

**CURSO DE ESPECIALIZAÇÃO EM GESTÃO DE PROJETOS
EM TECNOLOGIA DA INFORMAÇÃO**

DAVID ALVES DE SOUZA JUNIOR

**ANÁLISE DO PROCESSO DE TESTE DE UMA INSTITUIÇÃO
FINANCEIRA**

São Paulo
Maio/2012

**CURSO DE ESPECIALIZAÇÃO EM GESTÃO DE PROJETOS
EM TECNOLOGIA DA INFORMAÇÃO**

DAVID ALVES DE SOUZA JUNIOR

**ANÁLISE DO PROCESSO DE TESTE DE UMA INSTITUIÇÃO
FINANCEIRA**

Monografia apresentada à Escola Politécnica da
Universidade de São Paulo para a obtenção do
Título de Especialista em Gestão da Tecnologia da
Informação.

ORIENTADOR: André Leme Fleury

São Paulo
Maio/2012

RESUMO

O processo de teste de software destina-se a mostrar que um sistema está atendendo as expectativas do cliente. Esse processo envolve algumas atividades como inspeções e revisões em cada fase do processo de desenvolvimento, desde a definição dos requisitos até a entrega do sistema (SOMMERVILLE, 2005).

Este trabalho tem como objetivo identificar os principais pontos problemáticos do processo de teste dos sistemas em uma instituição financeira. Para se conhecer as melhores práticas existentes, houve um levantamento bibliográfico sobre as mais conceituadas práticas e técnicas das atividades de teste de software. Com a finalidade de se conhecer os processos de teste da instituição, uma pesquisa quantitativa exploratória foi elaborada e enviada a 20 profissionais da área de desenvolvimento de sistemas.

As 17 respostas obtidas foram analisadas e confrontadas com o levantamento bibliográfico, os itens que apontavam divergência entre a instituição e a literatura foram marcados como pontos que necessitam de melhoria no processo de teste dos sistemas. Houve a elaboração de uma proposta com ações de melhoria no processo de teste dos sistemas para cada um dos dez itens identificados.

Com a aplicação das práticas sugeridas através deste estudo, espera-se obter um ganho substancial de qualidade dos sistemas, consequentemente, diminuição da sobrecarga sobre a equipe de sustentação e aumento da satisfação do cliente.

SUMÁRIO

| | |
|---|-------------|
| RESUMO..... | iii |
| LISTA DE FIGURAS..... | vi |
| LISTA DE TABELAS..... | vii |
| LISTA DE ABREVIACÕES E SIGLAS..... | viii |
| 1. INTRODUÇÃO | 9 |
| 1.1. Contexto..... | 9 |
| 1.2. Problema | 9 |
| 1.3. Objetivo | 10 |
| 1.4. Justificativa | 11 |
| 1.5. Estrutura do trabalho..... | 11 |
| 2. REVISÃO DE LITERATURA..... | 13 |
| 2.1. Desenvolvimento de software..... | 13 |
| 2.1.1. Processo de software | 13 |
| 2.1.2. Modelos de processo de software..... | 13 |
| 2.2. Documentação dos testes | 14 |
| 2.3. Processo de teste de software..... | 15 |
| 2.3.1. Técnicas de teste de software | 16 |
| 2.3.2. Teste de desempenho..... | 17 |
| 2.3.3. Teste de integração | 17 |
| 2.3.4. Teste de aceitação..... | 17 |
| 2.3.5. Teste de usabilidade..... | 18 |
| 2.3.6. Teste de regressão..... | 19 |
| 2.4. Ferramentas de automação de teste | 20 |
| 3. DESENVOLVIMENTO DA PESQUISA..... | 21 |
| 3.1. Método da pesquisa | 25 |
| 3.2. A pesquisa..... | 25 |
| 4. RESULTADOS OBTIDOS E DISCUSSÕES | 26 |
| 4.1. Pontos problemáticos do processo de teste..... | 42 |

| | |
|-----------------------------|-----------|
| 5. CONCLUSÕES..... | 44 |
| 6. REFERÊNCIAS | 45 |

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1 – Modelo V comparando as atividades de desenvolvimento e teste de software (CRAIG e JASKIEL, 2002) | 16 |
| Figura 2– Questão 01 do questionário..... | 26 |
| Figura 3 – Questão 02 do questionário..... | 27 |
| Figura 4 – Questão 03 do questionário..... | 28 |
| Figura 5 – Questão 04 do questionário..... | 29 |
| Figura 6 – Questão 05 do questionário..... | 30 |
| Figura 7 – Questão 06 do questionário..... | 31 |
| Figura 8 – Questão 07 do questionário..... | 32 |
| Figura 9 – Questão 08 do questionário..... | 33 |
| Figura 10 – Questão 09 do questionário..... | 34 |
| Figura 11 – Questão 10 do questionário..... | 35 |
| Figura 12 – Questão 11 do questionário..... | 37 |
| Figura 13 – Questão 12 do questionário..... | 38 |
| Figura 14 – Questão 13 do questionário..... | 39 |
| Figura 15 – Questão 14 do questionário..... | 40 |
| Figura 16 – Questão 15 do questionário..... | 41 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 – Tabela síntese com as perguntas enviadas aos respondentes e as respostas extraídas da fundamentação teórica. | 25 |
| Tabela 2 – Tabela síntese com os pontos de melhoria. | 43 |

LISTA DE ABREVIACES E SIGLAS

SURVEY Mtodo de pesquisa

MDS Metodologia de desenvolvimento de sistemas

RTF Roteiro de Teste Funcional

1. INTRODUÇÃO

1.1. Contexto

O trabalho será realizado em uma empresa de seguros que faz parte da holding de um grande banco. Atualmente, a área de TI conta com um quadro de aproximadamente 250 funcionários, 50 prestadores de serviços e um número bem grande de sistemas utilizando as mais diversas tecnologias. Todos os projetos de desenvolvimento de sistemas estão aderentes a uma metodologia de desenvolvimento chamada MDS, ou seja, qualquer alteração nos sistemas segue um processo que define atividades, papéis e produtos que devem ser gerados durante o ciclo de vida do projeto.

A TI está estrategicamente dividida em equipes de desenvolvimento de acordo com o negócio e dentro dessas equipes há uma subdivisão em dois grupos:

- **Projetos:** Pessoal focado exclusivamente para o desenvolvimento de novos projetos da empresa. Na sua grande maioria, são projetos que visam o negócio, entretanto essa equipe também pode atuar em projetos de TI. Por exemplo, atualização de versão de banco de dados, performance, etc.
- **Sustentação:** Equipe focada exclusivamente para o atendimento dos erros que ocorrem em produção. Dentre as atividades estão: Acompanhamento de execução de processos batches, erros em funcionalidades durante a operação do sistema e esclarecimento de dúvidas dos usuários. Para se obter uma maior eficiência no atendimento, há um comitê no qual os usuários realizam uma priorização dos erros, que causam maior impacto na operação e, por este motivo, necessitam de uma atuação de TI com maior urgência.

1.2. Problema

Não há na empresa, uma área, ou profissionais, dedicados à realização do planejamento e execução dos testes para os projetos de desenvolvimento de melhorias ou correções de erros. Os

mesmos são realizados pelos próprios analistas de sistemas, que possuem a incumbência de analisar, documentar, desenvolver, testar, homologar com o cliente e implantar.

Como relatado no tópico anterior a equipe de TI é dividida em equipes de projeto e sustentação. Para o desenvolvimento de novos projetos é exigida uma série de documentos para o projeto estar aderente à metodologia, no qual são passíveis de auditoria, dentre estes documentos, está o roteiro de teste funcional (RTF), documento utilizado para armazenar os casos de testes realizados nos projetos implantados, usado também para comprovar que o desenvolvedor testou e o cliente validou determinada funcionalidade.

Para as correções de erros, a metodologia de desenvolvimento exige outras documentações, dentre estes, não consta o roteiro de teste funcional. O teste que o desenvolvedor realiza para corrigir aquele erro, não fica armazenado em lugar algum e o mesmo se aplica na hora do cliente validar, o cliente realiza seus testes, sem nenhum critério definido pela TI e, quando estiver confortável em aplicar aquela correção em produção, envia um e-mail para a equipe de desenvolvimento que efetua a implantação do sistema em produção.

Fato é que a empresa possui muitos erros nos seus sistemas em ambiente de produção, e a quantidade de chamados abertos para correção de erros, geralmente, é maior do que a quantidade de erros solucionados no mesmo período. Esses erros acabam acarretando insatisfação do cliente com a TI, risco para a imagem da empresa, perda financeira, retrabalho e um capacity menor da equipe para o desenvolvimento de projetos de negócio, pois se houvesse uma quantidade menor de erros, poderiam ser disponibilizados um capacity maior para a equipe de projetos.

1.3. Objetivo

Diante deste cenário, este trabalho tem o objetivo de identificar os principais pontos problemáticos do processo de teste dos sistemas, essa identificação será obtida através da comparação dos processos de testes da empresa baseado nas melhores práticas e com isso, obter uma melhor qualidade nas entregas dos projetos.

Para realizar essa identificação, há necessidade de conhecer, da empresa, quais são os processos de teste que o sistema é submetido antes do mesmo ser entregue ao usuário. Sendo assim, este trabalho contará com o envolvimento de vários analistas de sistemas de várias equipes, que terão a incumbência de responder uma pesquisa a respeito de como são realizados os

testes nos sistemas. Uma pesquisa quantitativa exploratória será elaborada e enviada por e-mail para 20 profissionais, sendo todos da área de desenvolvimento de sistemas, no qual terão 10 dias úteis para preencherem o questionário e retornar o mesmo por e-mail. É preciso também conhecer quais são as melhores práticas realizadas, para isso este trabalho focará no levantamento bibliográfico sobre as melhores práticas e técnicas de teste de software.

