

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Renata Gonzaga Salomão

**Análise da Relevância de Testes de Regressão
para o Mercado de Desenvolvimento de
Software do Triângulo Mineiro**

Uberlândia, Brasil

2016

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Renata Gonzaga Salomão

**Análise da Relevância de Testes de Regressão para o
Mercado de Desenvolvimento de Software do Triângulo
Mineiro**

Trabalho de conclusão de curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, Minas Gerais, como
requisito exigido parcial à obtenção do grau
de Bacharel em Sistemas de Informação.

Orientador: William Chaves de Souza Carvalho

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2016

Resumo

Este trabalho descreve quatro abordagens para a realização teste de regressão utilizadas em desenvolvimento de software e apresenta os resultados de uma pesquisa exploratória e quantitativa que foi realizada com 17 empresas de desenvolvimento de software da região do Triângulo Mineiro para identificar a relevância dos testes de regressão e qual volume de investimento pode ser esperado nesta área. Testes regressivos asseguram que mudanças feitas no software não alterem comportamentos de componentes que permaneceram inalterados. A quantidade e complexidade dos casos de teste tende a aumentar na medida em o projeto avança e a execução de testes de regressão em todas as suítes de testes pode ser inviável. Várias abordagens de seleção de casos de teste para regressão foram criadas para balancear o custo de execução e a qualidade dos resultados: reteste total, seleção de casos de teste, redução da suíte de teste e priorização de casos de teste. Os resultados da pesquisa mostram que apesar das empresas considerarem que os testes de regressão são relevantes, elas não investem suficientemente na área devido a falsas crenças criadas pelo não conhecimento total da teoria relacionada à qualidade de software. Na sua conclusão o artigo apresenta e discute alguns problemas em aberto e direções para trabalhos futuros.

Palavras-chave: Até, cinco, palavras-chave, separadas, por, vírgulas.

Lista de ilustrações

Figura 1 – Percentual de utilização de linguagens de programação pelas empresas	22
Figura 2 – Percentual de esforço em atividades de teste	24
Figura 3 – Importância dos testes de regressão para as empresas	25
Figura 4 – Profissionais responsáveis pelo planejamento dos testes de regressão . .	26
Figura 5 – Abordagens de testes de regressão utilizadas pelas empresas	26
Figura 6 – Forma de execução dos testes automatizados	27
Figura 7 – Tipo e percentual de distribuição de equipes de teste entre as empresas	27
Figura 8 – Investimento em ferramentas de teste por faixa de valores	28

Lista de tabelas

Tabela 1 – Vantagens e desafios da utilização de testes de regressão	18
--	----

Sumário

1	INTRODUÇÃO	7
1.1	Um pouco de História	7
1.2	Contexto	8
1.3	Objetivo	10
1.4	Hipótese	10
1.5	Organização do Trabalho	11
2	REFERENCIAL TEÓRICO	12
2.1	Teste de Software	12
2.2	Métricas e Indicadores para Teste	14
2.3	Teste Regressivo	15
2.3.1	Reteste Total	15
2.3.2	Seleção de Testes para Regressão	16
2.3.3	Redução da Suite de Teste	16
2.3.4	Priorização de Suite de Teste	16
2.4	Vantagens e Desafios dos Testes de Regressão	17
3	METODOLOGIA	19
3.1	Questionário	19
3.2	Variáveis de Interesse	20
3.3	Método de Análise	20
4	APRESENTAÇÃO DOS RESULTADOS	22
4.1	Perfil das Empresas	22
4.2	Características de Qualidade de Software	23
4.3	Relevância das Atividades de Teste	23
4.4	Documentação de Teste	24
4.5	Relevância dos Testes de Regressão	25
4.5.1	Percepção da importância dos testes de regressão	25
4.5.2	Perfil dos profissionais responsáveis pelo planejamento das regressões	25
4.5.3	Abordagens de seleção dos casos de teste para regressão	26
4.5.4	Uso de ferramentas de automatização de testes	26
4.6	Investimento em Teste	27
5	CONSIDERAÇÕES FINAIS	29

REFERÊNCIAS 30

—

1 Introdução

Com a globalização, os softwares estão modificando o cotidiano das pessoas e seu uso pode ser percebido nos celulares, bancos, cirurgias, aviões, entre outros (ALMEIDA, 2010). Como consequência de sua utilização e das novas tecnologias, os softwares estão evoluindo cada vez mais rápido sua complexidade para atender às necessidades dos clientes, que exigem que os produtos solicitados sejam entregues com boa qualidade (MORAES; SILVA; VASCONCELLOS, 2014).

Aqueles que não funcionam corretamente podem trazer vários prejuízos tanto para as empresas que os desenvolvem como para os usuários finais. Para evitar estes problemas, é preciso que as empresas invistam em testes no ambiente de desenvolvimento pois, quanto mais cedo os erros forem descobertos, menor será o custo de sua respectiva correção (ALMEIDA, 2010).

1.1 Um pouco de História

Para dar conta da demanda crescente de produtos de software, foram criadas diversos modelos e abordagens para a criação de sistemas e aplicações (MAHESHWARI; JAIN, 2012). Os primeiros modelos de processo propunham que a construção de sistemas de software acontecesse de acordo com a aplicação sistemática de princípios de Engenharia de Software, incluindo as áreas de Qualidade e Engenharia de Sistemas. Os métodos utilizados nestes processos davam ênfase às atividades das fases iniciais do projeto: entendimento de requisitos, planejamento e elaboração de arquiteturas bem estabilizadas, fartamente documentadas em contratos e especificações (SOMMERVILLE, 2011; PRES-SMAN, 2006).

Os modelos subsequentes evoluíram para dar suporte às alterações encontradas no ambiente de negócios que também se diversificara. Isso foi o bastante para impelir o mercado a demandar maiores esforços de desenvolvimento de software para atender suas necessidades, o que passou a exigir que fronteiras organizacionais e geográficas fossem ultrapassadas.

Os gestores precisaram ser criativos para encontrar formas efetivas de orquestração de competências dos mais variados perfis além de se adequarem a cronogramas restritos para atender expectativas agressivas dos clientes. O mercado de Tecnologia da Informação também precisou se adaptar ao uso de variadas tendências e tecnologias, como sistemas distribuídos, equipes transdisciplinares, desenvolvimento concorrente, aplicações open source, utilização de processos ágeis e aderência a padrões e modelos internacionais

de qualidade (BOEHM; BASILI, 2001; BOEHM, 1988).

