teste, especificação do projeto

de teste, especificação de procedimento de teste, relatório-resumo de teste, relatório de encaminhamento de itens de teste e por último, o diário de teste (conforme a tabela 2).

Tabela 2: Documentação utilizada pelas empresas.

<b>Documento</b>  <b>Empresa</b>	<b>Não é documentada</b>	<b>Plano de Teste</b>	<b>Especificação do Projeto de Teste</b>	<b>Especificação de Caso de Teste</b>	<b>Especificação de Procedimento de Teste</b>	<b>Diário de Teste</b>	<b>Relatório de Incidente de Teste</b>	<b>Relatório - Resumo de Teste</b>	<b>Relatório de Encaminhamento de itens de teste</b>
A	*								
B			*	*			*		*
C		*	*		*	*	*	*	*
D	*								
E		*							
F		*					*		*
G	*								
H		*		*			*		
I		*	*	*	*	*	*	*	*
J	*								
K		*	*	*	*		*	*	
L		*		*					
M		*	*	*	*		*	*	
N		*	*	*	*	*	*	*	*
TOTAL	4	9	6	7	5	3	8	5	5

Pode-se perceber que as empresas que não tem dificuldade em implantar um processo de teste, utilizam vários documentos para os testes. E as empresas que não documentam os testes, possuem alguma dificuldade em implantar o processo de teste. Visto isso, pode-se verificar o quão importante é ter um processo de teste bem documentado.

## CONCLUSÃO

Considerando as leituras realizadas para a elaboração deste trabalho, pode-se constatar que a preocupação com a qualidade de software tem se tornado cada vez maior em função do grande volume de softwares produzidos e das exigências dos usuários que almejam antes de tudo softwares confiáveis, eficientes e de boa qualidade.

O teste de software é uma das atividades que busca contribuir para a melhoria da qualidade de um sistema. O objetivo dos testes é revelar a presença de defeitos no software, para que estes possam ser corrigidos antes de serem entregues ao cliente.

Este trabalho apresentou os diferentes tipos de testes que podem ser executados para se obter um software de qualidade. Cabe cada organização analisar quais são os testes mais adequados para o sistema que está sendo desenvolvido.

A escolha do conjunto de testes depende muitas vezes das restrições de qualidade, do custo, do prazo e dos recursos disponíveis para o desenvolvimento de um sistema.

Identificar e remover tantas falhas quanto possível é um objetivo durante o desenvolvimento, mas encontrar todas as falhas é quase impossível. Os testes não podem durar para sempre. Os softwares devem ser entregues quando atingem um nível adequado de funcionalidade e qualidade (PEZZÈ e YOUNG, 2008).

Pode-se dizer que este trabalho é de extrema importância, pois auxilia no conhecimento da atividade de teste de software e dá uma idéia de como as empresas estão realizando os testes. E o resultado do trabalho foi satisfatório, pois foi alcançado o objetivo proposto.

As dificuldades encontradas para a realização deste trabalho de conclusão de curso foi a de encontrar empresas dispostas e responderem o questionário e o pouco material encontrado sobre testes automatizados e metodologias ágeis.

Como trabalhos futuros sugerem-se fazer visitas as empresas de desenvolvimento de software para verificar como são feitos os testes; implantar a atividade de teste em uma empresa de TI para analisar e comparar os resultados obtidos com a implantação; estudar ferramentas de testes automatizados.

## REFERÊNCIAS

BARTIÉ, Alexandre. **Garantia de Qualidade de Software**: adquirindo maturidade organizacional. Rio de Janeiro: Editora Campus, 2002.

BECK, Kent. Programação Extrema Explicada. Bookman, 1999. In: SOARES, Michel dos Santos. **Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software**. Artigo, Universidade Presidente Antônio Carlos (Unipac), Conselheiro Lafaiete, MG: [2004?]. Disponível em: <<http://www.dcc.ufla.br/infocomp/artigos/v3.2/art02.pdf>>. Acesso em: 30 abr. 2012.

BERNARDO, Paulo Cheque; KON, Fabio. **A Importância dos Testes Automatizados**. Artigo publicado na Engenharia de Software Magazine, 2008. Disponível em: <<http://www.ime.usp.br/~kon/papers/EngSoftMagazine-IntroducaoTestes.pdf>>. Acesso em: 16 fev. 2012.

BISSI, Wilson. **SCRUM - Metodologia de Desenvolvimento Ágil**. Artigo, Centro Universitário de Maringá (CESUMAR), Maringá, PR: 2007. Disponível em: <<http://revista.grupointegrado.br/revista/index.php/campodigital/article/view/312/146>>. Acesso em: 03 maio 2012.

CARVALHO, Eliana Lencioni. **Testes de Software e Automação de Testes**. 8 de março de 2010. Disponível em: <<http://www.testexpert.com.br/?q=node/1702>>. Acesso em: 07 maio 2012.

CORRÊA, Robson Agapito. **Uma Visão Sobre Documentação de Testes (IEEE 829)**. 8 de novembro de 2007. Disponível em: <<http://www.testexpert.com.br/?q=node/366>>. Acesso em: 10 abr. 2012.

CRESPO, Adalberto Nobiato. et al. **Uma Metodologia para Teste de Software no Contexto da Melhoria de Processo**. Artigo, São Paulo: [2004?]. Disponível em: <<http://www.testexpert.com.br/files/Uma%20Metodologia%20para%20Teste%20de%20Software%20no%20Contexto%20da%20Melhoria%20de%20Processo.PDF>>. Acesso em: 9 fev. 2012.

DIAS NETO, Arilo Claudio. **Uma Infra-Estrutura Computacional para Apoiar o Planejamento e Controle de Testes de Software**. Dissertação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ: abril de 2006. Disponível em <[https://docs.google.com/viewer?a=v&q=cache:957wus3l5Y4J:icomp.ufam.edu.br/arilo/index.php%3Foption%3Dcom\\_docman%26task%3Ddoc\\_download%26gid%3D21%26Itemid%3D44%26lang%3Dbr+&hl=pt-BR&gl=br&pid=bl&srcid=ADGEESgCElhLE-ZTkD5k7sGSd\\_rdN-zpX-SdUCBI1aHDwxllYblXGAncd0AuGWoCyhP3Zu\\_hpQe5buLkSB0Fsm8TPDQ8RrHpOdQRCM8JSAHjIF07XQwnWxiAlbws4jg0AZGi\\_mt2mT1C&sig=AHIEtbQ0M4wi-0qxojLMQgFxl5RRmvCrzg](https://docs.google.com/viewer?a=v&q=cache:957wus3l5Y4J:icomp.ufam.edu.br/arilo/index.php%3Foption%3Dcom_docman%26task%3Ddoc_download%26gid%3D21%26Itemid%3D44%26lang%3Dbr+&hl=pt-BR&gl=br&pid=bl&srcid=ADGEESgCElhLE-ZTkD5k7sGSd_rdN-zpX-SdUCBI1aHDwxllYblXGAncd0AuGWoCyhP3Zu_hpQe5buLkSB0Fsm8TPDQ8RrHpOdQRCM8JSAHjIF07XQwnWxiAlbws4jg0AZGi_mt2mT1C&sig=AHIEtbQ0M4wi-0qxojLMQgFxl5RRmvCrzg)>. Acesso em: 20 fev. 2012.

FURLAN, Fabiane Pifer. **Visualização de Informação como Apoio ao Planejamento do Teste de Software**. Dissertação de Mestrado, Universidade Metodista de Piracicaba, Piracicaba, SP: 2009. Disponível em: <<https://www.unimep.br/phpg/bibdig/pdfs/2006/PIICEDIGLYDU.pdf>>. Acesso em: 14 fev. 2012.

INTHURN, Cândida. Qualidade & Teste de Software. Florianópolis: Visual Books, 2001. In: ZIMMERMANN, Ana Paula. **Ferramenta para Automação de Testes de Caixa Preta**. Trabalho de Conclusão de Curso, Universidade do Vale do Itajaí, Itajaí, SC: julho de 2006. Disponível em: <<http://siaibib01.univali.br/pdf/Ana%20Paula%20Zimmermann.pdf>>. Acesso em: 13 fev. 2012.