Para ser atingido o objetivo de avaliar o processo de teste, haverá uma comparação entre as respostas dos analistas de sistemas e a literatura. Se não houver diferença, a conclusão será que a empresa está realizando as melhores práticas para execução dos testes nos seus sistemas, caso contrário, serão apontados pontos de melhorias em seus processos.

1.4. Justificativa

Por causa de diversos problemas no processo de teste dos sistemas, ultimamente a equipe de sustentação está com uma grande quantidade de erros em produção para serem corrigidos. Anualmente, esses erros causam um enorme dispêndio da empresa com a manutenção dos seus sistemas, insatisfação dos clientes e sobrecarga para a equipe de sustentação.

Com a identificação dos principais pontos problemáticos do processo de teste dos sistemas, a empresa poderá implementar a correção em seus processos e com isso obter uma qualidade maior nos seus projetos de desenvolvimento de sistemas, sejam eles de melhoria ou de correção.

A resolução deste trabalho trará benefícios para a empresa como um todo, pois o sistema com melhor qualidade significa menos retrabalho, maior satisfação do cliente, equipe de TI mais disponível para o desenvolvimento de projetos de negócio, menor quantidade de imprevistos, consequentemente, maior facilidade de planejamento e maior assertividade dos prazos acordados nos projetos.

1.5. Estrutura do trabalho

O presente trabalho foi dividido em seis capítulos. No capítulo um, é apresentado o contexto, o problema que será abordado, objetivo, justificativa e a estrutura do trabalho.

O capítulo dois apresenta a revisão da literatura, neste capítulo é exposto todo o referencial teórico que será utilizado como base para a identificação dos problemas e a proposta de solução. Neste capítulo são abordados dois temas: desenvolvimento de software e as atividades de teste.

O capítulo três apresenta qual foi a metodologia utilizada para o desenvolvimento da pesquisa.

O foco do capítulo quatro é apresentar os resultados que foram obtidos com a aplicação da pesquisa comparando as respostas com a literatura, identificar os principais pontos problemáticos do processo de teste dos sistemas e, a partir da literatura, propor melhorias.

No quinto capítulo é apresentada a conclusão do trabalho e perspectivas para trabalhos futuros.

Por fim, no capítulo seis são apresentadas as referências bibliográficas utilizadas neste trabalho.

2. REVISÃO DE LITERATURA

Este capítulo tem como objetivo principal buscar na literatura os conceitos que serão utilizados neste trabalho, que basicamente, possui dois temas principais: o desenvolvimento de software e as atividades de testes.

2.1. Desenvolvimento de software

2.1.1. Processo de software

Processo de software é um conjunto coerente de atividades e resultados, normalmente executados por engenheiros de software, que levam a produção de um software. Apesar de não haver nenhum processo de software ‘ideal’ há muita chance para melhorá-lo nas empresas, através de emprego de técnicas e a inclusão das melhores práticas (SOMMERVILLE, 2005). Dentro dos mais variados processo de desenvolvimento, há atividades que são fundamentais a todos eles, segundo SOMMERVILLE (2005) são:

- **Especificação de software:** Atividade de definição das funcionalidades e suas restrições na operação.
- **Projeto e implementação de software:** Atividade de produzir o software segundo sua especificação
- **Validação de software:** Atividade para garantir que o software faz o que o cliente deseja
- **Evolução de software:** Atividade necessária para atender as necessidades mutáveis do cliente.

2.1.2. Modelos de processo de software

Modelo de processo de software é uma reprodução abstrata de um processo de software. Cada modelo representa um processo, de uma maneira que disponibiliza apenas informações sobre o mesmo. Os modelos são abstrações úteis que podem ser utilizados de diferentes formas no desenvolvimento de software (SOMMERVILLE, 2005).

Segundo Sommerville (2005), os principais modelos são:

- **Modelo em cascata:** Modelo originado de outros processos de engenharia. O nome ‘cascata’ se deve a sequencia em cascata de uma fase para outra. O resultado de cada fase envolve documentos que devem ser aprovados. A fase seguinte não se inicia até que a fase anterior tenha sido finalizada.
- **Desenvolvimento evolucionário:** A ideia do desenvolvimento evolucionário é desenvolver uma parte inicial, apresentar o resultado aos usuários para apreciação, colher feedbacks e, dessa forma, aprimorar o software até sua conclusão.
- **Desenvolvimento formal de sistemas:** Modelo muito parecido com o modelo em cascata, entretanto a diferença principal é que seu processo de desenvolvimento emprega a transformação matemática de uma especificação de sistema em um programa executável.
- **Desenvolvimento orientado a reuso:** Reuso de software, normalmente ocorre em quase todos os projetos de desenvolvimento, porém de forma informal. Essa abordagem orientada a reuso, conta com uma base ampla de componentes de software reutilizáveis, que podem ser aproveitados e, valendo-se da infraestrutura, integrar facilmente esses componentes ao seu software em desenvolvimento.

2.2. Documentação dos testes

A documentação é de suma importância para o processo de desenvolvimento de software, incluindo o processo de qualidade. Documentos são essenciais para se manter uma base de conhecimento que possa ser reutilizado em projetos. Documentos bem estruturados aumentam a reusabilidade dos testes (PEZZÈ; YOUNG, 2008).

A inspeção em documentos é uma pratica que pode ser aplicada nos projetos e também no código fonte da aplicação. A inspeção pode ajudar na divulgação de boas práticas e padrões de qualidade. Qualquer fase do processo de software que não vier acompanhada de uma inspeção é uma oportunidade para um defeito aflorar (PEZZÈ; YOUNG, 2008).

2.3. Processo de teste de software

A finalidade do processo de teste de Software é mostrar que um software está dentro dos requisitos definidos no início de seu desenvolvimento e que atende a expectativa do cliente (SOMMERVILLE, 2005). Atividades de teste devem perdurar durante todo o projeto de desenvolvimento de sistemas de software, desde a elicitação dos requisitos até a entrega.(PEZZÈ;YOUNG, 2008)

Um processo de qualidade bem elaborado dilui atividades de testes durante todo o ciclo de desenvolvimento de sistema, organizando-as para ser o mais eficiente possível. Outro motivo para execução de atividades de qualidade o mais cedo possível, durante o projeto, é o tempo de introdução do erro e sua detecção. Um defeito introduzido durante a fase de construção é mais barato sua correção durante a fase de teste unitário do que se comparado com teste integrado, de sistema, aceitação ou em produção. (PEZZÈ;YOUNG, 2008)

Para serem mais eficazes, os testes devem ser dirigidos por pessoas ou equipes que não foram envolvidas com a codificação. Entende-se que dessa forma há maior probabilidade de se encontrar defeitos (BRUNELI, 2006, MYERS, 1979). Quando uma mesma pessoa é responsável por executar vários papéis, uma identificação bem definida de cada responsabilidade é essencial para garantir que tudo será realizado de forma adequada, pois em uma organização onde há equipes semiautônomas, ou seja, uma mesma pessoa executa um mesmo papel, normalmente, a ênfase em documentação formal é menor (PEZZÈ; YOUNG, 2008).

Um sistema que foi testado por um conjunto de testes sistemáticos, tende a possuir melhor nível de qualidade do que um sistema que foi testado superficialmente. Infelizmente não é possível gerar um conjunto de casos de testes que garanta que o sistema está isento de qualquer tipo de erro (PEZZÈ; YOUNG, 2008). Os casos de testes podem ser criados assim que os modelos de requisitos do cliente sejam finalizados, ou seja, após a consolidação do escopo do projeto (BRUNELI, 2006, MYERS, 1979). Os mesmos devem ser derivados a partir de uma especificação funcional do sistema, seja ela formal ou informal. Esse processo deve começar junto com a especificação de requisitos e continuar em cada nível do projeto (PEZZÈ; YOUNG, 2008). As organizações devem criar políticas de possíveis casos de testes ao invés de deixar essa responsabilidade por conta da equipe de desenvolvimento (SOMMERVILLE, 2005).

A figura 1 ilustra as fases do desenvolvimento e suas respectivas fases de teste de software.

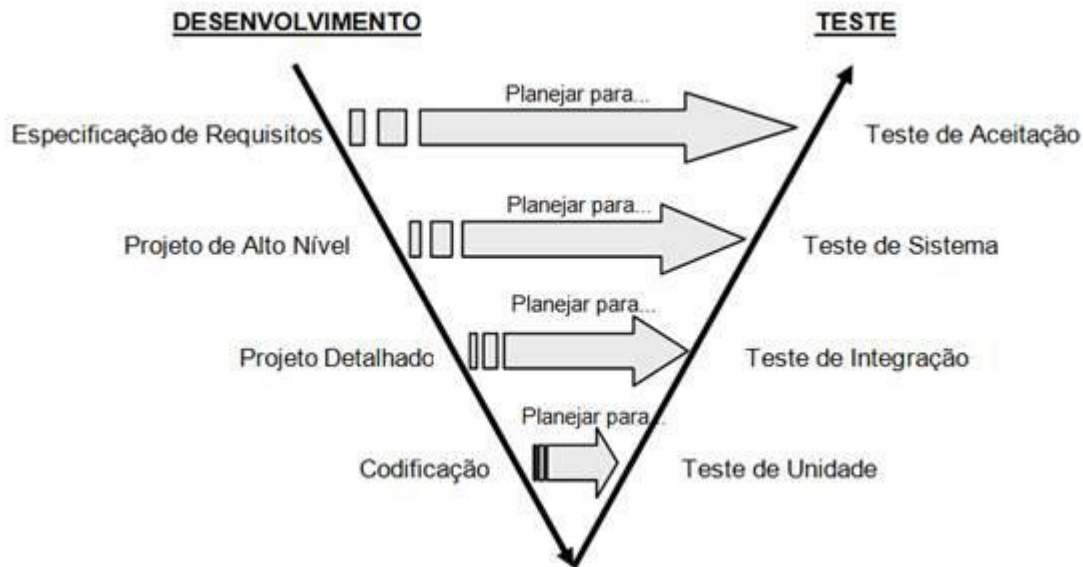


Figura 1 – Modelo V comparando as atividades de desenvolvimento e teste de software (CRAIG e JASKIEL, 2002).