A nova dinâmica dos projetos de desenvolvimento de software evidenciou que os profissionais de Tecnologia da Informação precisam abstrair e apreender, em pouco tempo, grandes volumes de informações, modelos, especificações e padrões. As pressões para entregar o software de acordo com parâmetros de custo e qualidade restritivos, aliadas à necessidade de processar esta grande quantidade de informações contribui para falhas de planejamento que podem resultar em compressões inadequadas do cronograma, falhas de comunicação e retrabalho (KULIK, 2000; BOEHM; HANSEN, 2000; BOEHM; BASILI, 2001).

Apesar disso, o mercado e os consumidores de produtos de Tecnologia da Informação permanecem fiéis a uma das mais elementares expectativas do ser humano, que é busca pela qualidade. O teste de software é uma das atividades que procuram contribuir para a melhoria da qualidade dos sistemas desenvolvidos em geral. É a adoção de práticas de teste que garante que um determinado sistema esteja sendo desenvolvido de acordo com os requisitos especificados.

O teste de software consiste no processo de execução de um produto para verificar se ele está de acordo com as especificações e se está funcionando sem erros. As equipes de testes tem como objetivo identificar as falhas do produto, para que possam ser corrigidas pela equipe de desenvolvimento da empresa antes de fazer a entrega final, visando o aumento da confiança do produto fornecido ao cliente (CLAUDIO, 2008; CHEON; LEAVENS, 2002).

No processo de implementação, todos os erros são humanos e, mesmo utilizando excelentes métodos, as falhas ainda continuam sendo verificadas nos produtos, o que torna a execução dos testes essencial para o desenvolvimento do software. Fatores como o tamanho do projeto e a quantidade de pessoas alocadas para executar as tarefas devem ser levados em conta quando se está desenvolvendo o produto pois, quanto mais pessoas envolvidas e quanto maior o projeto, maior será a complexidade desse trabalho e, consequentemente, maior será a possibilidade de surgirem mais problemas (CLAUDIO, 2008; PRESSMAN, 2006).

1.2 Contexto

Estudos apontam que de 40% a 50% do esforço de um projeto é gasto com retrabalho que poderia ser evitado (BOEHM; BASILI, 2001). Além disso, o custo para detectar e corrigir defeitos cresce à medida que eles são propagados para fases posteriores do ciclo de vida de desenvolvimento, o que exige a elaboração e adoção de estratégias que resolvam ou, pelo menos, minimizem essas questões.

A evolução e utilização intensiva de técnicas de teste associada à especialização e capacitação de profissionais específicos para utilizá-las é crucial para o processo de desenvolvimento de software porque permite que os desenvolvedores e projetistas possam se concentrar no correto entendimento, modelagem e codificação das regras de negócio, deixando as etapas de verificação sucessiva e de validação final dos produtos com os profissionais de teste e qualidade (DURAN; NTAFOSS, 1984; NTAFOSS, 2001; CHEN; YU, 1996).

Essa especialização de funções vai ao encontro tanto das necessidades de mercado quanto da consecução do principal objetivo da engenharia de software que é a aplicação de teoria, modelos, formalismos, técnicas e ferramentas da ciência da computação e áreas afins para o desenvolvimento de software (BUSE; ZIMMERMANN, 2010). Como a qualidade de um produto de software relaciona-se diretamente à sua quantidade de defeitos, estes devem ser detectados o mais cedo possível para evitar retrabalho (BOEHM; HANSEN, 2000; BOEHM; BASILI, 2001). Várias técnicas de teste podem ser utilizadas no ciclo de desenvolvimento de software, porém neste estudo, são abordados os testes de regressão.

Vários métodos e tipos de testes podem ser aplicados pela equipe de testes durante o processo, mas é essencial que se tenha toda a documentação com os requisitos atualizados para que o sistema tenha qualidade e seja desenvolvido conforme o cliente solicitou (MORAES; SILVA; VASCONCELLOS, 2014; CHEON; LEAVENS, 2002; LEDRU et al., 2001).

Um destes tipos é o teste de regressão, que tem papel crucial na garantia da qualidade do produto final, pois eles são realizados para identificar os efeitos negativos de mudanças feitas no software, envolvendo desde a criação de novos defeitos até redução da confiabilidade, desempenho ou funcionalidade do produto (TÉCNICAS, 2004; PRES-SMAN, 2005). Testes de regressão são complexos e podem ser ainda mais desafiadores quando realizados em unidades organizacionais que adotam práticas baseadas em algumas das recentes tendências de desenvolvimento de software (LEDRU et al., 2001).

Por exemplo, o desenvolvimento baseado em componentes que tende a utilizar muitos componentes de terceiros (SOMMERVILLE, 2011). Quaisquer alterações nestes componentes podem interferir em outras partes do software, no entanto, é difícil de realizar testes de regressão em componentes de terceiros, porque sua lógica interna não é conhecida pelos programadores que os utilizam. Além disso, quanto mais curto for o ciclo de vida do desenvolvimento de software, como o sugerido pelos métodos ágeis, maiores serão as restrições e limitações sobre como testes de regressão podem ser realizados com recursos limitados (BECK, 2002).

O contexto atual da área de testes de software remete a práticas ainda pouco padronizadas e utilizadas apenas sob demanda. Em empresas de desenvolvimento de software é comum que parte dos profissionais que atuam com qualidade de software (realização dos

testes) sejam alocados em tempo parcial em projetos nos quais não atuam na realização de sua atividade principal em períodos em que existe baixa demanda de testes de software.

Numa época em que as atividades de teste estão em alta, percebe-se claramente que elas não são avaliadas de acordo com critérios únicos e objetivos. Isso gera subjetivismos e não dá margem para questionamentos referentes à qualidade objetiva das atividades de teste. Ou seja, cada projeto é avaliado de uma maneira, pois não existem mecanismos que permitam medir o atraso na entrega de alguma atividade de teste, por exemplo, ou a taxa de retrabalho originada por sucessivas entregas que apresentem erros ou defeitos não identificados em testes de regressão.

No entanto, mesmo diante de indicações claras de que os testes regressivos sejam essenciais para a qualidade, existe a suspeita de que empresas desenvolvedoras não testem regressivamente seus produtos por considerarem que isso seja dispensável, desde que o software tenha sido submetido pelo menos a uma bateria completa de testes. As razões pelas quais isso acontece são discutidas na apresentação dos resultados da pesquisa, no entanto, identificou-se que a causa raiz do problema é o baixo nível de investimento feito pelas empresas em testes devido à falta de percepção da real importância desta atividade.

1.3 Objetivo

O objetivo desta pesquisa é investigar a relevância, percebida pelas empresas de desenvolvimento de software da região do Triângulo Mineiro, das práticas de teste de regressão e também seu nível de investimento nestas atividades. A questão fundamental é: qual é a relevância dos testes de regressão e quanto é investido em testes de regressão pelas empresas de desenvolvimento de software?