KOLM, Everton Luiz. **Sistema para Gerenciamento de Testes Funcionais de Software**. Trabalho de estágio supervisionado, Universidade Regional de Blumenau, Blumenau, SC: novembro de 2001. Disponível em: <<http://campeche.inf.furb.br/tccs/2001-II/2001-2evertonluizkolmvf.pdf>>. Acesso em: 09 fev. 2012.

LEAL, Igor Gonçalves. **Requisitos de Metodologias de Teste de Software para Processos Ágeis**. Artigo, Universidade Federal de Minas Gerais, Belo Horizonte, MG: [2009?]. Disponível em: <<http://homepages.dcc.ufmg.br/~rodolfo/dcc823-1-09/Entrega2Pos/igor2.pdf>>. Acesso em: 26 abr. 2012.

MACHADO, Alex de Magalhães. **A Implantação de Testes Automatizados de Aplicações Web num Ambiente Hostil**. Artigo, Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC: [2009?]. Disponível em: <[http://www.inf.ufsc.br/~alexmag/artigo\\_estudo\\_de\\_caso.pdf](http://www.inf.ufsc.br/~alexmag/artigo_estudo_de_caso.pdf)>. Acesso em: 26 abr. 2012.

MALDONADO, José Carlos. et al. **Aspectos Teóricos e Empíricos de Teste de Cobertura de Software**. Artigo: [1998?]. Disponível em: <<http://www.labes.icmc.usp.br/site/sites/default/files/NotaDidatica31.pdf>>. Acesso em: 15 fev. 2012.

MATSUDO, Douglas Hideki. **Sprints - Planejamento, execução e controle**. 10 de agosto de 2009. Disponível em: <<http://www.matsudo.com.br/scrum/11-sprints-planejamento-execucao-e-controle>>. Acesso em: 03 maio 2012.

MEIRELLES, Paulo Roberto Miranda. **Teste Integrado de Software e Hardware: Reusando Casos de Teste de Software em Teste de Microprocessadores**. Dissertação (mestrado), Universidade Federal do Rio Grande do Sul, Porto Alegre, RS: junho de 2008. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/25520/000751158.pdf?sequence=1>>. Acesso em: 13 fev. 2012.

MERLIN, José Reinaldo. **Análise e Projeto de Sistemas III**. Universidade Estadual do Norte Paraná, Bandeirantes: 2006. Disponível em:

<<http://ramses.ffalm.br/falm/info/professores/merlin/An%E1lise%20e%20Projeto%20Sistemas%203/APSIII.pdf>>. Acesso em: 16 fev. 2012.

MYERS, Glenford J. The Art of Software Testing. 1979. In: BARTIÉ, Alexandre. **Garantia de Qualidade de Software**: adquirindo maturidade organizacional. Rio de Janeiro: Editora Campus, 2002.

MYERS, Glenford J. The Art of Software Testing. 1979. In: JORGE, Rodrigo Funabashi. **Teste de Softwares Embarcados**. Universidade Federal de Mato Grosso do Sul (UFMS), Campo Grande, MS: [2009?]. Disponível em: <<http://www.ic.unicamp.br/~rodolfo/Cursos/mo401/2s2009/t2/099173-A.pdf>>. Acesso em: 13 fev. 2012.

MYERS, Glenford J. The Art of Software Testing. 1979. In: VILLAS BOAS, André Luiz de Castro. **Gestão de Configuração para Teste de Software**. Dissertação, Universidade Estadual de Campinas, Campinas, SP: junho de 2003. Disponível em: <<http://www.bibliotecadigital.unicamp.br/document/?code=vtls000308144>>. Acesso em: 15 fev. 2012.

NEVES, Luciane. **Projeto de uma Metodologia de Testes para Sistemas de Informação Corporativa**. Monografia, Curitiba, PR: 1999. Disponível em: <[https://docs.google.com/viewer?a=v&q=cache:jeOzfNIDmXUJ:celepar7cta.pr.gov.br/portfolio.nsf/b239398b4e7d02ec03256d9c003fdb8/68a2bab0e99b141d03256c2f0073e31e/%24FILE/versao5.doc+&hl=pt-BR&gl=br&pid=bl&srcid=ADGEESjOvxrnzyA9P6zYUS8liqp5AJNoBmV8F-MGhxmuoCj0AukFQJvwbaO6kF71t3g4Pe5jC4fjczOyOxdF7TQbD--5tkP8fVbnh8FKSbO5l2nfoByCIQLUGxYCn61RCtzcpeZ1YIBn&sig=AHIEtbQ\\_koMXsKNv4ZqKV8emq08PlxnYDQ](https://docs.google.com/viewer?a=v&q=cache:jeOzfNIDmXUJ:celepar7cta.pr.gov.br/portfolio.nsf/b239398b4e7d02ec03256d9c003fdb8/68a2bab0e99b141d03256c2f0073e31e/%24FILE/versao5.doc+&hl=pt-BR&gl=br&pid=bl&srcid=ADGEESjOvxrnzyA9P6zYUS8liqp5AJNoBmV8F-MGhxmuoCj0AukFQJvwbaO6kF71t3g4Pe5jC4fjczOyOxdF7TQbD--5tkP8fVbnh8FKSbO5l2nfoByCIQLUGxYCn61RCtzcpeZ1YIBn&sig=AHIEtbQ_koMXsKNv4ZqKV8emq08PlxnYDQ)>. Acesso em: 09 fev. 2012.

PAULA FILHO, Wilson de Pádua. **Engenharia de Software**: Fundamentos, Métodos e Padrões. 2ª edição. Rio de Janeiro: LTC, 2005. 602 p.

PAULA FILHO, Wilson de Pádua. **Manual do Engenheiro de software**. Módulo Técnico. janeiro de 2000. Disponível em: <<http://pt.scribd.com/doc/7570545/MANualEngSW>>. Acesso em: 23 fev. 2012.

PEZZÈ, Mauro; YOUNG, Michal. **Teste e Análise de Software**: processos, princípios e técnicas. Tradução Bernardo Copstein, Flavio Moreira de Oliveira, Porto Alegre: Bookman, 2008. 512p.

PFLEEGER, Shari Lawrence. Engenharia de Software – Teoria e Prática. 2ª edição. São Paulo: Prentice Hall, 2004. In: FURLAN, Fabiane Pifer. **Visualização de Informação como Apoio ao Planejamento do Teste de Software**. Dissertação de Mestrado, Universidade Metodista de Piracicaba, Piracicaba, SP: 2009. Disponível em: <<https://www.unimep.br/phpg/bibdig/pdfs/2006/PIICEDIGLYDU.pdf>>. Acesso em: 18 fev. 2012.

PRESSMAN, Roger S. **Engenharia de Software**. São Paulo: Makron Books, 1995. 1056p.



PRESSMAN, Roger S. Engenharia de Software; tradução Rosângela Dellosso Penteado. 6ª ed. São Paulo: McGraw-Hill, 2006. In: PINTO JÚNIOR, José Gonçalves. **O uso da Metodologia XP no Desenvolvimento de Software e os Impactos da Gestão de Riscos**. Trabalho de Conclusão de Curso, Centro Universitário da Fundação de Ensino Octávio Bastos, São João de Boa Vista, SP: 2009. Disponível em: <<http://camilolopes.files.wordpress.com/2009/12/monografia-jose-goncalves-p-junior.pdf>>. Acesso em: 20 jan. 2012.

RIOS, Emerson; FILHO, Trayahú Moreira. **Teste de Software**. Rio de Janeiro: Alta Books, 2003.

RIOS, Emerson. **Documentação de Teste**. Dissecando a norma ou padrão IEEE 829. Instituto de Teste de Software: Setembro de 2007. Disponível em: <<http://www.docstoc.com/docs/115787396/Documenta%EF%BF%BD%EF%BF%BD-o-de-Teste>>. Acesso em: 10 abr. 2012.

SANCHEZ, Ivan. **Scrum em 2 minutos**. 7 de fevereiro de 2007. Disponível em: <<http://dojofloripa.wordpress.com/2007/02/07/scrum-em-2-minutos/>>. Acesso em: 03 maio 2012.