2.3.1. Técnicas de teste de software

É impossível encontrar todos os erros de um programa. Deve-se estabelecer uma estratégia para cobrir a lógica do programa e garantir que todas as condições do projeto foram satisfeitas (CHAN et AL., 2007, MYERS, 2004). O projeto de testar um software pode ser tão desafiador quanto desenvolvê-lo. Elaborar casos de teste para encontrar todos os erros é uma atividade impraticável e impossível. O processo de testar um software pode ser feito de duas maneiras: Teste de caixa preta, que consiste em validar somente o comportamento da função que foi alterada. Teste de caixa branca, técnica que analisa o código fonte e realiza um caso de teste para cada caminho do código fonte (BRUNELI, 2006, PRESSMAN, 2002).

2.3.2. Teste de desempenho

Teste de desempenho é usado para garantir que o sistema conseguirá processar sua carga projetada. Isso envolve planejar testes no qual a carga no sistema é aumentada gradativamente até o sistema não dar conta de tamanha carga ou o tempo de processamento se tornar inviável (SOMMERVILLE, 2005).

Segundo SOMMERVILLE (2005), o teste de desempenho possui duas funções:

- Testar o comportamento de falha do sistema: O excesso de carga pode causar uma combinação de falhas que, de maneira alguma, possam causar erros mais graves como corrupção de dados, falhas em serviços, etc.
- Apresentação de erros inesperados: Devido à carga de processamento, excessivamente alta, podem surgir erros que em volume de processamento normal não se manifestariam.

2.3.3. Teste de integração

Teste de integração pode ser definido como um processo que visa identificar problemas de integração e compatibilidade entre os módulos do sistema. Falhas no teste de integração são causadas, muita das vezes, por especificações incompletas, teste unitário ineficiente ou implementações de interfaces, uso de recursos ou propriedades solicitadas com problemas (PEZZÈ; YOUNG, 2008).

Com um bom gerenciamento de projeto e boas práticas de programação é possível mitigar ou até mesmo extinguir alguns tipos de problemas de integração. Por exemplo, exigir o uso de alternativas seguras para: limpeza de memória, prevenção ao estouro de buffer, etc. (PEZZÈ; YOUNG, 2008).

2.3.4. Teste de aceitação

O teste de aceitação é um processo de validação, no qual o cliente participa, com o objetivo de julgar a utilidade e usabilidade ao invés de conformidade com a especificação de requisitos, com o propósito de auxiliar a decisão de liberar o produto para implantação. Essa

decisão pode ser baseada em medidas de produtos e de processos. As medidas baseadas em produtos consistem em inferências sobre testes estatísticos. Medidas de processos são fundamentadas em comparações e experiências prévias (PEZZÈ; YOUNG, 2008).

Um problema enfrentado em teste estatístico é que pode ser preciso uma grande quantidade de testes para se obter evidência do erro. Um exemplo disso é um sistema que executa um grande volume de processamento com uma taxa de assertividade de 99,9999%, falha de aproximadamente 31 vezes ao ano. Isto pode demandar um grande esforço de teste (PEZZÈ; YOUNG, 2008).

Uma abordagem menos formal, mas comumente utilizada é validação com os usuários. O procedimento normalmente executado é liberar, para os usuários, uma versão preliminar do sistema para ser testado no site do desenvolvedor, também chamado de teste alfa. Os usuários ficam incumbidos de darem retorno sobre usabilidade e as falhas encontradas. O chamado teste beta são as versões finais do sistema que são liberadas para serem testadas nos sites do cliente (PEZZÈ; YOUNG, 2008).

2.3.5. Teste de usabilidade

Um sistema usável torna a operação mais eficiente, facilita o aprendizado e é agradável de utilizar. Um teste de usabilidade envolve critérios como tempo, quantidade de operações para executar determinada funcionalidade, frequência de erros além da subjetiva satisfação (PEZZÈ; YOUNG, 2008).

Segundo PEZZÈ e YOUNG (2008) mesmo que a usabilidade dependa da percepção do usuário, pode se verificar e validar a usabilidade incluindo os seguintes passos:

- **Inspecionar as especificações.** A inspeção fornece uma resposta imediata da usabilidade
- **Testar os primeiros protótipos.** Um protótipo inicial pode não incluir funcionalidade alguma, pode ser simplesmente uma sequência de tela, mas dessa maneira o usuário consegue visualizar como será a usabilidade do sistema.
- **Testar as versões de maneira incremental.** A finalidade é acompanhar a evolução e se antecipar aos problemas.

- **Executar testes de sistema e aceitação.** É uma validação geral da usabilidade. Com esses testes é possível identificar as dificuldades e os obstáculos que os usuários encontram quando utilizam o sistema.

Um teste de usabilidade bem planejado e organizado produz resultados que são objetivos e diagnosticam com precisão a usabilidade nos usuários. Este teste pode ser aplicado nas diferentes fases dos projetos e com diferentes objetivos: trabalhar com o modelo mental do usuário, avaliar alternativa de projeto, padrões, etc. (PEZZÈ; YOUNG, 2008).

2.3.6. Teste de regressão

Quando ocorre uma alteração no sistema e se cria uma nova versão, uma funcionalidade pode ter sido alterada de forma não intencional. Algumas vezes, uma pequena alteração pode produzir resultados que levam o sistema a falhas. Quando uma nova versão do sistema alterou uma funcionalidade que deveria ser preservada, diz-se que a nova versão do sistema regrediu (PEZZÈ; YOUNG, 2008). Teste de regressão é importante para projetos de manutenção, pois o mesmo ratifica que as alterações introduzidas não afetaram o comportamento do sistema (BRUNELI, 2006, PRESSMAN, 2002).

O projeto disciplinado e técnicas de desenvolvimento diminuem a probabilidade de se obter uma regressão. Uma enfoque simples do teste de regressão consiste em re-executar todos os casos de testes das versões anteriores. Embora a solução seja simples, pode ser impossível executar todos os casos de testes de todas as versões, além de inviável operacionalmente, o custo disso pode ser altíssimo (PEZZÈ; YOUNG, 2008).

Uma boa documentação de testes é muito importante, pois uma solução para se evitar a regressão é manter ao longo das versões do sistema conjunto de casos de testes de alta qualidade, para isso é necessário remover casos de testes obsoletos, identificar testes redundantes e incluir novos casos de testes (PEZZÈ; YOUNG, 2008).

2.4. Ferramentas de automação de teste

As ferramentas de teste de software tendem a minimizar a influência humana durante os testes, aumentando a produtividade e a qualidade, influenciando a confiabilidade do software testado (MALDONADO, 1998, SOUZA, 1996). Os humanos são lentos e sujeitos a erros quando exercem atividades repetitivas. Por outro lado, as atividades simples e repetitivas são fáceis de automatizar, enquanto que julgamento e solução criativa permanecem fora do alcance de uma ferramenta de automação. Automatizar as atividades repetitivas, não apenas reduz custo como melhora a precisão. (PEZZÈ; YOUNG, 2008).

3. DESENVOLVIMENTO DA PESQUISA

Como já descrito, a área de TI veem enfrentando alguns problemas de qualidade nos sistemas entregues para os clientes. Ou seja, os sistemas são muitos frágeis. Frequentemente novos erros são descobertos e passados para a equipe de sustentação, no qual precisam atuar rapidamente para não interromper a operação e, dessa forma, tentar amenizar o impacto do erro sobre o cliente, até que se encontre a solução definitiva para o mesmo.

Fato é que esses erros acabam gerando insatisfação do cliente com a TI, perda financeira, foco menor da TI em relação aos novos projetos de negócio, risco para a imagem da empresa em relação aos clientes e parceiros, dentre outros problemas ocasionados por problemas no processo de teste de seus sistemas.

Sendo assim, este trabalho tem como objetivo, identificar os principais pontos problemáticos do processo de teste dos sistemas, através de uma comparação entre o modo que a empresa realiza seus testes e como a literatura sugere que seja feito. Com base nos resultados obtidos na comparação e no estudo bibliográfico, propor melhorias no processo de teste dos sistemas. Espera-se que após correção dos problemas apontados, a empresa possa obter uma qualidade maior nos seus sistemas.

A seguir são apresentadas as questões que foram enviadas aos respondentes e suas respectivas fundamentações teóricas:

| # | TEORIA | PERGUNTA |
|---|--|--|
| 1 | Atividades de teste devem perdurar durante todo o projeto de desenvolvimento de sistemas de software, desde a elicitação dos requisitos até a entrega (seção 2.3) (PEZZÈ;YOUNG, 2008). | Atividades de teste são executadas dentro de uma fase específica do ciclo de vida de desenvolvimento ou são executadas em vários momentos ao longo do projeto? |
| 2 | Quando uma mesma pessoa é responsável por executar vários papéis, uma identificação bem definida de cada responsabilidade é essencial para garantir que tudo será realizado de forma adequada (PEZZÈ;YOUNG, 2008). | Há uma norma na empresa que atribua a responsabilidade da execução dos testes para um cargo específico? |
| 3 | Um processo de qualidade bem elaborado dilui | Quais tipos de testes são executados nos |