Dessa forma, a realização desta pesquisa visa contribuir para o preenchimento de uma lacuna da área de teste de software na região de Uberlândia o que dificulta a tomada de decisões e a melhoria de parâmetros do projeto como, por exemplo, diminuição da taxa de retrabalho causado pelos sucessivos ciclos de teste. Isso acontece porque, na medida em que não se sabe qual é a taxa real de retrabalho, não é possível estabelecer uma meta efetiva para sua diminuição. Em outras palavras: será que o retrabalho está tão baixo que não se justifica gastar tempo e dinheiro com iniciativas para diminuí-lo?

1.4 Hipótese

A hipótese construída sobre esta questão é que as empresas consideram os testes de regressão relevantes, porém não investem suficientemente na área porque existe a crença de que a realização de uma bateria completa de testes é suficiente para assegurar

que o software não apresentará falhas, mesmo que as mudanças tenham sido feitas em componentes aparentemente desconexos.

1.5 Organização do Trabalho

Para cumprir seus objetivos, organizar o conteúdo e facilitar sua compreensão pelos leitores, o trabalho foi organizado em cinco capítulos, sendo eles:

- Capítulo Um: O primeiro capítulo corresponde à introdução, em que são apresentados o contexto, o problema, a motivação, a justificativa e os objetivos da pesquisa.
- Capítulo Dois: O segundo capítulo é destinado a apresentar os conceitos, fundamentos essenciais à compreensão da pesquisa e trabalhos relacionados à pesquisa.
- Capítulo Três: O terceiro capítulo descreve a metodologia utilizada, envolvendo tanto os procedimentos para obtenção dos dados quanto para analisá-los, além das variáveis de interesse para a pesquisa.
- Capítulo Quatro: O quarto capítulo é responsável por apresentar os resultados da análise dos dados referentes às variáveis de interesse.
- Capítulo Cinco: Por fim, o quinto capítulo envolve as considerações finais, conclui o trabalho, apresenta algumas questões em aberto e aponta direções para pesquisas futuras.

2 Referencial Teórico

Este capítulo apresenta os fundamentos e conceitos básicos utilizados pela disciplina de testes, sobretudo no que concerne aos testes de regressão e respectivas técnicas de minimização, seleção e priorização.

2.1 Teste de Software

O teste mostra que o programa faz o que foi destinado a fazer e também serve para descobrir as falhas antes do uso. Os testes são feitos utilizando dados fictícios e os resultados servem para verificar se há erros, anormalidades no programa.

O conceito de teste de software não é tão intuitivo como pode parecer. Pesquisadores de engenharia de software não chegam a divergir sobre este conceito, porém cada qual define teste de software com nuances diferenciados. (HETZEL, 1973) afirma que “testar é o processo de certificar que o programa faz o que era suposto fazer”. (MYERS, 1979) por outro lado afirma que “testar é o processo de executar um programa ou sistema com objetivo de encontrar erros”(MYERS, 1979). Por fim, (PRESSMAN, 2005) ensina que “o teste de software é um elemento crítico da garantia de qualidade de software e representa a última revisão da especificação, do projeto e da codificação”.

As definições destes três autores guiam a criação da base conceitual deste trabalho ao delinear os elementos necessários ao desenvolvimento de software de qualidade, deixando claro quais objetivos devem ser perseguidos pela equipe do projeto e qual é a real importância da execução de testes relevantes nos produtos de trabalho construídos ao longo do ciclo.

Há no processo de teste dois objetivos diferentes. O primeiro objetivo consiste em mostrar ao desenvolvedor e ao cliente que o programa atende aos requisitos. Se for um software customizado, deve haver pelo menos um teste para cada requisito do documento de especificação e se for um software genérico, deve haver testes para todas as particularidades do sistema. Este objetivo leva a testes de validação e são utilizados casos de testes para garantir que o sistema execute corretamente. O segundo objetivo consiste em descrever os casos em que o sistema pode se comportar de maneira distinta das especificações. Este objetivo leva a testes de defeitos e é utilizado casos de testes para expor as falhas do sistema. Não há limites estabelecido entre estas abordagens, visto que ao longo da execução dos testes de validação podem-se encontrar falhas do sistema e durante os testes de defeitos pode ser mostrado que o programa está de acordo com os requisitos.

De acordo com Boehm (1979), pioneiro da engenharia de software, o objetivo do

processo de verificação é checar e se o software está de acordo com as suas especificações funcionais e não funcionais. Já o processo de validação tem como objetivo garantir que o software desenvolvido para o cliente atenda às suas expectativas.

Os testes não podem comprovar que o software está livre de erros ou se irá se comportar de acordo com as especificações em qualquer cenário pois, um teste esquecido, pode ser aquele que mostraria falhas no sistema. Segundo [Dijkstra, Dahl e Hoare \(1972\)](#) “Os testes podem mostrar apenas a presença de erros, e não sua ausência”. De modo geral, há três estágios que o sistema de software comercial deve passar. São eles:

1. **Testes em Desenvolvimento:** Os testes são feitos durante o desenvolvimento do software em busca de bugs e falhas e os desenvolvedores podem estar envolvidos neste processo.
2. **Teste de *release*:** Uma equipe de testes independente testa a versão completa do sistema antes de entregá-lo aos usuários e também verifica se o software contempla todos os requisitos especificados pelos stakeholders de sistema.
3. **Teste de usuário:** Os usuários ou potenciais usuários testam o sistema e, para produtos de software, estes usuários podem ser um grupo de marketing interno, que determinará se o produto está pronto ou não para ser comercializado. Outro tipo de teste de usuário é o teste de aceitação em que o sistema é testado formalmente pelo cliente para que ele decida se deve ser aceito por parte do fornecedor do sistema ou se um desenvolvimento adicional será necessário.

Na prática, em geral, são envolvidos testes manuais e automatizados no processo de teste. No teste manual o programa é executado por um testador, que possui dados de teste, e compara com o resultado esperado. Se há problemas, ele faz suas anotações e reporta os defeitos para os desenvolvedores.

Já nos testes automatizados, os testes são codificados em um programa e o mesmo é executado cada vez que software desenvolvido é testado. Os testes automatizados são mais rápidos que os manuais, principalmente quando abrange os chamados testes de regressão, que consistem na reexecução de testes anteriores em busca de defeitos após as alterações no sistema.

Nos últimos anos a utilização de testes automatizados tem aumentado, porém, eles só podem testar se o sistema faz o que a ele é proposto. Estes testes não podem ser totalmente automatizados pois, na prática, é impossível utilizar esse tipo de teste para testar se um sistema não tem efeitos colaterais indesejados.

2.2 Métricas e Indicadores para Teste

Métricas de software são padrões quantitativos de medidas de vários aspectos do projeto de software. A medição dessas métricas em um projeto de software pode apoiar estimativas, o controle de qualidade, a produtividade da equipe e o controle do projeto (PRESSMAN, 2006).