SCHUSTER, Vanessa. **O Uso da Ferramenta de Teste de Software**. Universidade Regional do Noroeste do Estado do Rio Grande do Sul, Ijuí, RS: 2009.

SILVA, Maykel Melo da. **Desenvolvimento de uma Ferramenta de Testes para um Sistema Integrado em Software e Hardware**. Monografia, Universidade Federal de Lavras, Lavras, MG: 2005. Disponível em: <[http://www.bcc.ufla.br/monografias/2005/Desenvolvimento\\_de\\_uma\\_ferramenta\\_de\\_testes\\_para\\_um\\_sistema\\_integrado\\_em\\_software\\_e\\_hardware.pdf](http://www.bcc.ufla.br/monografias/2005/Desenvolvimento_de_uma_ferramenta_de_testes_para_um_sistema_integrado_em_software_e_hardware.pdf)>. Acesso em: 2 fev. 2012.

SOARES, Michel dos Santos. **Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software**. Artigo, Universidade Presidente Antônio Carlos (Unipac), Conselheiro Lafaiete, MG: [2004?]. Disponível em: <<http://www.dcc.ufla.br/infocomp/artigos/v3.2/art02.pdf>>. Acesso em: 30 abr. 2012.

SOMMERVILLE, Ian. **Engenharia de Software**. Tradução André Maurício de Andrade Ribeiro. 6ª edição. São Paulo: Pearson Addison Wesley, 2003. 592 p.

SOMMERVILLE, Ian. Engenharia de Software. 8ª edição, Pearson – Prentice Hall, 2007. In: JORGE, Rodrigo Funabashi. **Teste de Softwares Embarcados**. Universidade Federal de Mato Grosso do Sul (UFMS), Campo Grande, MS: [2009?]. Disponível em: <<http://www.ic.unicamp.br/~rodolfo/Cursos/mo401/2s2009/t2/099173-A.pdf>>. Acesso em: 13 fev. 2012.

SOUZA, Luciano Malaquias de. **Método Ágil XP (Extreme Programming)**. Revista Eletrônica da FIA, Vol. III, Jul-Dez/2007. Disponível em:

<[http://intranet.fia.edu.br/acesso\\_site/fia/academos/revista3/6.pdf](http://intranet.fia.edu.br/acesso_site/fia/academos/revista3/6.pdf)>. Acesso em: 10 maio 2012.

TOMELIN, Marcio. **Testes de Software a partir da Ferramenta Visual Test**. Trabalho de Conclusão de Curso, Universidade Regional de Blumenau, Blumenau, SC: junho de 2001. Disponível em: <<http://campeche.inf.furb.br/tccs/2001-I/2001-1marciotomelinvf.pdf>>. Acesso em: 20 jan. 2012.

ZANOTTO, Tiago Elias. **Definição de Requisitos para o Módulo de Inspeções da Ferramenta Case Film**. Projeto de Diplomação, Universidade de Caxias do Sul, Caxias do Sul, RS: janeiro de 2002. Disponível em: <<http://www.ucs.br/portais/projetofilm/documentos/9300/download/>>. Acesso em: 18 fev. 2012.

WK Sistemas. **Testes Automatizados: preocupação com o cliente**. 27 de novembro de 2003. Disponível em: <[http://www.wk.com.br/Noticias/testes-automatizados-preocupacao-com-o-cliente\[919\].aspx](http://www.wk.com.br/Noticias/testes-automatizados-preocupacao-com-o-cliente[919].aspx)>. Acesso em: 27 fev. 2012.

## APÊNDICE A – Questionário

### Questionário Teste de Software

Este questionário auxiliará na pesquisa do Trabalho de Conclusão de Curso. O objetivo principal é verificar os métodos de testes existentes nas empresas de desenvolvimento de software na área de TI.

As informações fornecidas serão disponibilizadas somente para este trabalho.

Aguardo a sua contribuição.

Obrigada!

---

**\*Obrigatório**

1. Nome da empresa? Local (cidade)?\*

2. A empresa atua no mercado há quantos anos?\*

- ☐ Menos de 1 ano
- ☐ De 1 a 3 anos
- ☐ De 3 a 5 anos
- ☐ De 5 a 7 anos
- ☐ De 7 a 10 anos
- ☐ Mais de 10 anos

3. A empresa possui quantos funcionários?\*

- ☐ Menos de 10
- ☐ De 10 a 20
- ☐ De 20 a 30
- ☐ De 30 a 40
- ☐ De 40 a 50
- ☐ Mais de 50

4. Qual o público alvo da empresa?\*

- ☐ Empresas locais
- ☐ Empresas regionais
- ☐ Empresas de outros estados
- ☐ Empresas de outros países
- ☐ Outro:

5. Há quanto tempo a atividade de teste de software foi implantada na empresa?\*

*Caso responda "Não faço testes de software" pule para a questão 22.*

- ☐ Menos de 1 ano
- ☐ De 1 a 3 anos
- ☐ De 3 a 5 anos
- ☐ De 5 a 7 anos
- ☐ De 7 a 10 anos
- ☐ Mais de 10 anos
- ☐ Não faço testes de software (*Pule para a questão 22*)

6. Quais testes de softwares são feitos?

- ☐ Teste funcional (caixa preta)
- ☐ Teste estrutural (caixa branca)
- ☐ Teste de unidade
- ☐ Teste de integração
- ☐ Teste de sistema
- ☐ Teste de aceitação
- ☐ Teste de regreção
- ☐ Teste de carga
- ☐ Teste de configuração
- ☐ Teste de desempenho/performance
- ☐ Teste de instalação
- ☐ Teste de recuperação
- ☐ Teste de segurança
- ☐ Teste de usabilidade
- ☐ Teste de volume
- ☐ Teste paralelo
- ☐ Outro:

7. Caso utilize teste de caixa preta marque os critérios utilizados?

- ☐ Particionamento de equivalência
- ☐ Análise de valor limite
- ☐ Grafo de causa e efeito
- ☐ Desconheço os critérios indicados nesta questão

☐ Outro:

8. Caso utilize teste de caixa branca marque os critérios utilizados?

- ☐ Critérios baseados em fluxo de controle
- ☐ Critérios baseados em fluxo de dados
- ☐ Critérios baseados na complexidade
- ☐ Desconheço os critérios indicados nesta questão
- ☐ Outro:

9. Quem são os responsáveis por realizar os testes de software?

- ☐ Equipe terceirizada especializada em testes
- ☐ Equipe de testes da empresa
- ☐ Os próprios desenvolvedores do sistema
- ☐ Os clientes
- ☐ Os usuários
- ☐ Outro:

10. Quando é iniciada a execução dos testes?

- ☒ Na especificação de requisitos
- ☒ Na elaboração do projeto de software
- ☒ Na fase de codificação
- ☒ Na fase de implementação do software
- ☒ Outro:

11. Onde está concentrada a maior incidência dos erros?

- ☐ Nas fases iniciais do processo de desenvolvimento
- ☐ No meio do processo de desenvolvimento
- ☐ Nas fases finais do processo de desenvolvimento
- ☐ Outro:

12. Quais são os erros mais comuns de serem encontrados nos sistemas desenvolvidos na empresa?

- ☐ Erros de interpretação de análise dos requisitos
- ☐ Erros de código
- ☐ Erros de funcionalidade
- ☐ Erros de digitação
- ☐ Erros de interface
- ☐ Outro:

13. Quais as dificuldades encontradas na realização dos testes de software?

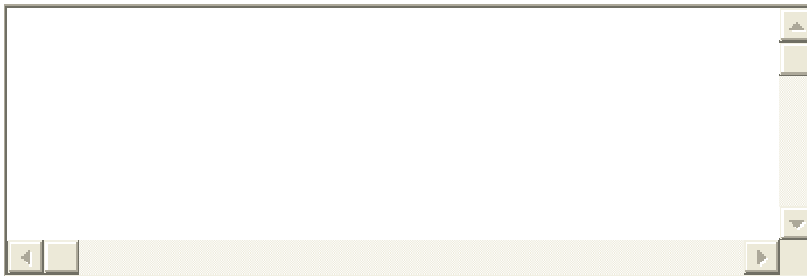
- ☐ O teste é um processo caro
- ☐ A atividade de teste é limitada por restrições de cronograma
- ☐ Há a falta de profissionais especializados na área de teste
- ☐ Existem dificuldades em implantar um processo de teste
- ☐ Há o desconhecimento de um procedimento de teste adequado
- ☐ Há o desconhecimento de técnicas de teste adequadas
- ☐ Há o desconhecimento sobre como planejar a atividade de teste
- ☐ Outro:

14. Como os testes são realizados?

*Caso responda "Manualmente" pule para a questão 17.*

- ☒ Manualmente (*Pule para a questão 17*)
- ☐ Através da automatização
- ☐ Ambos

15. Se automatizado, quais ferramentas são utilizadas?

A large, empty rectangular text box with a thin black border, intended for the user to list the tools used for automated testing. It has a small scroll bar on the right side.