| | | |
|---|---|--|
| | <p>atividades durante todo o ciclo de desenvolvimento de sistema, organizando-as para serem o mais eficiente possível (PEZZÈ;YOUNG, 2008).</p> <p>Outro motivo para execução de atividades de qualidade o mais cedo possível, durante o projeto, é o tempo de introdução do erro e sua detecção. Um defeito introduzido durante a fase de construção é mais barato sua correção durante a fase de teste unitário do que se comparado com teste integrado, de sistema, aceitação ou em produção (seção 2.3) (PEZZÈ;YOUNG, 2008).</p> | <p>projetos? Em que fase os mesmos são executados? Há alguma documentação?</p> |
| 4 | <p>Os casos de testes podem ser criados assim que os modelos de requisitos do cliente sejam finalizados, ou seja, após a consolidação do escopo do projeto (seção 2.3) (BRUNELI, 2006, MYERS, 1979).</p> | <p>Na prática, em que fase do projeto o RTF é criado?</p> |
| 5 | <p>Os casos de testes devem ser derivados a partir de uma especificação funcional do sistema, seja ela formal ou informal (seção 2.3) (PEZZÈ;YOUNG, 2008).</p> | <p>Como são criados os casos de teste?</p> |
| 6 | <p>A inspeção pode ser aplicada à maioria dos documentos dos projetos e também no código fonte. A inspeção pode ajudar na divulgação de boas práticas e padrões de qualidade (PEZZÈ;YOUNG, 2008). Qualquer fase do processo de software que não vier acompanhada de uma inspeção, é uma oportunidade para um defeito aflorar (seção 2.2) (PEZZÈ;YOUNG, 2008).</p> | <p>Há algum processo de inspeção (validação) dos documentos de testes criados?</p> |

| | | |
|----|--|---|
| 7 | O teste de aceitação é um processo de validação, no qual o cliente participa, com o objetivo de julgar a utilidade e usabilidade ao invés de conformidade com a especificação de requisitos (seção 2.3.4) (PEZZÈ;YOUNG, 2008). | Há alguma relação entre os requisitos do projeto e os casos de testes criados no RTF? |
| 8 | Os humanos são lentos e sujeitos a erros quando exercem atividades repetitivas. Por outro lado, as atividades simples e repetitivas são fáceis de automatizar, enquanto que julgamento e solução criativa permanecem fora do alcance de uma ferramenta de automação. Automatizar as atividades repetitivas, não apenas reduz custo como melhora a precisão (seção 2.4) (PEZZÈ;YOUNG, 2008). | Há algum nível de automação de casos de teste? |
| 9 | Documentos são essenciais para se manter uma base de conhecimento que possa ser reutilizado em projetos. Documentos bem estruturados aumentam a reusabilidade dos testes (seção 2.2) (PEZZÈ;YOUNG, 2008). | Para manutenções corretivas, os testes efetuados são documentados? |
| 10 | Para serem mais eficazes, os testes devem ser dirigidos por pessoas ou equipes que não foram envolvidas com a codificação. Entende-se que dessa forma há maior probabilidade de se encontrar defeitos (seção 2.3) (BRUNELI, 2006, MYERS, 1979). Em uma organização onde há equipes semiautônomas, ou seja, uma mesma pessoa executa um mesmo papel, normalmente, a ênfase em documentação formal é menor (seção 2.3) (PEZZÈ;YOUNG, 2008). | Nos projetos em que houve utilização do RTF, cite: <ul style="list-style-type: none"> • Quem criou o RTF? (Analista de sistemas, Programador, Analista de Teste, etc.). • Quem desenvolveu o sistema? (Analista de sistemas, Programador, Analista de Teste, etc.). • Quem validou o RTF? (Analista de |

| | | |
|----|---|---|
| | | <p>sistemas, Programador, Analista de Teste, etc.).</p> <ul style="list-style-type: none"> • Quem acompanhou o cliente na homologação? (Analista de sistemas, Programador, Analista de Teste, etc.). |
| 11 | Um sistema usável torna a operação mais eficiente, facilita o aprendizado e é agradável de utilizar (seção 2.3.5) (PEZZÈ; YOUNG, 2008). | Há algum processo que avalie a usabilidade do sistema? |
| 12 | Teste de regressão é importante para projetos de manutenção, pois o mesmo ratifica que as alterações introduzidas não afetaram o comportamento do sistema (seção 2.3.6) (BRUNELI, 2006, PRESSMAN, 2002). | Para preservar as funcionalidades já existentes, há algum procedimento nos projetos que realize os mesmos testes de projetos anteriores? |
| 13 | O processo de testar um software pode ser feito de duas maneiras: Teste de caixa preta, que consiste em validar somente o comportamento da função que foi alterada. Teste de caixa branca, técnica que analisa o código fonte e realiza um caso de teste para cada caminho do código fonte (BRUNELI, 2006, PRESSMAN, 2002). | Para criar os casos de testes, qual técnica de testes é utilizada? |
| 14 | As ferramentas de software tendem a minimizar a influência humana durante os testes, aumentando a produtividade e a qualidade, influenciando a confiabilidade do software testado (seção 2.4) (MALDONADO, 1998, SOUZA, 1996). | Há alguma ferramenta que auxilie para criação e/ou execução de testes? |
| 15 | É impossível encontrar todos os erros de um programa (CHAN et AL., 2007, MYERS, | Qual é o critério utilizado para estabelecer |

| | | |
|--|--|--|
| | 2004). Deve-se estabelecer uma estratégia para cobrir a lógica do programa e garantir que todas as condições do projeto foram satisfeitas (seção 2.3.1) (CHAN et AL., 2007, PRESSMAN, 2005). | o número de casos de testes suficientes? |
|--|--|--|

Tabela 1 – Tabela síntese com as perguntas enviadas aos respondentes e as respostas extraídas da fundamentação teórica.

3.1. Método da pesquisa

Para conhecer quais são os processos de teste que a empresa utiliza em seus sistemas, foi elaborado um questionário utilizando o método de pesquisa *survey*. Embasado na fundamentação teórica apresentada no capítulo 2, foram elaboradas 15 perguntas abertas com o objetivo de obter a opinião dos respondentes em relação aos processos de desenvolvimento e teste dos sistemas nos projetos de TI. Cada pergunta está associada a uma citação de um autor. Através das respostas do questionário, será possível identificar quais são os principais pontos problemáticos do processo.

3.2. A pesquisa

A pesquisa foi submetida a 20 profissionais, dos quais foram obtidas 17 respostas. Os respondentes são todos da área de desenvolvimento de sistemas e estão divididos entre os cargos de Coordenador, analistas Júnior, Pleno e Sênior. O canal utilizado para envio do questionário para os respondentes foi o e-mail, no qual tiveram 10 dias úteis para responderem a pesquisa utilizando-se do mesmo canal para envio da resposta. Para corroborar a imparcialidade da análise e o anonimato dos respondentes, as pesquisas respondidas foram armazenadas em um diretório específico, utilizando como nome do arquivo um número sequencial.

4. RESULTADOS OBTIDOS E DISCUSSÕES

Segue análise das respostas obtidas.

Questão 01: Atividades de teste são executadas dentro de uma fase específica do ciclo de vida de desenvolvimento ou são executadas em vários momentos ao longo do projeto?



Figura 2– Questão 01 do questionário

Fonte: Elaborado pelo Autor

Análise: A prática de realizar atividades de testes ao longo do projeto é exercida nos projetos da empresa. É de suma importância que as atividades de testes sejam realizadas desde a elicitação dos requisitos até a entrega do sistema, pois há um ganho substancial em qualidade do produto final.

Questão 02: Há uma norma na empresa que atribua a responsabilidade da execução dos testes para um cargo específico?

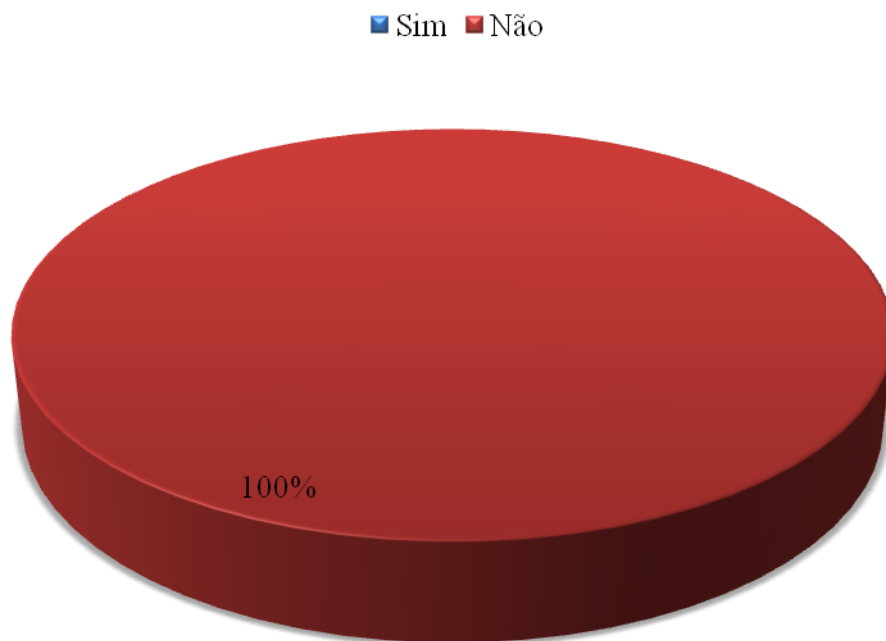


Figura 3 – Questão 02 do questionário

Fonte: Elaborado pelo Autor

Análise: Não há na empresa um cargo específico para a realização dos testes. Os testes acabam sendo realizados pela mesma pessoa que analisou e desenvolveu. A atividade de teste executada por uma mesma equipe diminui sua eficácia na detecção de erros.

Proposta: Segundo PEZZÈ e YOUNG (2008) quando uma mesma pessoa é responsável por executar vários papéis, uma identificação bem definida de cada responsabilidade é essencial para garantir que tudo será realizado de forma adequada. Sendo assim, uma boa prática será a **criação da figura do testador dentro de cada equipe, no qual será o responsável pela execução dos testes nos projetos.**

Questão 03: Quais tipos de testes são executados nos projetos? Em que fase os mesmos são executados? Há alguma documentação?