Além disso, um conjunto de métricas bem projetado pode ser utilizado para medir a qualidade de produtos de trabalho, dar suporte à tomada de decisão dos gerentes de projeto e aumentar o retorno de investimento do produto (KULIK, 2000).

Em métodos ágeis de desenvolvimento, a utilização de métricas apoia a medição contínua do produto e do processo, permitindo que os ciclos de desenvolvimento sejam constantemente inspecionados, adaptados e melhorados (HARTMANN; DYMOND, 2006).

Diversas características do software e de projetos de software podem ser medidas, como por exemplo, o seu tamanho, complexidade, qualidade, confiabilidade e aderência ao processo. Medidas, métricas e indicadores são conceitos importantes na área de métricas de software e muitas vezes são confundidos. Ragland (1995) descreve esses conceitos a partir de definições da IEEE¹ e SEI²:

1. **Medida:** uma avaliação comparada a um padrão, ela fornece uma indicação quantitativa da extensão, quantidade, dimensão, capacidade ou tamanho de algum atributo do processo ou do produto. Exemplo - linhas de código, que podem variar conforme a linguagem ou regras para o cálculo de linhas de código.
2. **Métrica:** um método quantitativo para determinar um certo atributo em relação ao sistema, componente ou processo. Ela geralmente é calculada ou composta por duas ou mais medidas. Exemplo - quantidade de casos de teste produzidos em uma iteração. As medidas que compõem essa métrica são o número de casos de teste produzidos e a iteração em que esses casos de teste foram produzidos.
3. **Indicador:** um aparelho ou variável que pode ser configurado para um determinado estado conforme a ocorrência de um processo ou a ocorrência de uma situação especificada. Um indicador é algo que chama a atenção de uma pessoa para uma situação particular e pode considerar um comparativo entre métricas. Exemplo - aumento substancial de defeitos encontrados na última versão, que pode ser um indicador de que a qualidade do software piorou. Indicadores permitem avaliar o status de um projeto em andamento, acompanhar riscos em potencial, descobrir

¹ Institute of Electrical and Electronics Engineers.

² Software Engineering Institute.

áreas-problema, ajustar o fluxo de trabalho ou tarefas e avaliar a capacidade da equipe de controlar a qualidade dos produtos produzidos (KAN, 2002).

Métricas de software podem ser classificadas em três categorias: métricas de processo, métricas de projeto e métricas de produto (KAN, 2002). Métricas do processo são utilizadas para a melhoria no desenvolvimento do software e manutenção. Exemplos de métricas de processo incluem a taxa de descoberta de defeitos, ou o tempo gasto para consertar defeitos encontrados.

Métricas de projeto descrevem características do projeto e sua execução, e são utilizadas para minimizar atrasos, riscos e para avaliar a qualidade durante o desenvolvimento. Exemplos de métricas de projeto incluem o número de desenvolvedores, custos, cronograma e produtividade. Métricas de produto descrevem características do produto como tamanho, complexidade, características ligado a projeto do código, performance e nível de qualidade.

As métricas de qualidade de software são um subconjunto de métricas que focam em aspectos de qualidade do produto, processo e projeto. Essas métricas podem ser divididas em métricas de qualidade do produto final ou métricas de qualidade calculadas durante todo o ciclo de desenvolvimento (chamadas de in-process metrics). Segundo Kan (2002), a essência da engenharia de qualidade de software é investigar as relações entre as métricas in-process, características do projeto, e a qualidade do produto final, e baseado nessas descobertas, melhorar a qualidade do processo e do produto.

2.3 Teste Regressivo

De acordo com (BASTOS et al., 2007) a realização de testes regressivos envolve re-testar “segmentos já testados após a realização de uma mudança em uma outra parte do software”(BASTOS et al., 2007), o que vai ao encontro das definições de (HETZEL, 1973), (MYERS, 1979) e (PRESSMAN, 2005) que são unânimes ao afirmar que os testes de regressão visam garantir a integridade do software após a realização de modificações.

2.3.1 Reteste Total

O reteste total se baseia na re-execução de todos os casos de teste especificados para o software. Esta é a técnica que exige mais tempo e esforço para ser executada, visto que, segundo (ROTHERMEL et al., 2004), existe a necessidade de reexecutar todos os cenários de teste previamente executados.

Esta é a abordagem mais simples para resolver o problema. No entanto, como o software evolui, o tamanho do conjunto de teste tende a crescer, o que significa que pode ser proibitivamente caro reexecutar toda a suíte de testes. Esta limitação exige que outras

técnicas que reduzam o esforço necessário para o ensaio de regressão sejam consideradas. Importante ressaltar também que casos de teste obsoletos ou desatualizados devem ser refeitos, adequando-os às novas especificações do sistema, ou até mesmo descartados, caso não reflitam mais o comportamento desejado pelo cliente.

2.3.2 Seleção de Testes para Regressão

A seleção de testes para regressão consiste no reuso dos casos de teste, porém de forma seletiva. Esta técnica pode apresentar um bom custo-benefício, levando em consideração que não serão refeitos todos os testes, logo o tempo gasto será menor. No entanto, é essencial que os critérios de seleção assegurem a cobertura das funcionalidades mais relevantes.

Além disso, a seleção dos testes de regressão deve garantir que casos de testes que não foram selecionados não revelem falhas na nova versão. Uma das maneiras de assegurar isso seria executando um grande número de casos de teste, mas isso faz com que a técnica ganhe em segurança, mas perca eficiência.

Achar um ponto de equilíbrio entre estes fatores é o principal desafio desta técnica. Alguns estudos, como o de (ROTHERMEL et al., 2004), apontam que esta técnica tem um bom custo-benefício.

2.3.3 Redução da Suite de Teste

A redução da suíte de testes envolve a exclusão de parte do conjunto dos casos de teste da suíte de forma permanente. A porção do conjunto que é excluída corresponde aos casos de teste mais antigos, porque na medida em que o software evolui, os novos casos de teste criados podem gerar redundância no conjunto original.

Deve-se ressaltar que a redução da suíte de teste pode economizar custo de detecção de falhas, porém ela pode igualmente reduzir a eficácia dos testes. Apesar disso, alguns estudos indicam que esta é uma técnica que gera economia no custo da detecção de falhas de acordo com (WONG et al., 1998).

2.3.4 Priorização de Suite de Teste

A técnica de priorização de casos de teste envolve a ordenação dos casos de teste de acordo com a sua ordem de prioridade para o negócio. Assim, aqueles com maior prioridade serão executados antes daqueles de menor prioridade. Exemplos de critérios de priorização são cobertura de código, tempo de execução, funcionalidades mais utilizadas e probabilidade de detecção de erro.