16. Na sua opinião, quais as vantagens dos testes automatizados em relação aos testes feitos de forma manual?

- ☐ Eliminação do trabalho repetitivo de inserção de dados e observação dos resultados
- ☐ Melhoria da qualidade do processo e produto de software
- ☐ Maior confiança quanto à liberação do produto, a cada nova versão
- ☐ Reusabilidade dos scripts de testes, com grande facilidade para alteração de informações
- ☐ Diminuição do tempo gasto com a execução dos testes
- ☐ Aumento da produtividade e diminuição do custo destinado às atividades de teste
- ☐ Aumento da cobertura (abrangência) dos testes, através do número crescente de implementações de casos de testes
- ☐ Outro:



17. Quais documentos abaixo fazem parte da atividade de teste na empresa?

- ☐ Não é documentada
- ☐ Plano de teste
- ☐ Especificação do projeto de teste
- ☐ Especificação de caso de teste
- ☐ Especificação de procedimento de teste
- ☐ Diário de teste
- ☐ Relatório de Incidente de Teste
- ☐ Relatório-resumo de teste
- ☐ Relatório de encaminhamento de itens de teste
- ☐ Outro:

18. Qual percentual de tempo do projeto de desenvolvimento do software é consumido com a atividade de testes?

- ☐ Até 5%
- ☐ De 5% a 10%
- ☐ De 10% a 20%
- ☐ De 20% a 30%
- ☐ De 30% a 40%
- ☐ De 40% a 50%
- ☐ Acima de 50%

19. Qual o valor efetivo dos testes de software realizados na empresa?

- ☐ Apresentam pequena contribuição

- ☐ Apresentam média contribuição
- ☐ Apresentam grande contribuição

20. Com relação aos custos, após a implantação da atividade de teste observou-se que:

- ☐ Diminuíram
- ☐ Aumentaram
- ☐ Não teve alteração

21. Depois da implantação da atividade de teste de software, os erros encontrados após a entrega do produto ao cliente:

- ☐ Diminuíram
- ☐ Aumentaram
- ☐ Não teve alteração

22. Porque a empresa não realiza testes de softwares?

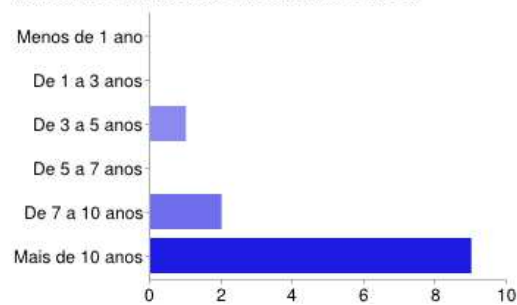
*Só responda caso a empresa não realize teste de software*

## APÊNDICE B – Respostas do Questionário

Respostas do questionário apresentado no APÊNDICE A.

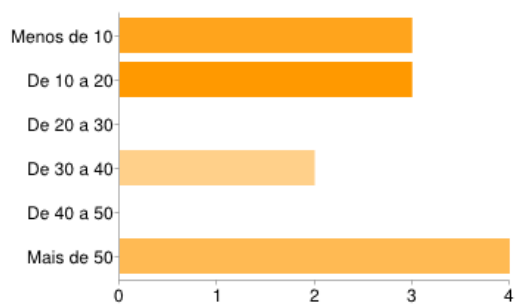
a) Respostas das 12 empresas que desenvolvem softwares:

### 2. A empresa atua no mercado há quantos anos?



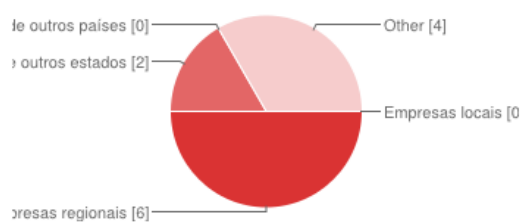
Menos de 1 ano	0	0%
De 1 a 3 anos	0	0%
De 3 a 5 anos	1	8%
De 5 a 7 anos	0	0%
De 7 a 10 anos	2	17%
Mais de 10 anos	9	75%

### 3. A empresa possui quantos funcionários?



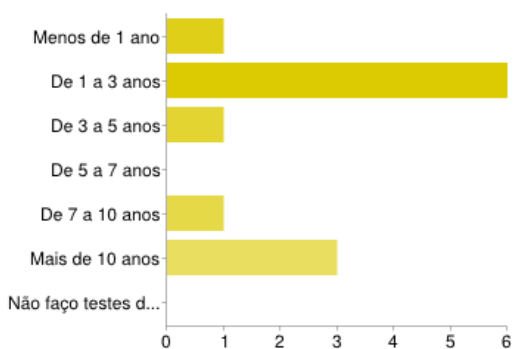
Menos de 10	3	25%
De 10 a 20	3	25%
De 20 a 30	0	0%
De 30 a 40	2	17%
De 40 a 50	0	0%
Mais de 50	4	33%

### 4. Qual o público alvo da empresa?



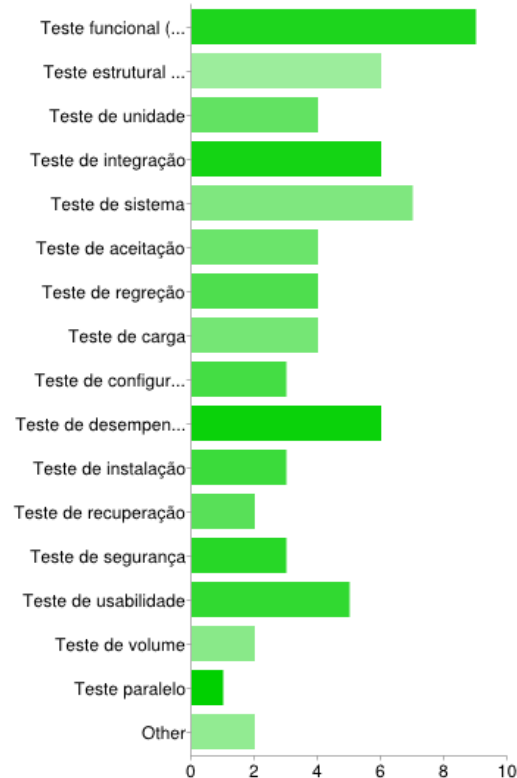
Empresas locais	0	0%
Empresas regionais	6	50%
Empresas de outros estados	2	17%
Empresas de outros países	0	0%
Other	4	33%

### 5. Há quanto tempo a atividade de teste de software foi implantada na empresa?



Menos de 1 ano	1	8%
De 1 a 3 anos	6	50%
De 3 a 5 anos	1	8%
De 5 a 7 anos	0	0%
De 7 a 10 anos	1	8%
Mais de 10 anos	3	25%
Não faço testes de software (Pule para a questão 22)	0	0%

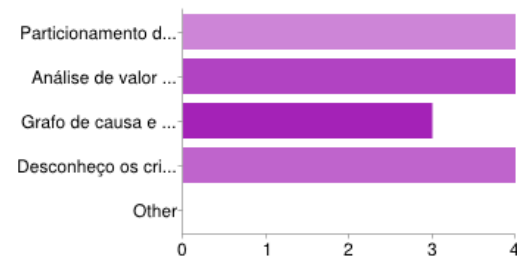
#### 6. Quais testes de softwares são feitos?