■ Integrado ■ Unitário, Integrado, Homologação ■ Não há ■ Homologação

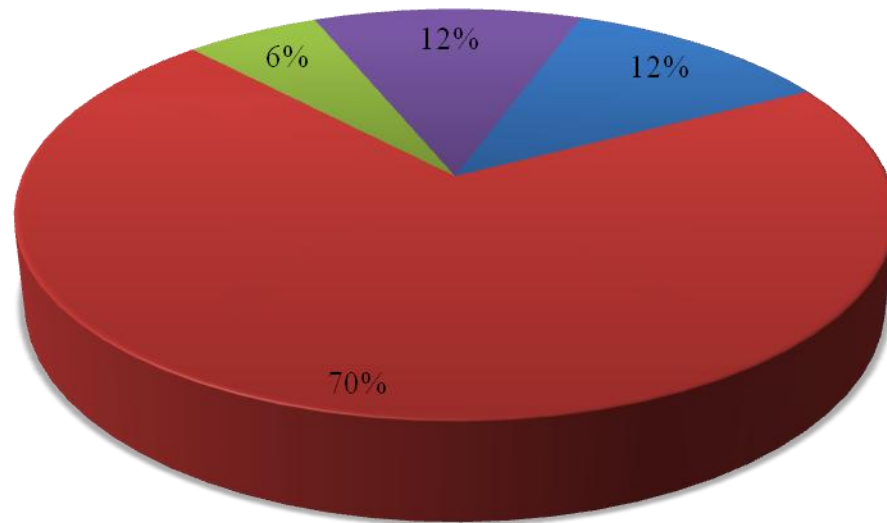


Figura 4 – Questão 03 do questionário

Fonte: Elaborado pelo Autor

Análise: Os testes estão divididos da seguinte forma nos projetos:

- Teste Unitário: O desenvolvedor realiza os testes unitários durante a codificação dos programas. Realizado na fase de construção.
- Teste Integrado: O analista realiza os testes após a codificação. Realizado na fase de teste integrado.
- Homologação: A homologação do sistema é realizada pelo usuário da aplicação, após a liberação do teste integrado. Realizado na fase de homologação.

O documento utilizado para auxiliar nos testes é o roteiro de teste funcional (RTF). Apesar de 30% desconhecer ou não executar todos os testes, no geral os processos de testes estão bem definidos e bem diluídos durante o ciclo de vida do projeto.

Questão 04: Na prática, em que fase do projeto o RTF é criado?

■ Após a construção ■ Análise e a Contrução ■ Análise ■ Não utiliza

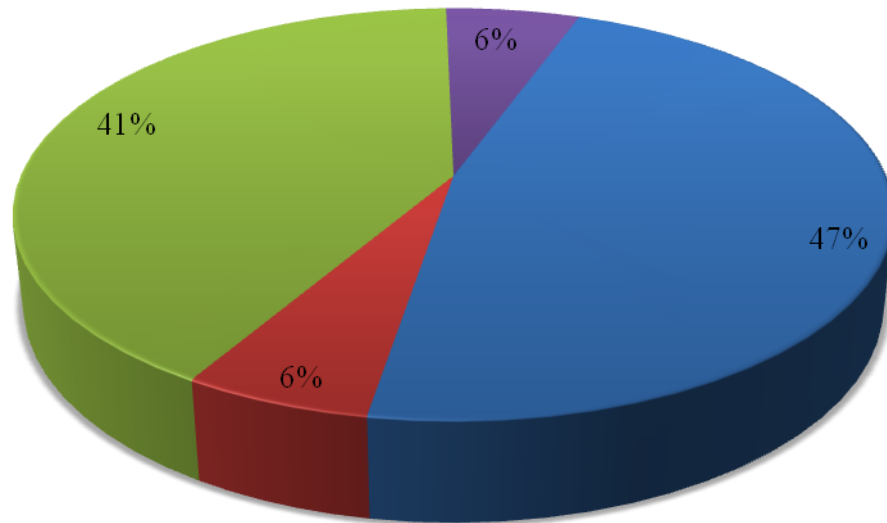


Figura 5 – Questão 04 do questionário

Fonte: Elaborado pelo Autor

Análise: 59% dos respondentes deixam para criar os casos de teste durante ou após a fase de construção. Essa prática acaba por criar casos de testes superficiais e ineficientes, pois há uma tendência normal de criar casos de testes somente onde houve alteração de código fonte.

Proposta: Segundo BRUNELI (2006) e MYERS (1979) os casos de testes podem ser criados assim que os modelos de requisitos do cliente sejam finalizados, ou seja, após a consolidação do escopo do projeto. Seguindo essa prática, **há necessidade de se criar um mecanismo de inspeção que não permita que o projeto avance para a fase de construção se o RTF não estiver criado.**

Questão 05: Como são criados os casos de teste?

- Baseado na DEP
- Baseado no que foi alterado / criado
- Baseado no DRS
- Não é criado

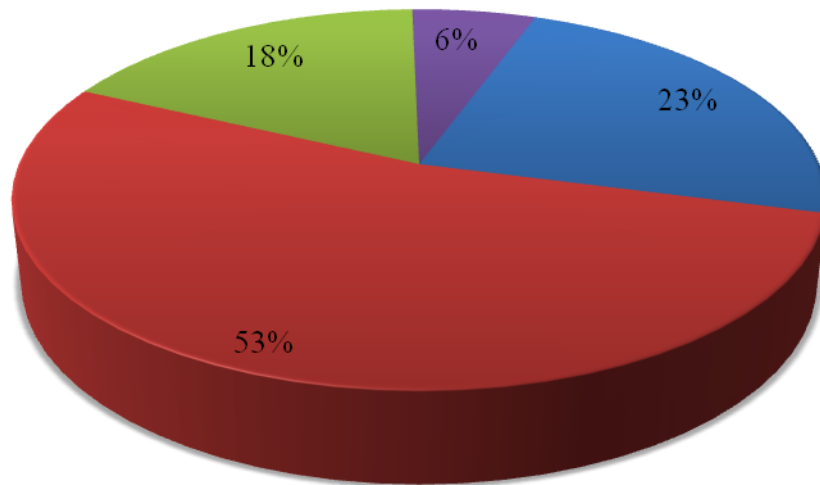


Figura 6 – Questão 05 do questionário

Fonte: Elaborado pelo Autor

Análise: Os casos de teste são criados baseado na funcionalidade alterada ou criada no projeto, isso foi o que 53% dos respondentes informaram. 18% baseiam-se no Documento de Requisitos do Sistema (DRS), enquanto que 23% utilizam a Definição do Escopo do Projeto (DEP) para criar seus testes. A resposta desta questão ratifica o que foi respondido na questão 04, no qual uma grande parte dos respondentes criam seu documento de testes após a construção do código fonte. Essa prática impacta na qualidade do sistema, pois não há uma verificação se todas as especificações do projeto foram validadas.

Proposta: Segundo PEZZÈ e YOUNG (2008) os casos de testes devem ser derivados a partir de uma especificação funcional do sistema, seja ela formal ou informal. **Os documentos de testes criados para a execução dos testes integrados devem possuir um campo que faça o relacionamento entre os requisitos do sistema e os casos de testes.**

Questão 06: Há algum processo de inspeção (validação) dos documentos de testes criados?

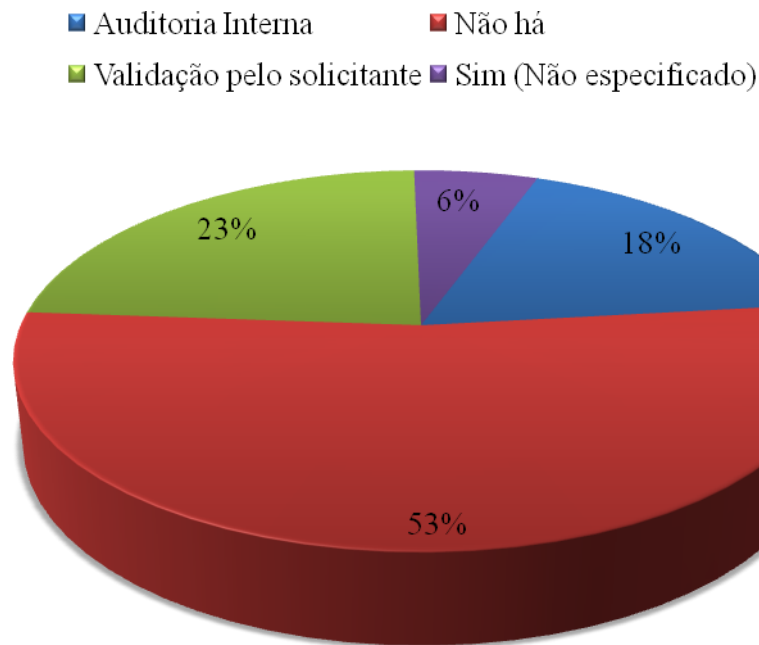


Figura 7 – Questão 06 do questionário

Fonte: Elaborado pelo Autor

Análise: A prática de inspeção não é executada nos projetos. Isso é o que foi apontado por 71% dos respondentes, pois a prática de inspeção da auditoria interna, apontada por 18%, pode ser realizada meses após a finalização do projeto. O fato de não haver inspeção dos documentos ou de código fonte cria-se uma grande possibilidade artefatos de baixa qualidade serem produzidos, impactando diretamente a qualidade dos sistemas.

Proposta: Segundo PEZZÈ e YOUNG (2008) a inspeção pode ser aplicada à maioria dos documentos dos projetos e também no código fonte. A inspeção pode ajudar na divulgação de boas práticas e padrões de qualidade. Qualquer fase do processo de software que não vier acompanhada de uma inspeção, é uma oportunidade para um defeito aflorar. **Com a finalidade da garantia da qualidade, há necessidade de se criar inspeção em código fonte e nos documentos criados nos projetos.**

Questão 07: Há alguma relação entre os requisitos do projeto e os casos de testes criados no RTF?