Ainda segundo (ROTHERMEL; HARROLD, 1994), (ROTHERMEL; HARROLD, 1998), quando o tempo necessário para executar a totalidade de uma suíte de testes for pequeno, o trabalho de priorização pode não ser interessante por não trazer uma boa relação de custo-benefício, visto que os casos podem ser executados em qualquer ordem (ROTHERMEL; HARROLD, 1994). Por outro lado, quando a execução desta suíte demandar muito tempo, a priorização é benéfica porque os objetivos prioritários do teste podem ser atingidos mais rápido.

Como a técnica de priorização de casos de teste não rejeita casos de testes, ela possui vantagens sobre as técnicas de seleção de testes de regressão e redução da suíte de testes; o que contribui para reduzir os riscos de perdas na cobertura devido à remoção dos casos.

2.4 Vantagens e Desafios dos Testes de Regressão

Em situações em que a rejeição dos casos de testes é aceitável, é possível utilizar a técnica de priorização juntamente com as demais técnicas, para dar prioridade aos casos de teste da suíte de teste selecionada ou minimizada. Feita a escolha, decidem-se como os testes de regressão serão executados.

Isso pode acontecer de forma manual ou automatizada. A execução manual acontece quando o trabalho de teste regressivo é feito utilizando-se apenas o trabalho do analista de teste, sem suporte de ferramentas de automatização. Para isso, o analista escolhe a técnica que será usada, identifica o escopo dos testes, planeja e executa os casos de teste regressivos selecionados.

A execução automatizada também é feita pelo analista de testes, porém, com suporte de ferramentas de automatização, o que exige que ferramentas adequadas sejam adquiridas e institucionalizadas. Para que esta forma de execução de testes de regressão gere benefícios perceptíveis, o esforço de institucionalização da ferramenta também deve considerar os custos decorrentes de contratação ou capacitação de profissionais para que ela possa ser corretamente executada.

Como nos testes manuais, a execução dos testes automatizados pressupõe que o analista selecione a técnica, identifique e planeje a execução dos casos de teste. Feito isso, são criados os scripts de automatização, os quais serão executados pela ferramenta de teste. Uma vez criados o script dos casos de teste, estes poderão ser executados tantas vezes quantas forem necessárias, de forma bem mais rápida do que manualmente.

Independentemente da técnica ou da forma de realização dos testes de regressão, sua adoção dá origem tanto a vantagens quanto a desafios para a organização, os quais são sintetizados na Tabela 1.

Tabela 1 – Vantagens e desafios da utilização de testes de regressão

Vantanges	Desafios
Melhoria da confiança e segurança no que foi desenvolvido e/ou alterado	Necessidade de desembolso financeiro para custear treinamentos e aquisição de ferramentas
Melhorar a previsibilidade no lançamento das versões ou do próprio sistema	Aumento do esforço e da duração dos projetos devido realização dos testes de regressão
Avaliação objetiva a qualidade do que está sendo desenvolvido	Investimento na estruturação de equipe de automatização de testes
Redução de gastos com manutenções	

À luz dos conceitos e fundamentos apresentados, a próxima seção deste trabalho apresenta a metodologia utilizada para coletar e analisar os dados necessários para determinar a relevância e os níveis de investimento aceitáveis para as empresas consideradas na pesquisa.

3 Metodologia

De acordo com a classificação metodológica apresentada por (GIL, 2010) e (YIN, 2010), esta pesquisa tem caráter exploratório com relação aos objetivos, quantitativa com relação à abordagem do problema e estudo de caso quanto aos procedimentos adotados.

Primeiramente foi realizado um levantamento bibliográfico acerca de testes de regressão. No segundo momento, foi elaborado um questionário que foi enviado para um conjunto de 30 PMEs (Pequenas e Médias Empresas) de desenvolvimento de software da região do Triângulo Mineiro, abrangendo as cidades de Uberlândia, Uberaba, Ituiutaba, Patrocínio, Patos de Minas e Monte Carmelo, todas no estado de Minas Gerais. Para delimitar o escopo de PME foi utilizado o conceito do SEBRAE¹ como descrito em (SIGNIFICADOS, 2013).

As empresas foram selecionadas após consultas realizadas junto a universidades, Centro de Diretores Lojistas (CDL) e prefeituras municipais destas cidades. A região do Triângulo Mineiro, sobretudo a cidade de Uberlândia, foi selecionada por ser residência da autora da pesquisa, o que facilita os contatos.

Os questionários foram enviados por e-mail para as 30 empresas, e destas, 17 o responderam, o que representa uma taxa de retorno de 56,67%. As empresas que responderam situam-se nas cidades de Uberlândia, Uberaba e Patrocínio, com percentuais de 76,47%, 17,65% e 5,88%, respectivamente.

As próximas seções apresentam a sistemática de elaboração do questionário, da definição das variáveis de interesse do método de análise dos resultados.

3.1 Questionário

O questionário foi elaborado utilizando Microsoft Office Word 2010 e estruturado em três partes, sendo que a primeira parte contém um quadro de identificação da empresa, a segunda é composta por sete questões destinadas a delinear características gerais da empresa e a terceira parte conta com 18 questões específicas de teste. Todas as questões da primeira e segunda parte são de múltipla escolha.

As questões da primeira parte abrangem 1) Tempo da empresa no mercado; 2) Quantidade de Funcionários; 3) Quantidade de projetos que podem ser desenvolvidos simultaneamente; 4) Regiões em que a empresa presta serviços; 5) Quantidade de funcionários exclusivos para teste; 6) Nível de investimento em testes e qualidade e 7) Principais linguagens de desenvolvimento.

¹ Serviço Brasileiro de Apoio às Micro e Pequenas Empresas

O mesmo padrão de questões foi adotado para elaborar as questões da segunda parte do questionário, as quais se referem a 8) Frequência de execução de testes; 9) Presença de equipes de teste; 10) Frequência com que os testes são reexecutados; 11) Importância dos testes de regressão para a empresa; 12) Realização de testes automatizados; 13) Forma de execução dos testes de regressão; 14) Responsável pelos testes de regressão; 15) Caracterização da gestão dos testes; 16) Fase do ciclo de vida em que é encontrada a maior quantidade de falhas no software; 17) Justificativa para executar testes de regressão; 18) Percentual de esforço do projeto dedicado a testes; 19) Investimento que a empresa estaria disposta a fazer em ferramentas de teste; 20) Abordagem de teste regressivo adotada pela empresa; 21) Estratégia de gestão de falhas de software; 22) Documentos de teste elaborados pela empresa; 23) Frequência de geração de relatórios de qualidade; 24) Atributos de qualidade mais relevantes e 25) Papel da atividade de teste para a empresa.