Teste funcional (caixa preta)	9	75%
Teste estrutural (caixa branca)	6	50%
Teste de unidade	4	33%
Teste de integração	6	50%
Teste de sistema	7	58%
Teste de aceitação	4	33%
Teste de regreção	4	33%
Teste de carga	4	33%
Teste de configuração	3	25%
Teste de desempenho/performance	6	50%
Teste de instalação	3	25%
Teste de recuperação	2	17%
Teste de segurança	3	25%
Teste de usabilidade	5	42%
Teste de volume	2	17%
Teste paralelo	1	8%
Other	2	17%

As pessoas podem marcar mais de uma caixa de seleção, então a soma das percentagens pode ultrapassar 100%.

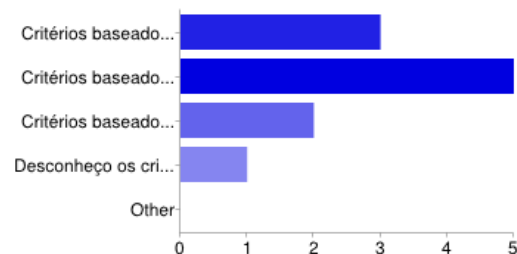
#### 7. Caso utilize teste de caixa preta marque os critérios utilizados?



Particionamento de equivalência	4	44%
Análise de valor limite	4	44%
Grafo de causa e efeito	3	33%
Desconheço os critérios indicados nesta questão	4	44%
Other	0	0%

As pessoas podem marcar mais de uma caixa de seleção, então a soma das percentagens pode ultrapassar 100%.

#### 8. Caso utilize teste de caixa branca marque os critérios utilizados?



Critérios baseados em fluxo de controle	3	50%
Critérios baseados em fluxo de dados	5	83%
Critérios baseados na complexidade	2	33%
Desconheço os critérios indicados nesta questão	1	17%
Other	0	0%

As pessoas podem marcar mais de uma caixa de seleção, então a soma das percentagens pode ultrapassar 100%.

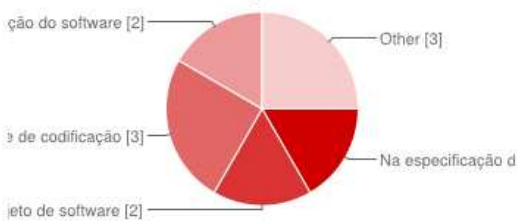
**9. Quem são os responsáveis por realizar os testes de software?**



Equipe terceirizada especializada em testes	0	0%
Equipe de testes da empresa	10	83%
Os próprios desenvolvedores do sistema	8	67%
Os clientes	2	17%
Os usuários	1	8%
Other	0	0%

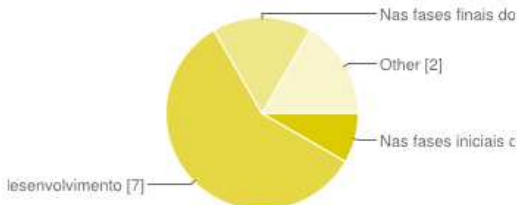
As pessoas podem marcar mais de uma caixa de seleção, então a soma das percentagens pode ultrapassar 100%.

**10. Quando é iniciada a execução dos testes?**



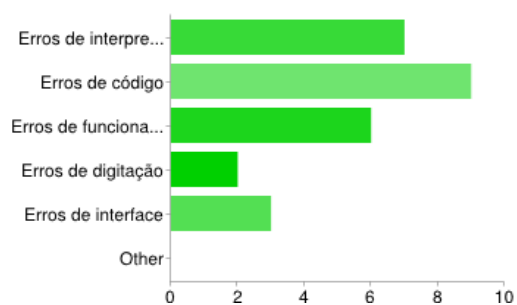
Na especificação de requisitos	2	17%
Na elaboração do projeto de software	2	17%
Na fase de codificação	3	25%
Na fase de implementação do software	2	17%
Other	3	25%

**11. Onde está concentrada a maior incidência dos erros?**



Nas fases iniciais do processo de desenvolvimento	1	8%
No meio do processo de desenvolvimento	7	58%
Nas fases finais do processo de desenvolvimento	2	17%
Other	2	17%

**12. Quais são os erros mais comuns de serem encontrados nos sistemas desenvolvidos na empresa?**



Erros de interpretação de análise dos requisitos	7	58%
Erros de código	9	75%
Erros de funcionalidade	6	50%
Erros de digitação	2	17%
Erros de interface	3	25%
Other	0	0%

As pessoas podem marcar mais de uma caixa de seleção, então a soma das percentagens pode ultrapassar 100%.

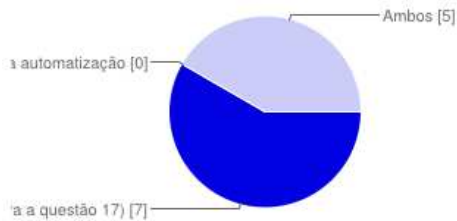
**13. Quais as dificuldades encontradas na realização dos testes de software?**



O teste é um processo caro	4	33%
A atividade de teste é limitada por restrições de cronograma	5	42%
Há a falta de profissionais especializados na área de teste	5	42%
Existem dificuldades em implantar um processo de teste	4	33%
Há o desconhecimento de um procedimento de teste adequado	2	17%
Há o desconhecimento de técnicas de teste adequadas	2	17%
Há o desconhecimento sobre como planejar a atividade de teste	0	0%
Other	2	17%

As pessoas podem marcar mais de uma caixa de seleção, então a soma das percentagens pode ultrapassar 100%.

#### 14. Como os testes são realizados?

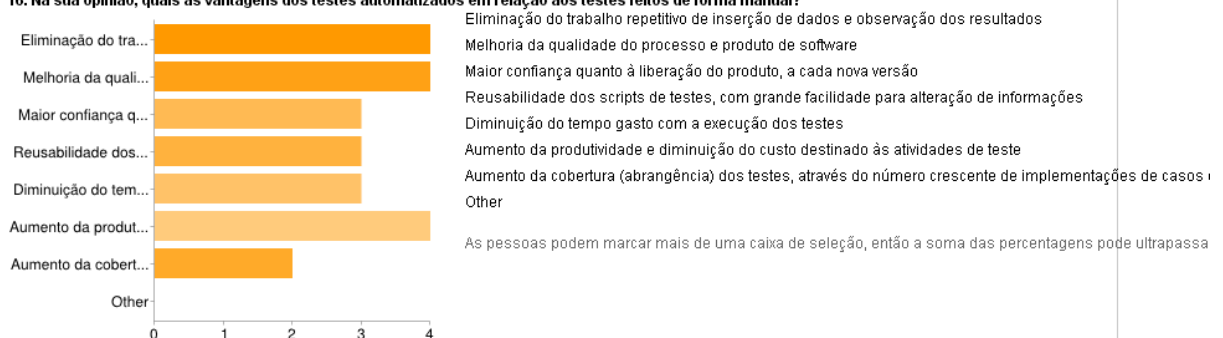


Manualmente (Pule para a questão 17)	7	58%
Através da automização	0	0%
Ambos	5	42%

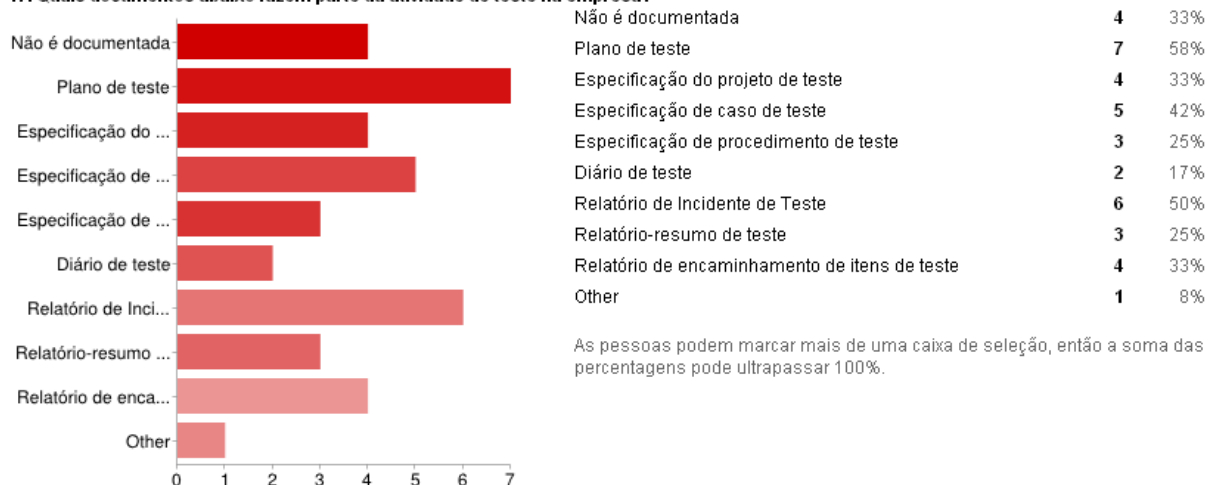
#### 15. Se automatizado, quais ferramentas são utilizadas?