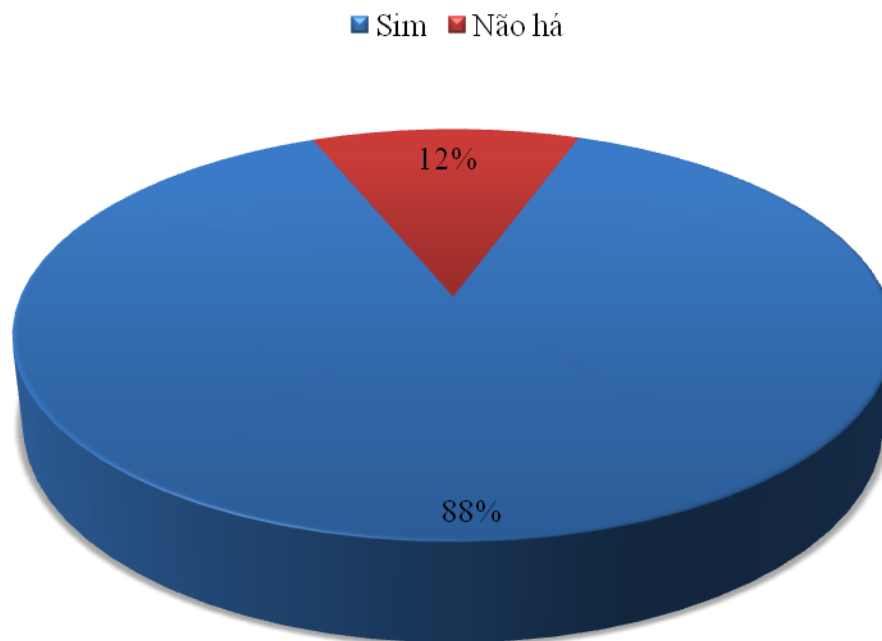


Figura 8 – Questão 07 do questionário

Fonte: Elaborado pelo Autor

Análise: A pesquisa constatou que o documento que é utilizado como base para a realização da homologação possui uma ligação com os requisitos do projeto. Neste item a empresa reforça o teste de aceite realizando uma ligação entre os requisitos do projeto com os casos de teste. Um ponto de atenção para esta prática é o cuidado que se deve ter para não se criar somente casos de teste que contenham ligação com os requisitos do projeto.

Questão 08: Há algum nível de automação de casos de teste?

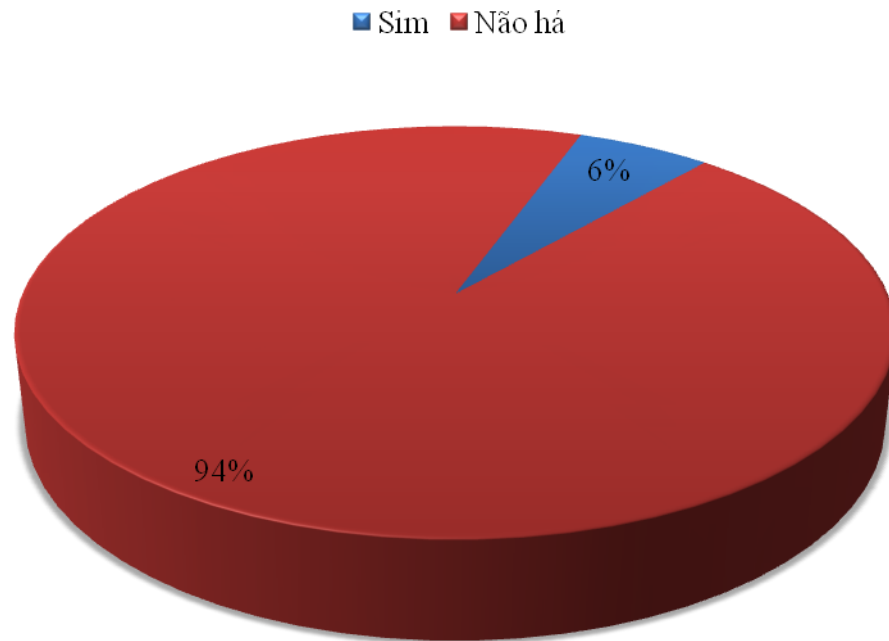


Figura 9 – Questão 08 do questionário

Fonte: Elaborado pelo Autor

Análise: Não há automatização para a realização dos testes, todos os casos de testes são criados e executados manualmente. Essa prática pode acabar encobertando erros simples que não são descobertos simplesmente por causa que o analista, por algum motivo, não criou caso de teste para determinada situação.

Proposta: Segundo PEZZÈ e YOUNG (2008) os humanos são lentos e sujeitos a erros quando exercem atividades repetitivas. Por outro lado, as atividades simples e repetitivas são fáceis de automatizar, enquanto que julgamento e solução criativa permanecem fora do alcance de uma ferramenta de automação. Automatizar as atividades repetitivas, não apenas reduz custo como melhora a precisão. Sendo assim, **é preciso preparar os sistemas para que os mesmos possam ser testados através de ferramentas de automatização de testes e, dessa forma, ajudar a diminuir esforços nos projetos e melhorar a qualidade do produto final.**

Questão 09: Para manutenções corretivas, os testes efetuados são documentados?

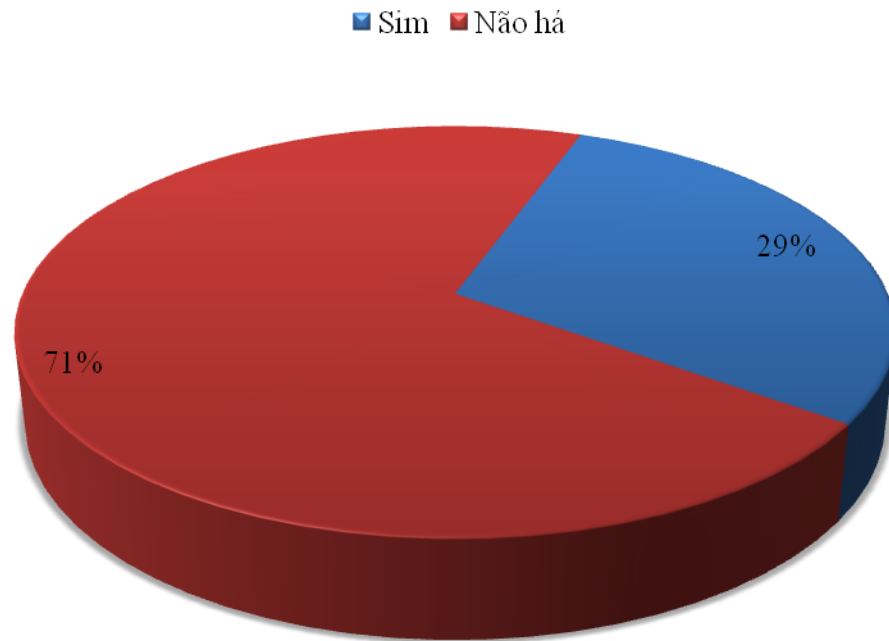


Figura 10 – Questão 09 do questionário

Fonte: Elaborado pelo Autor

Análise: Quando se trata de correção de erro em produção, não há documentação dos testes que foram realizados. A utilização desta prática inviabiliza a realização desse mesmo teste para outros projetos, além de se perder o conhecimento à respeito dos testes no qual um sistema foi submetido.

Proposta: Segundo PEZZÈ e YOUNG (2008) documentos são essenciais para se manter uma base de conhecimento que possa ser reutilizado em projetos. Documentos bem estruturados aumentam a reusabilidade dos testes. Dessa forma **há necessidade de se utilizar o RTF para também documentar projetos de correção de erros.**

Questão 10: Nos projetos em que houve utilização do RTF, cite:

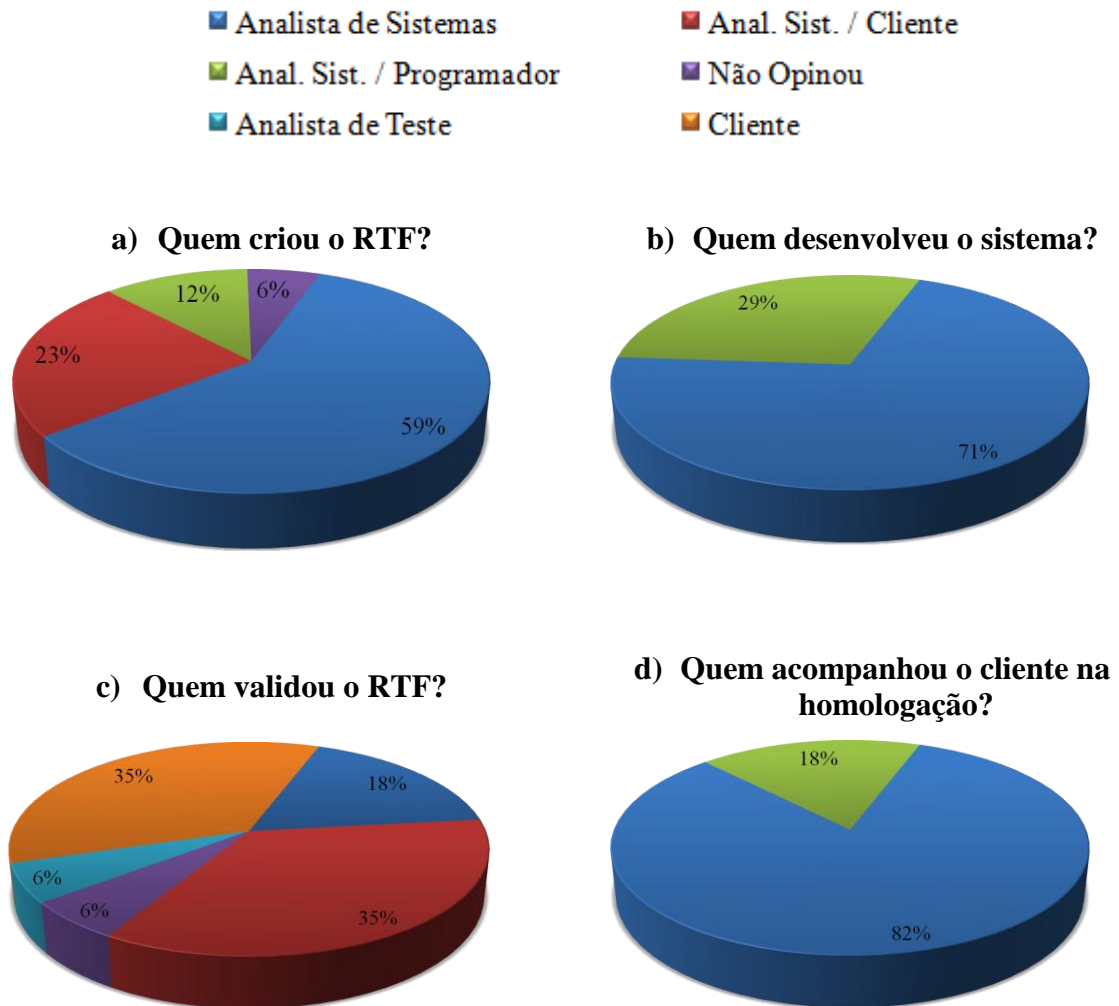


Figura 11 – Questão 10 do questionário

Fonte: Elaborado pelo Autor

Análise: O analista de sistemas é o responsável pela criação do RTF, desenvolvimento e homologação do sistema, além de ser, junto com o cliente, responsável pela validação do roteiro de teste funcional. Para esse item, há um **acúmulo de papéis**, pois o analista de sistema é cobrado pelo desenvolvimento e pelo acompanhamento nos testes. Este acúmulo de funções afeta diretamente a qualidade do produto final.