Antes de enviar o questionário para as empresas ele foi submetido a duas sessões de críticas, sendo que todas as observações pertinentes foram acatadas. A coleta de dados aconteceu nos meses de março e abril de 2016. É importante ressaltar que foi feito contato inicial, por e-mail, com todas as empresas selecionadas. Neste momento, além do objetivo e do formato do trabalho ser explicado, foi pedido que, caso a empresa se dispusesse a responder o questionário, que isso fosse feito por alguém com profundo conhecimento das práticas e processos da organização.

3.2 Variáveis de Interesse

As variáveis de interesse da pesquisa foram definidas e isoladas para fornecer uma percepção consistente dos significados das atividades de teste de regressão nos moldes propostos por (SILVA, 2006). Para isso construiu-se um modelo ontológico da utilização de testes regressivos nas empresas para explicitar os pressupostos deste domínio e separar este conhecimento dos procedimentos operacionais.

Apesar das questões terem sido elaboradas de acordo com um modelo comum, a extração de informação para analisar as variáveis pode ser tão difícil quanto limitada, devido a inconsistências nas terminologias e limitações de busca por palavras chave, respectivamente. As variáveis de interesse da pesquisa são: 1) Perfil das empresas; 2) Características de qualidade de software; 3) Relevância das atividades de testes; 4) Relevância dos testes de regressão; 5) Documentação de testes e 6) Investimento em teste.

3.3 Método de Análise

Após receber os questionários respondidos pelas empresas, os dados foram tabulados e categorizados. A análise quantitativa dos dados foi realizada utilizando estatística

descritiva e foi apoiada pelo Microsoft Office Excel 2010, no qual os gráficos também foram elaborados. A isso sucedeu a tarefa de análise qualitativa dos dados e consolidação dos resultados, os quais são apresentados na próxima seção deste trabalho.

4 Apresentação dos Resultados

Este capítulo apresenta o resultado da análise das variáveis de interesse da pesquisa apresentadas na seção 3. As variáveis são analisadas para determinar sua correlação com a relevância dos testes de regressão para as empresas e o volume médio de investimento que elas estão dispostas a fazer. A análise comparativa foi feita agrupando-se as questões cujos dados se referem às variáveis de interesse.

Variações significativas nos valores das variáveis nem sempre foram descartadas porque estes pontos podem contribuir com o entendimento de aspectos que estejam além dos limites do escopo da pesquisa. Nestes casos, são feitos apontamentos para que não se perca a riqueza das informações coletadas, mesmo que não sejam exaustivamente tratadas. As próximas subseções apresentam os principais pontos de destaque da análise destes resultados.

4.1 Perfil das Empresas

A primeira variável de interesse tratada refere-se ao perfil das empresas pesquisadas. A consolidação dos dados mostrou que 70,59% delas tem mais de sete anos de atuação no mercado e 41,18% contam com mais de 50 funcionários em seu quadro. Além disso, 47,06% das empresas tem capacidade de conduzir pelo menos 10 projetos simultaneamente, podendo ultrapassar 20 em 11,76% das empresas. Das 17 empresas pesquisadas, 82,35% prestam serviços para clientes da região e outros centros nacionais e internacionais, ou seja, apenas 17,65% das fábricas de software têm clientes apenas no Triângulo Mineiro. As principais linguagens de programação das empresas são Java, .NET, PHP e Delphi. Seu percentual de utilização está representado na Figura 1.

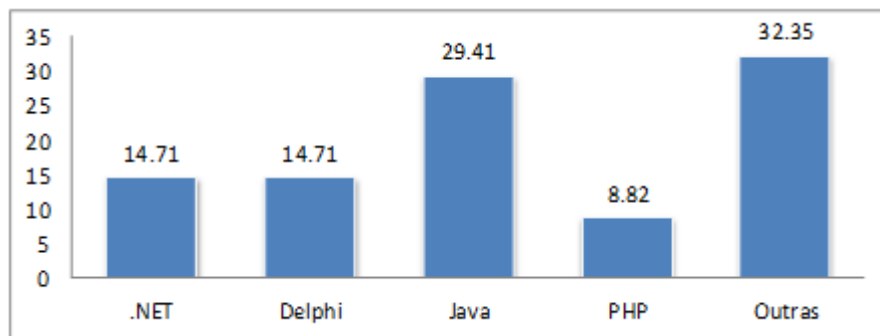


Figura 1 – Percentual de utilização de linguagens de programação pelas empresas

Estes dados indicam que as empresas possuem mercado para os seus produtos e serviços, desenvolvendo projetos em diferentes linguagens. Isso permite que elas se conso-

lidem mesmo estando situadas fora de grandes centros. Além disso, os resultados corroboram a informação de que nos últimos cinco anos Uberlândia se tornou a segunda cidade do interior do país em número de geração de postos de trabalho, conforme noticiado por (TIINSIDE, 2011).

Mesmo as empresas com menor tempo de mercado têm boas perspectivas porque dentre as empresas com mais de sete anos 64,71% têm clientes no Triângulo Mineiro e em pelo menos outro grande centro, como São Paulo, Rio de Janeiro ou Brasília. Isso reforça o perfil de expansão e consolidação das empresas da região, aproveitando-se no potencial universitário da região do Triângulo Mineiro, onde é formada expressiva quantidade de mão-de-obra anualmente.

4.2 Características de Qualidade de Software

Existem diferentes visões da qualidade do produto e de suas métricas em diferentes estágios do ciclo de vida do software. A NBR ISO/IEC 9126-1 descreve seis características de qualidade, sendo que uma delas é a manutenibilidade, que abrange a capacidade do produto de software permitir que, quando modificado, seja validado¹(IEC, 2003). Por isso, a pesquisa identificou também tanto as características de qualidade de software mais relevantes para as empresas quanto as consideradas menos relevantes. As duas características mais relevantes são funcionalidade e confiabilidade para 41,17% e 35,29% das empresas respectivamente. Já as características menos relevantes são portabilidade para 70,59% das empresas e manutenibilidade para outros 23,53%.

Apesar da análise dessa variável da pesquisa ser indicativa de que empresas preferam entregar produtos mais funcionais e confiáveis do que manuteníveis e portáteis, convém destacar que o percentual de irrelevância da portabilidade é três vezes maior que o da capacidade de manutenção, o que parece indicar o aumento da percepção de que os testes são cruciais para a garantia da qualidade dos produtos.

4.3 Relevância das Atividades de Teste

Segundo (PRESSMAN, 2005), é recomendado que o percentual de esforço de teste em um projeto varie no intervalo de 30 a 40%. O expressivo percentual de empresas com mais de sete anos de mercado identificado na subseção que analisou o perfil das empresas sugere que suas práticas de desenvolvimento de software sejam maduras e bem sucedidas, porém os dados da pesquisa indicam uma situação adversa, conforme ilustrado no gráfico da Figura 2.