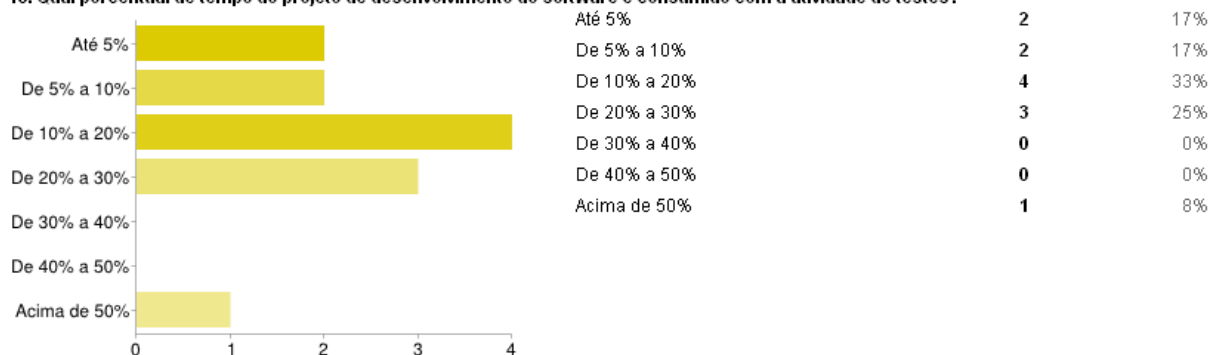
#### 16. Na sua opinião, quais as vantagens dos testes automatizados em relação aos testes feitos de forma manual?



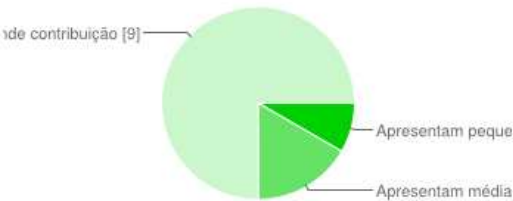
#### 17. Quais documentos abaixo fazem parte da atividade de teste na empresa?



#### 18. Qual percentual de tempo do projeto de desenvolvimento do software é consumido com a atividade de testes?

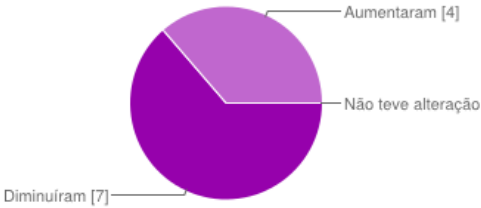


19. Qual o valor efetivo dos testes de software realizados na empresa?



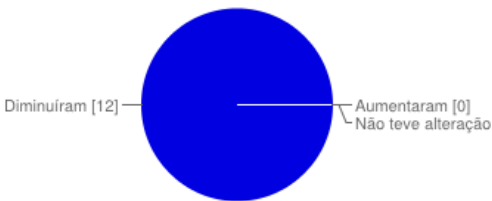
Apresentam pequena contribuição	1	8%
Apresentam média contribuição	2	17%
Apresentam grande contribuição	9	75%

20. Com relação aos custos, após a implantação da atividade de teste observou-se que:



Diminuíram	7	58%
Aumentaram	4	33%
Não teve alteração	0	0%

21. Depois da implantação da atividade de teste de software, os erros encontrados após a entrega do produto ao cliente:



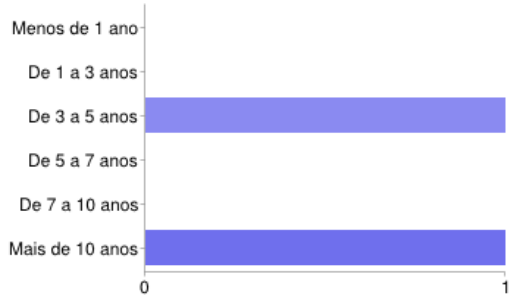
Diminuíram	12	100%
Aumentaram	0	0%
Não teve alteração	0	0%

22. Porque a empresa não realiza testes de softwares?



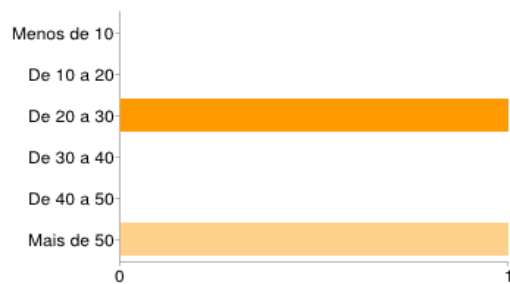
b) Respostas das 2 empresas que prestam serviço de teste de software para as empresas:

2. A empresa atua no mercado há quantos anos?



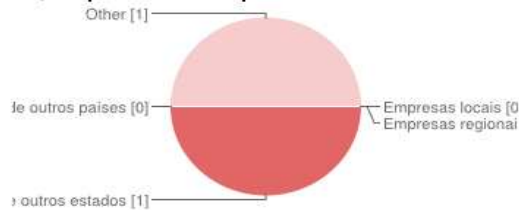
Menos de 1 ano	0	0%
De 1 a 3 anos	0	0%
De 3 a 5 anos	1	50%
De 5 a 7 anos	0	0%
De 7 a 10 anos	0	0%
Mais de 10 anos	1	50%

### 3. A empresa possui quantos funcionários?



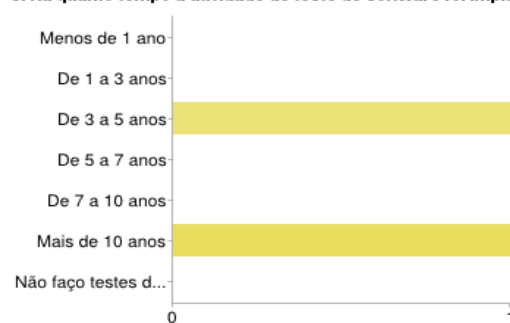
Menos de 10	0	0%
De 10 a 20	0	0%
De 20 a 30	1	50%
De 30 a 40	0	0%
De 40 a 50	0	0%
Mais de 50	1	50%

### 4. Qual o público alvo da empresa?



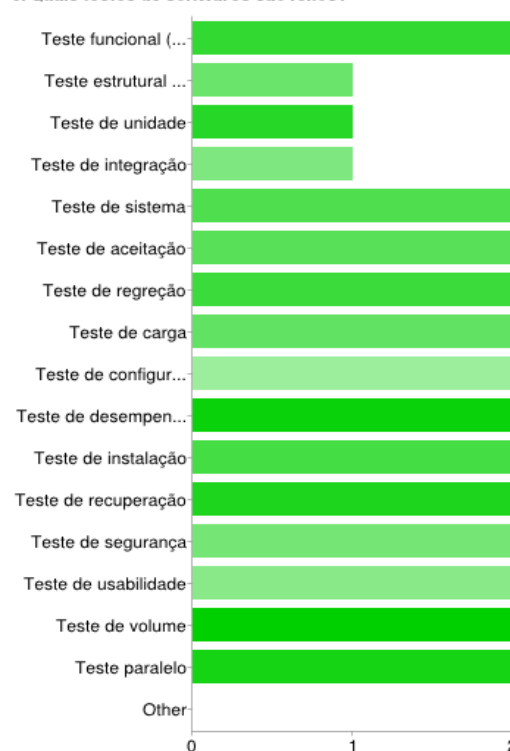
Empresas locais	0	0%
Empresas regionais	0	0%
Empresas de outros estados	1	50%
Empresas de outros países	0	0%
Other	1	50%