Proposta: Segundo BRUNELI (2006) e MYERS (1979) para serem mais eficazes, os testes devem ser dirigidos por pessoas ou equipes que não foram envolvidas com a codificação. Entende-se que dessa forma há maior probabilidade de se encontrar defeitos. Sendo assim, **uma boa prática será a utilização da figura do testador, proposto na questão 02 como responsável pelos testes nos projetos.**

Questão 11: Há algum processo que avalie a usabilidade do sistema?

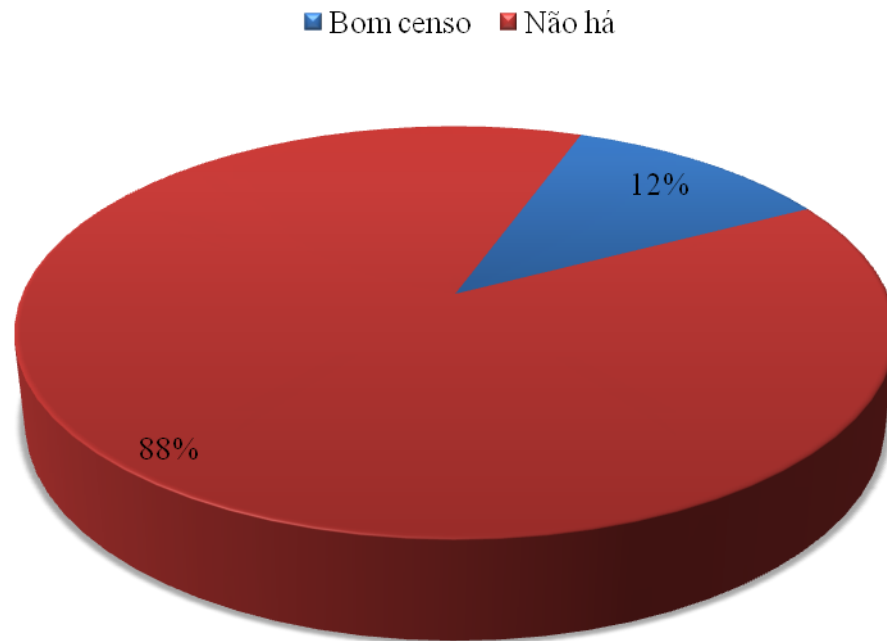


Figura 12 – Questão 11 do questionário

Fonte: Elaborado pelo Autor

Análise: Segundo os respondentes, **não há nenhum processo de avaliação da usabilidade do sistema**. A partir do momento que não há nenhuma validação ou regra de funcionamento sobre usabilidade dos sistemas, há uma possibilidade maior de problemas surgirem a esse respeito. Um sistema com usabilidade ruim pode induzir os usuários a cometerem erros mais facilmente. Sendo que, sistemas com boa usabilidade contribuem para o aumento de eficiência operacional, consequentemente, melhorando os resultados da companhia.

Proposta: Segundo PEZZÈ e YOUNG (2008) um sistema usável torna a operação mais eficiente, facilita o aprendizado e é agradável de utilizar. Portanto, **há necessidade de se criar um processo que, quando necessário, realize uma validação da usabilidade dos sistemas nos projetos**.

Questão 12: Para preservar as funcionalidades já existentes, há algum procedimento nos projetos que realize os mesmos testes de projetos anteriores?

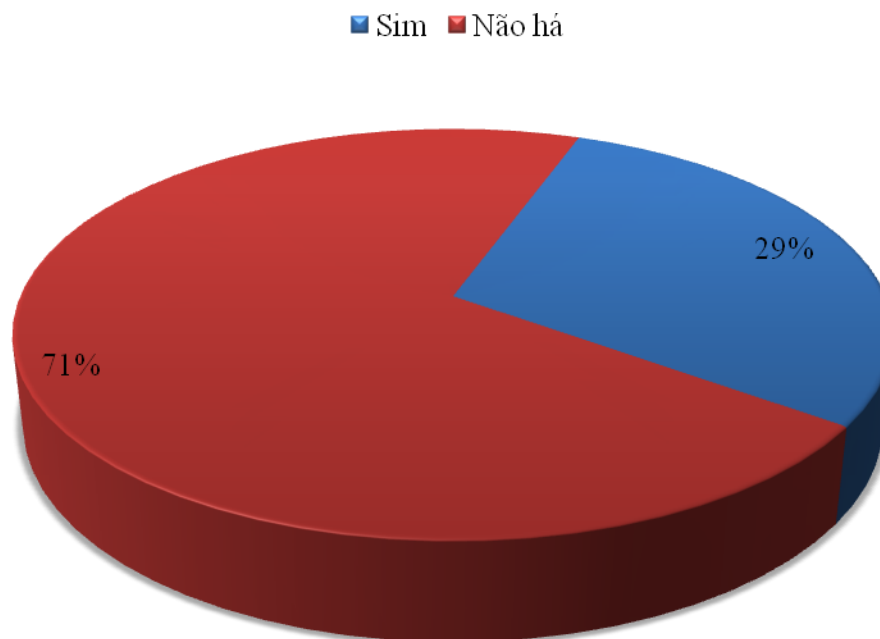


Figura 13 – Questão 12 do questionário

Fonte: Elaborado pelo Autor

Análise: A atividade de realizar testes de projetos anteriores não é realizada. Apenas 29% realizam esse tipo de atividade, trata-se de uma parcela muito baixa. É de suma importância a realização de testes que possuam a finalidade de verificar se a correção ou alteração efetuada não impactou as funcionalidades pré-existentes.

Proposta: Segundo BRUNELI (2006) e PRESSMAN (2002) teste de regressão é importante para projetos de manutenção, pois o mesmo ratifica que as alterações introduzidas não afetaram o comportamento do sistema. Baseado na literatura **há necessidade de exigir que nos projetos de melhoria ou correção, exista a atividade de teste de projetos anteriores, cuja única finalidade é verificar se a alteração realizada não causou nenhum impacto em funcionalidades já existentes.**

Questão 13: Para criar os casos de testes, qual técnica de testes é utilizada?

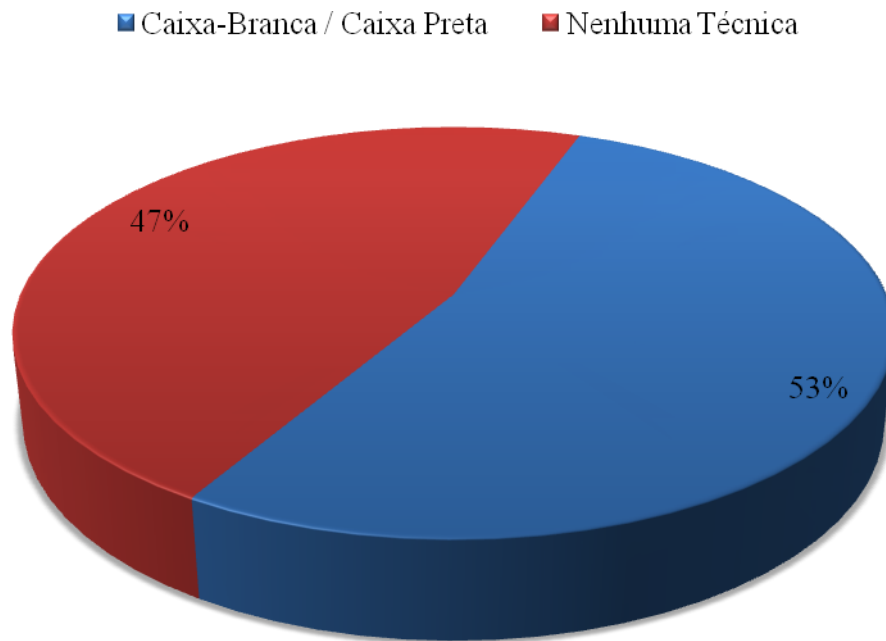


Figura 14 – Questão 13 do questionário

Fonte: Elaborado pelo Autor

Análise: Para este item 53% dos respondentes utilizam a técnica de caixa-branca ou caixa-preta para a criação de seus casos de testes. A utilização de técnicas para a criação de testes é de suma importância para a elaboração de testes eficientes.

Questão 14: Há alguma ferramenta que auxilie para criação e/ou execução de testes?