¹ As características de qualidade consideradas pela NBR ISO/IEC 9126-1 são: Funcionalidade, Confiabilidade, Usabilidade, Eficiência, Manutenibilidade e Portabilidade(IEC, 2003).

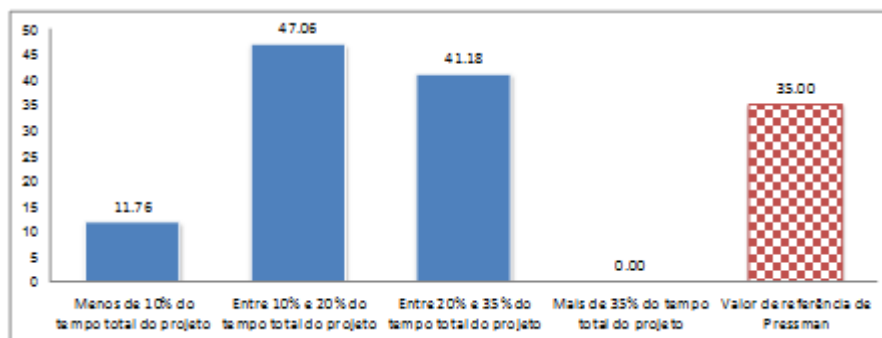


Figura 2 – Percentual de esforço em atividades de teste

Isso significa que 58.82% das empresas dedicam até 20% do esforço de seus projetos com atividades de teste, o que equivale a aproximadamente metade do percentual recomendado por (PRESSMAN, 2005). Outros dados corroboram a hipótese de que a relevância da atividade de teste não é percebida pelas empresas. Ao serem questionadas sobre a importância das atividades de teste para o sucesso do projeto 5,88% declararam que esta é a atividade menos importante, 76,47% apontaram que ela possui a mesma importância das demais e 17,65% disseram que ela é a mais importante.

4.4 Documentação de Teste

A elaboração de documentos específicos de teste é importante, pois 1) Subsidia o gerente em tarefas de planejamento, execução, verificação e ajuste do plano de projeto; 2) Possibilita verificar objetivamente a qualidade dos produtos e identificar pontos de melhoria; e, 3) Fornece informações técnicas essenciais para os profissionais designados para testar e corrigir o software. Por isso, uma das variáveis de interesse da pesquisa é o conjunto de documentos produzido pelas empresas para suportar as atividades de teste.

Diante disso, constatou-se que 70,59% das empresas produzem Planos de Testes, documento que “descreve como o teste deverá ser executado, traça uma linha mestra a ser seguida, e evita que alguns problemas sejam tratados apenas no momento que venham a ocorrer” conforme (BASTOS et al., 2007), e casos de teste, que “é a especificação mais detalhada do teste, estabelecendo quais informações serão empregadas durante os testes dos cenários e quais serão os resultados esperados” (BASTOS et al., 2007; PRESSMAN, 2005).

Dito de outra forma, estas empresas põem em prática as recomendações destes autores, documentando as diretrizes, o planejamento, execução e resultado dos testes realizados.

A pesquisa também constatou que 64,71% das empresas produzem relatórios de falhas, sendo que destas 54,55% os fazem semanalmente. Estes resultados evidenciam que

as empresas efetivamente monitoram e medem objetivamente a qualidade do software em desenvolvimento.

4.5 Relevância dos Testes de Regressão

Os resultados da análise da variável de interesse relevância dos testes de regressão são apresentados em quatro componentes, como se segue.

4.5.1 Percepção da importância dos testes de regressão

Apesar da relevância e do esforço realizado em atividades de teste estar aquém do recomendado de acordo com os resultados apresentados na seção 4.3, quando as empresas foram especificamente questionadas sobre a importância da execução de testes de regressão, o resultado foi diverso, como pode ser visto no gráfico da Figura 3.

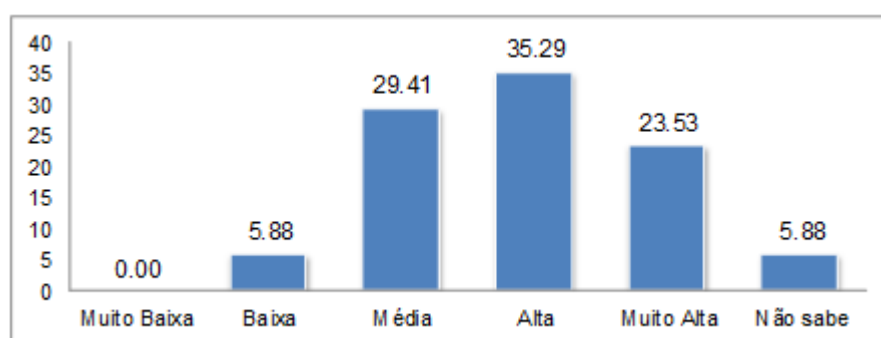


Figura 3 – Importância dos testes de regressão para as empresas

Isso indica que o conceito e a importância dos testes de regressão são claros para a maioria das empresas, pois além de 58,82% afirmarem que a importância desta atividade é alta ou muito alta, 76,47% delas também disseram realizar testes regressivos sempre que alterações são feitas nos sistemas e que a principal justificativa para execução dos testes de regressão em seus sistemas é estabilizar o que foi desenvolvido/corrigido.

4.5.2 Perfil dos profissionais responsáveis pelo planejamento das regressões

Apesar do percentual de empresas que afirmaram realizar testes regressivos não ser desprezível, nota-se que problemas elementares de gestão – como a realização de uma atividade por um profissional não capacitado para tanto –, interferem na sua plena efetivação, pois em 47,05% das empresas o planejamento de testes de regressão fica a cargo de profissionais alheios às práticas de teste. O gráfico da Figura 4 ilustra os percentuais de empresas por perfil de profissional responsável pelo planejamento dos testes.

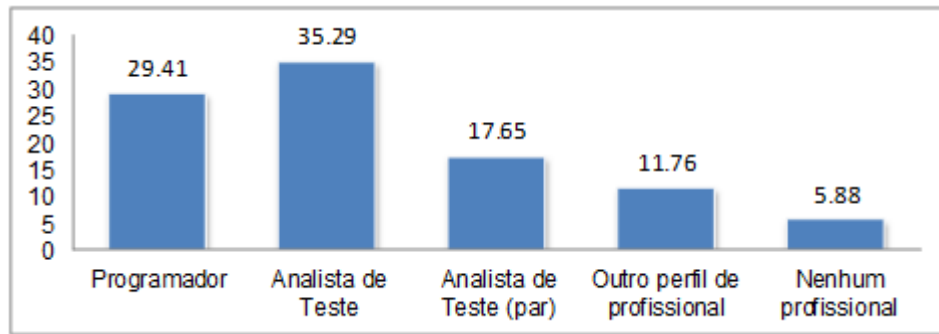


Figura 4 – Profissionais responsáveis pelo planejamento dos testes de regressão

Vale ressaltar que 11,76% das empresas afirmaram que o planejamento é feito por outros tipos de profissionais, sendo que em uma empresa o planejamento é feito por um projetista de teste específico para este fim.