### 5. Há quanto tempo a atividade de teste de software foi implantada na empresa?



Menos de 1 ano	0	0%
De 1 a 3 anos	0	0%
De 3 a 5 anos	1	50%
De 5 a 7 anos	0	0%
De 7 a 10 anos	0	0%
Mais de 10 anos	1	50%
Não faço testes de software (Pule para a questão 22)	0	0%

### 6. Quais testes de softwares são feitos?

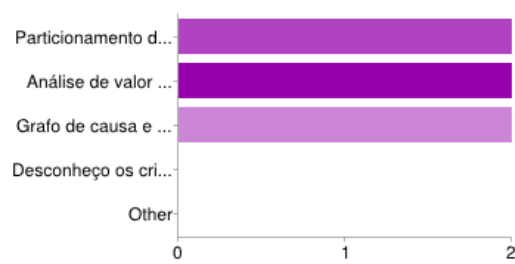


Teste funcional (caixa preta)	2	100%
Teste estrutural (caixa branca)	1	50%
Teste de unidade	1	50%
Teste de integração	1	50%
Teste de sistema	2	100%
Teste de aceitação	2	100%
Teste de regreção	2	100%
Teste de carga	2	100%
Teste de configuração	2	100%
Teste de desempenho/performance	2	100%
Teste de instalação	2	100%
Teste de recuperação	2	100%
Teste de segurança	2	100%
Teste de usabilidade	2	100%
Teste de volume	2	100%
Teste paralelo	2	100%
Other	0	0%

As pessoas podem marcar mais de uma caixa de seleção, então a soma das percentagens pode ultrapassar 100%.



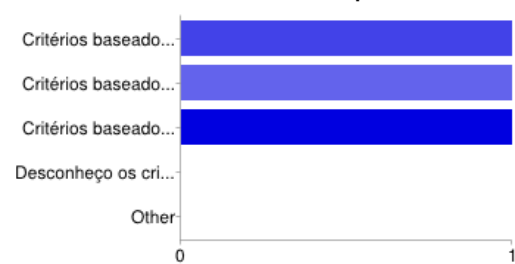
**7. Caso utilize teste de caixa preta marque os critérios utilizados?**



Particionamento de equivalência	2	100%
Análise de valor limite	2	100%
Grafo de causa e efeito	2	100%
Desconheço os critérios indicados nesta questão	0	0%
Other	0	0%

As pessoas podem marcar mais de uma caixa de seleção, então a soma das percentagens pode ultrapassar 100%.

**8. Caso utilize teste de caixa branca marque os critérios utilizados?**



Critérios baseados em fluxo de controle	1	100%
Critérios baseados em fluxo de dados	1	100%
Critérios baseados na complexidade	1	100%
Desconheço os critérios indicados nesta questão	0	0%
Other	0	0%

As pessoas podem marcar mais de uma caixa de seleção, então a soma das percentagens pode ultrapassar 100%.

**9. Quem são os responsáveis por realizar os testes de software?**



Equipe terceirizada especializada em testes	0	0%
Equipe de testes da empresa	2	100%
Os próprios desenvolvedores do sistema	0	0%
Os clientes	0	0%
Os usuários	0	0%
Other	0	0%

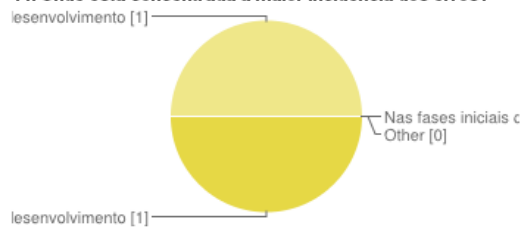
As pessoas podem marcar mais de uma caixa de seleção, então a soma das percentagens pode ultrapassar 100%.

**10. Quando é iniciada a execução dos testes?**



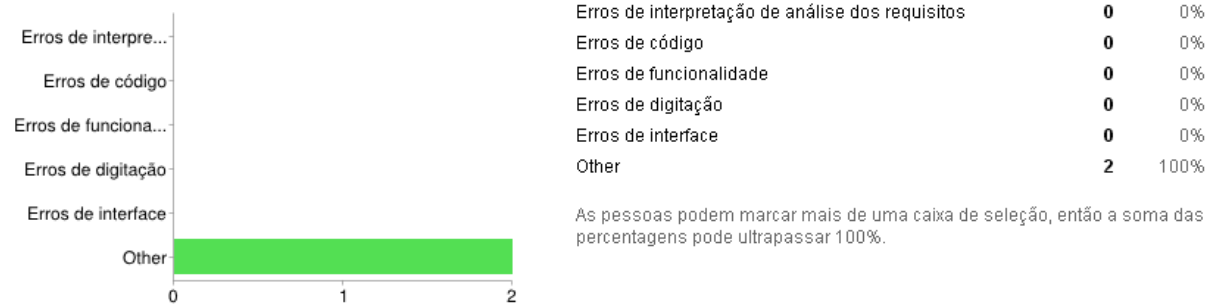
Na especificação de requisitos	1	50%
Na elaboração do projeto de software	0	0%
Na fase de codificação	0	0%
Na fase de implementação do software	1	50%
Other	0	0%

**11. Onde está concentrada a maior incidência dos erros?**

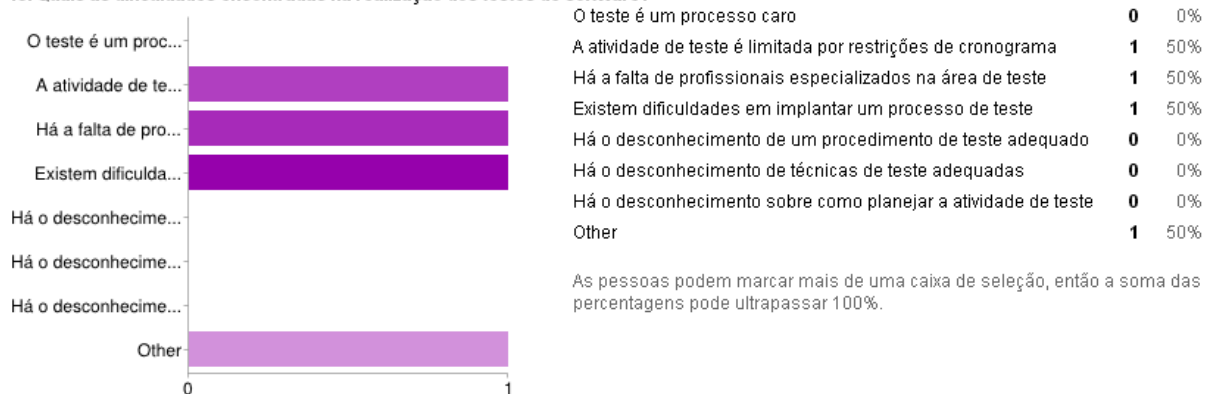


Nas fases iniciais do processo de desenvolvimento	1	50%
No meio do processo de desenvolvimento	0	0%
Nas fases finais do processo de desenvolvimento	1	50%
Other	0	0%

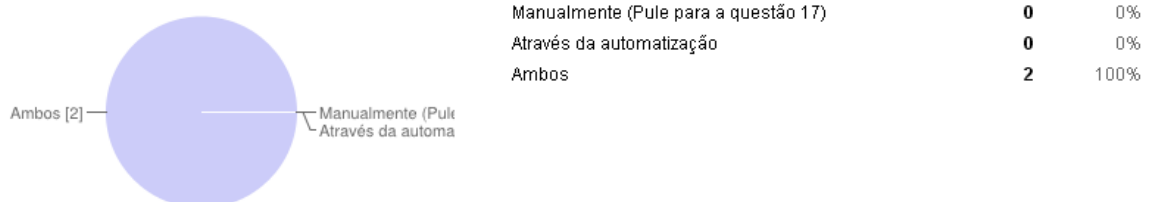
### 12. Quais são os erros mais comuns de serem encontrados nos sistemas desenvolvidos na empresa?