■ Não há

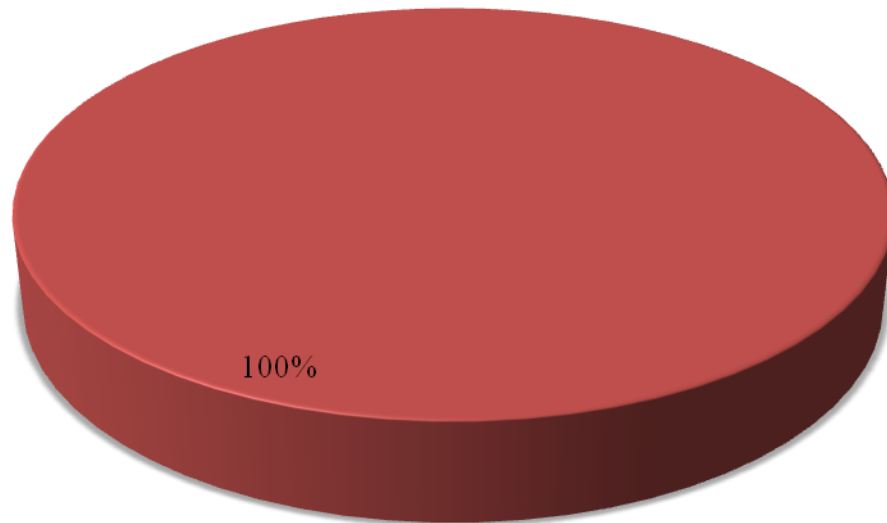


Figura 15 – Questão 14 do questionário

Fonte: Elaborado pelo Autor

Análise: Não há na empresa nenhum tipo de ferramenta que auxilie na criação, execução ou na gestão dos testes. A resposta obtida nesta questão e na questão 08 corrobora a tese de que todos os testes são criados e executados manualmente. Com isso, a qualidade do software está totalmente dependente de seus analistas, não há nenhuma ferramenta que aponte, previamente, que há um problema no sistema desenvolvido.

Proposta: Segundo MALDONADO (1998) e SOUZA (1996) as ferramentas de software tendem a minimizar a influência humana durante os testes, aumentando a produtividade e a qualidade, influenciando a confiabilidade do software testado. Com base na literatura, **há necessidade de utilizar ferramentas de testes com a finalidade de diminuir a influência direta do analista com a qualidade final dos sistemas.**

Questão 15: Qual é o critério utilizado para estabelecer o número de casos de testes suficientes?

■ Funcionalidades Relacionadas ■ Não há
■ Quantidade de Requisitos ■ Acordo entre TI e Cliente
■ Funcionalidades Alteradas

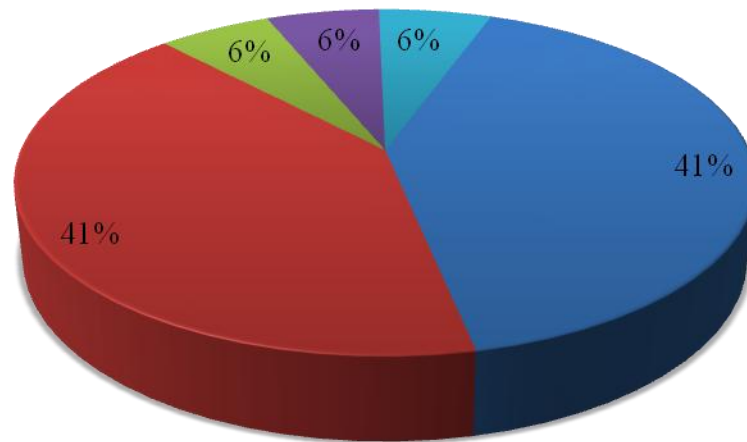


Figura 16 – Questão 15 do questionário

Fonte: Elaborado pelo Autor

Análise: Para este item 59% dos respondentes estabelecem uma estratégia para que seus testes possam ter a maior cobertura possível. Esta atividade a empresa vem executando de acordo com o sugerido pela literatura. Estabelecer o número de casos de testes que serão criados para validar determinado projeto é importante, pois, se for estabelecido um número excessivo de casos de testes, o projeto ficará mais caro e possivelmente haverá atraso no cronograma, por conta da quantidade de casos de testes, o oposto também é ruim, pois dessa forma os testes realizados serão superficiais e os sistemas estarão mais suscetíveis à erros em produção.

4.1. Pontos problemáticos do processo de teste

No tópico anterior, foram apresentadas as respostas obtidas através da aplicação do questionário na empresa. As respostas foram tabuladas, analisadas e comparadas com as boas práticas que a literatura sugere. Propostas de melhoria foram apresentadas para cada item onde se apresentou divergência entre a prática na empresa e o que sugere a literatura.

No quadro abaixo será apresentada uma síntese dos dez pontos identificados no tópico anterior que necessitam de melhoria.

| PROBLEMAS IDENTIFICADOS | PROPOSTAS DE MELHORIA |
|---|--|
| Não há na empresa um cargo específico para a realização dos testes. | Criação da figura do testador dentro de cada equipe, no qual será o responsável pela execução dos testes nos projetos. |
| 59% dos respondentes deixam para criar os casos de teste durante ou após a fase de construção. | Criar um mecanismo de inspeção que não permita que o projeto avance para a fase de construção se o RTF não estiver criado. |
| Os casos de teste são criados baseado na funcionalidade alterada ou criada no projeto. | Os documentos de testes criados para a execução dos testes integrados devem possuir um campo que faça o relacionamento entre os requisitos do sistema e os casos de testes. |
| A prática de inspeção não é executada nos projetos. | Criar inspeção em código fonte e nos documentos criados nos projetos. |
| Não há automatização para a realização dos testes, todos os casos de testes são criados e executados manualmente. | Preparar os sistemas para que os mesmos possam ser testados através de ferramentas de automatização de testes e, dessa forma, ajudar a diminuir esforços nos projetos e melhorar a qualidade do produto final. |
| Quando se trata de correção de erro em produção, não há documentação dos testes que foram realizados. | Utilizar o RTF para também documentar projetos de correção de erros. |
| Acúmulo de papéis. | Utilização da figura do testador, proposto na questão 02, como responsável pelos testes nos |

| | |
|---|---|
| | projetos. |
| Não há nenhum processo de avaliação da usabilidade do sistema. | Criar um processo que, quando necessário, realize uma validação da usabilidade dos sistemas nos projetos. |
| A atividade de realizar testes de projetos anteriores não é realizada. | Exigir que nos projetos de melhoria ou correção, exista a atividade de teste de projetos anteriores, cuja única finalidade é verificar se a alteração realizada não causou nenhum impacto em funcionalidades já existentes. |
| Não há na empresa nenhum tipo de ferramenta que auxilie na criação, execução ou na gestão dos testes. | Utilizar ferramentas de testes com a finalidade de diminuir a influência direta do analista com a qualidade final dos sistemas. |

Tabela 2 – Tabela síntese com os pontos de melhoria.

5. CONCLUSÕES

Neste trabalho foram apresentados dois assuntos principais: Desenvolvimento de software e atividades de testes, nos quais foram utilizados para dar embasamento ao trabalho. Houve a aplicação de um questionário na empresa que teve a finalidade de conhecer as principais atividades de testes.

Das pessoas que receberam o questionário, 85% responderam a pesquisa em até 10 dias úteis. Após a análise dos resultados, foi realizada uma comparação entre as respostas obtidas e a literatura, no qual foram identificados dez pontos divergentes.

Com a identificação desses pontos problemáticos do processo de teste, para cada item, buscou-se na literatura uma proposta de melhoria das atividades de teste.

A expectativa com a aplicação das melhores práticas aqui sugeridas é obter uma significativa melhora da qualidade dos sistemas, conseqüentemente, diminuição da sobrecarga de trabalho para a equipe de sustentação e aumento da satisfação dos clientes com a área de desenvolvimento de sistemas.

A sugestão para a continuidade a esse trabalho é implantar as propostas sugeridas neste estudo nas atividades de teste da empresa.

6. REFERÊNCIAS

- SOMMERVILLE, I. **Engenharia de Software** 6. ed. São Paulo: Addison-Wesley Brasil, 2005. 592 p.
- BRUNELI, Marcus V. D. Q. **A Utilização de uma Metodologia de Teste no Processo da Melhoria da Qualidade**, Dissertação de Mestrado, UNICAMP, Campinas, SP, Fevereiro 2006
- CRAIG, R.D.; JASKIEL, S. P., **Systematic Software Testing**.1.ed. Boston: Artech House Publishers, Estados Unidos, 2002.
- SOUZA, S.R.S.; **Avaliação do Custo e Eficácia do Critério Análise de Mutantes na Atividade de Teste de Software**, Dissertação de Mestrado, ICMSC/USP, São Carlos, SP, Junho, 1996.
- CHAN, Alessandra ; CAGNIN, Maria Istela ; MALDONADO, José Carlos ; BRAGA, R. T. V. . **Um ambiente para apoiar a utilização de padrões de software e requisitos de teste no desenvolvimento de aplicações**. In: Sexta Conferência Latino-americana em Linguagens de Padrões para Programação, 2007, Porto de Galinhas - PE. Proceedings do SugarLoafPLoP 2007. Recife - PE : Universidade de Pernambuco, 2007. v. 1. p. 235-250
- MALDONADO, Jose Carlos ; VINCENZI, Auri Marcelo Rizzo ; BARBOSA, Ellen Francine ; SOUZA, S. R. S. ; DELAMARO, Márcio Eduardo . **Aspectos Teóricos e Empíricos de Teste de Cobertura de Software**. In: Curitiba; Blumenau; Pelotas. (Org.). Revista da Escola de Informática da SBC - Regional Sul. : , 1998, v. I, p. 53-86.
- PEZZÈ, MAURO; YOUNG, MICHAL **Teste e Análise de Software**. São Paulo: Bookman, 2008 512 p.
- MYERS, G. **The Art of Software Testing**. New York: Wiley, 1979
- PRESSMAN, R. S. **Engenharia de Software**. 5ª ed. Rio de Janeiro: McGraw-Hill, 2002, cap. 17-18.
- MYERS, G. J. (2004). The art of software testing. John Wiley & Sons, Inc., 2th. edition.
- PRESSMAN, R. S. (2005). Engenharia de Software. McGraw-Hill, 6th. edition.