### 13. Quais as dificuldades encontradas na realização dos testes de software?



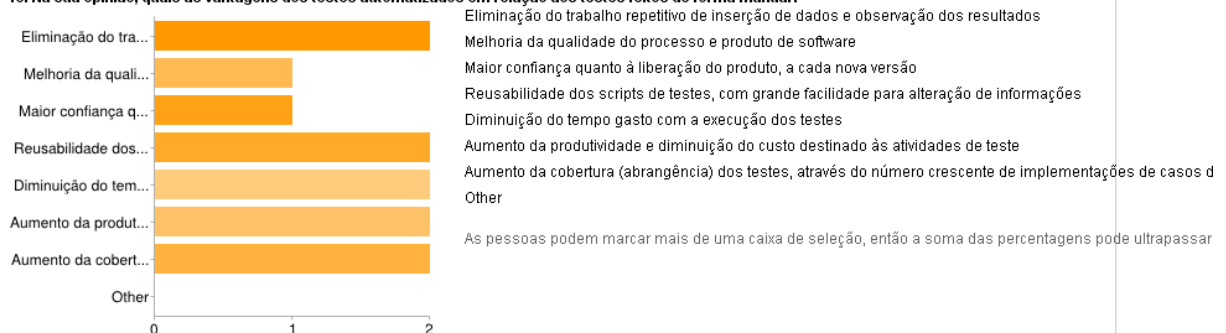
### 14. Como os testes são realizados?



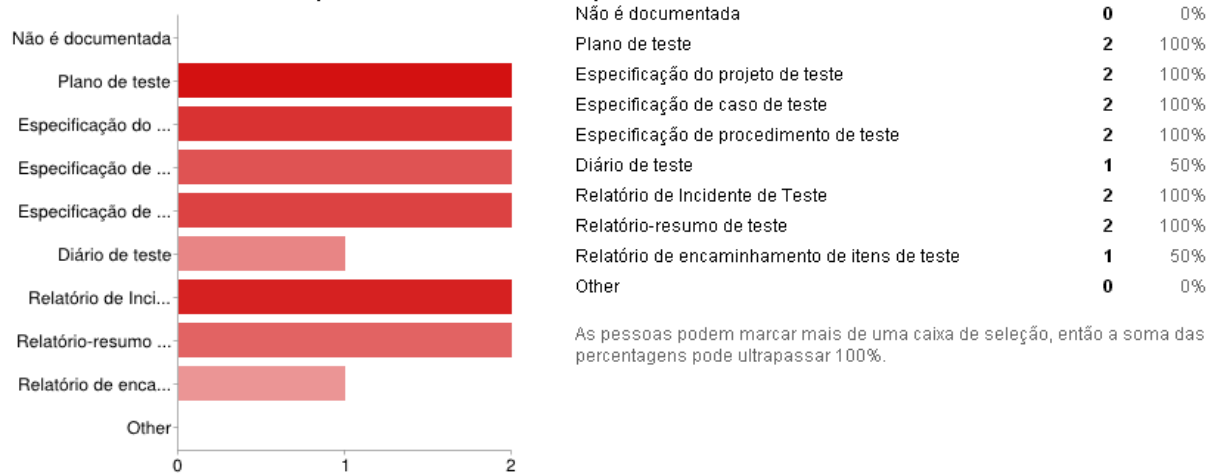
### 15. Se automatizado, quais ferramentas são utilizadas?

Funcional: Testcomplete, Sikuli, Rational Robot, Selenium Performance: Loadster, JMeter, Webload Segurança: Netsparker Selenium, IBM/Rational, HP, MicroFocus, JMeter dentre outras.

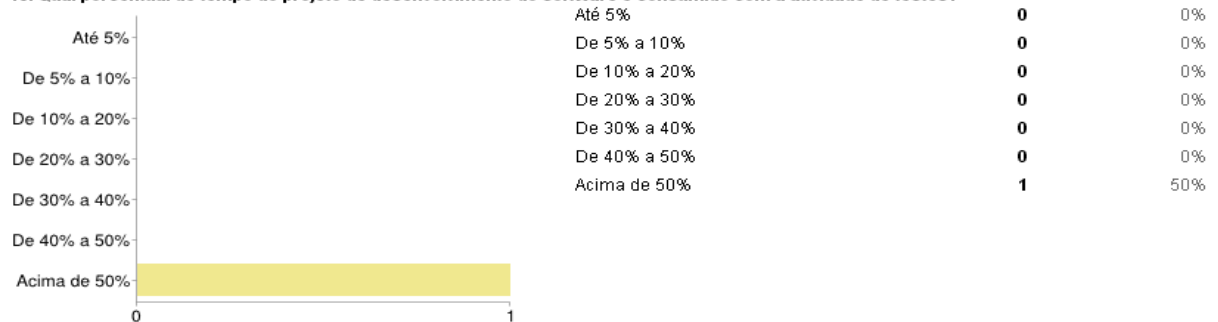
### 16. Na sua opinião, quais as vantagens dos testes automatizados em relação aos testes feitos de forma manual?



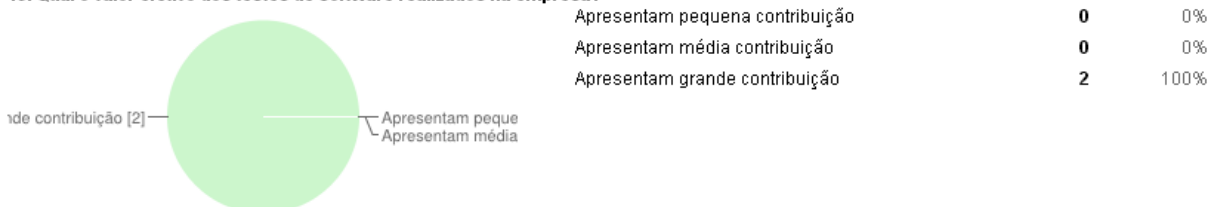
**17. Quais documentos abaixo fazem parte da atividade de teste na empresa?**



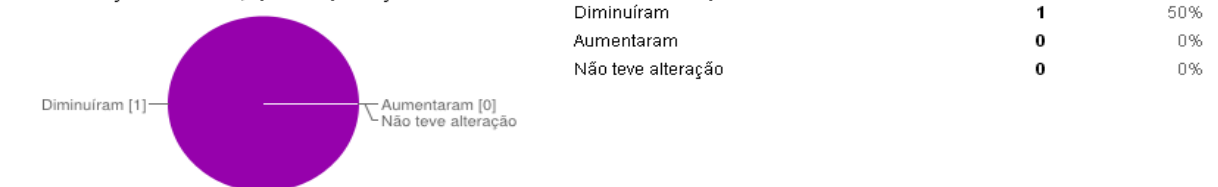
**18. Qual percentual de tempo do projeto de desenvolvimento do software é consumido com a atividade de testes?**



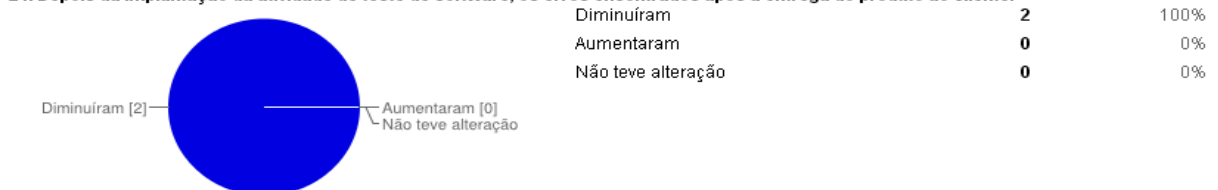
**19. Qual o valor efetivo dos testes de software realizados na empresa?**



**20. Com relação aos custos, após a implantação da atividade de teste observou-se que:**



**21. Depois da implantação da atividade de teste de software, os erros encontrados após a entrega do produto ao cliente:**



**22. Porque a empresa não realiza testes de softwares?**