4.5.3 Abordagens de seleção dos casos de teste para regressão

Segundo a classificação dos testes de regressão apresentadas no referencial teórico deste trabalho, as empresas foram questionadas sobre quais delas fazem uso. O resultado para este questionamento pode ser visto na Figura 5.

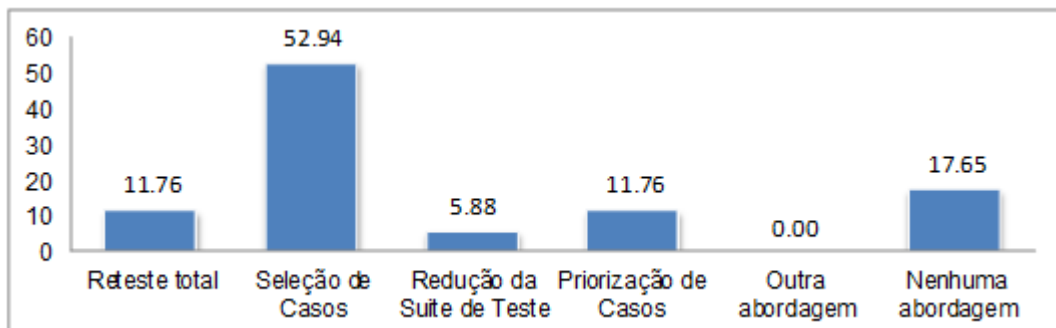


Figura 5 – Abordagens de testes de regressão utilizadas pelas empresas

As empresas também informaram que reusam as suítes de casos de teste de forma seletiva, o que gera economia para as empresas porque os casos de teste não serão refeitos em sua totalidade. Outro dado que merece destaque é que mesmo cientes da importância dos testes de regressão, algumas empresas não fazem uso de nenhuma abordagem específica, o que evidencia um certo distanciamento entre elas e as abordagens teórico-acadêmicas.

4.5.4 Uso de ferramentas de automatização de testes

Considerando que os testes de regressão consistem na re-execução de testes feitos anteriormente, sua automatização é uma boa prática, pois ao reduzir o tempo e o custo

dessa atividade, contribui com o aumento da produtividade. Apesar disso, dentre as empresas pesquisadas apenas 11,76% delas automatizam todos os tipos de teste, inclusive os de regressão e 35,26% selecionam alguns casos de testes para serem automatizados.

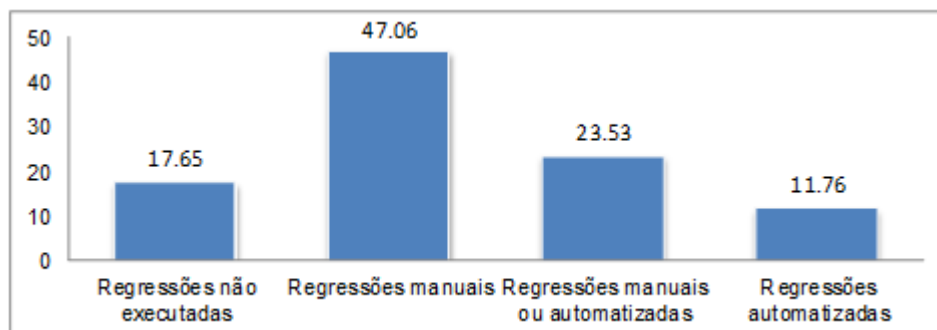


Figura 6 – Forma de execução dos testes automatizados

As formas com que as empresas podem conduzir os testes de regressão são ilustradas pelo gráfico da Figura 6 e ressalta o predomínio dos testes manuais sobre os automatizados.

4.6 Investimento em Teste

Do total de empresas pesquisadas 29,41% delas declararam não contar com profissionais dedicados exclusivamente à área de testes, e outros 35,29% informaram possuir entre 1 e 5 funcionários em seu quadro para desempenhar tais atividades. Além disso, uma empresa em específico citou que os funcionários que tem atividades de teste, possuem outras atividades em primeiro plano. Estes resultados são preocupantes considerando o perfil das empresas, analisado na subseção 4.1. O gráfico da Figura 7 ilustra a distribuição das empresas no que se refere à presença de equipes de teste.

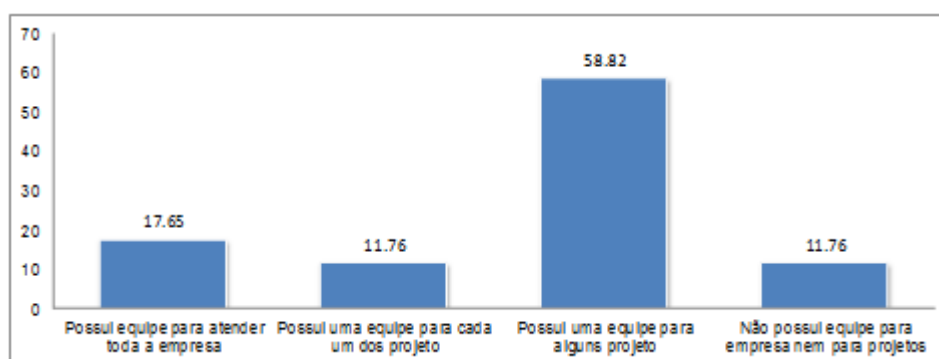


Figura 7 – Tipo e percentual de distribuição de equipes de teste entre as empresas

Os baixos percentuais corroboram com a percepção de que a importância dos testes realmente não é percebida pelas empresas, o que parece ser responsável até mesmo pela

desvalorização dos profissionais da área de qualidade. Apesar disso, 64,71% das empresas responderam que todos os softwares desenvolvidos passam por várias baterias de teste, 29,41% afirmaram que executam uma única bateria de testes e 5,88% selecionam apenas alguns projetos para teste. Isso significa que 35,29% das empresas entregam os produtos finais sem a correta cobertura de testes, o que evidentemente pode gerar problemas para seus clientes e para sua imagem.

Quando questionadas sobre quanto estariam dispostas a gastar com novas ferramentas (ou tecnologias) de teste, 35,29% das empresas responderam que não pretendem gastar nada além do custo de implantação de soluções open source. Outros 47,06% responderam que estão dispostos a gastar até R\$ 3.000,00 (três mil reais), porém a preferência também seria por soluções open source.

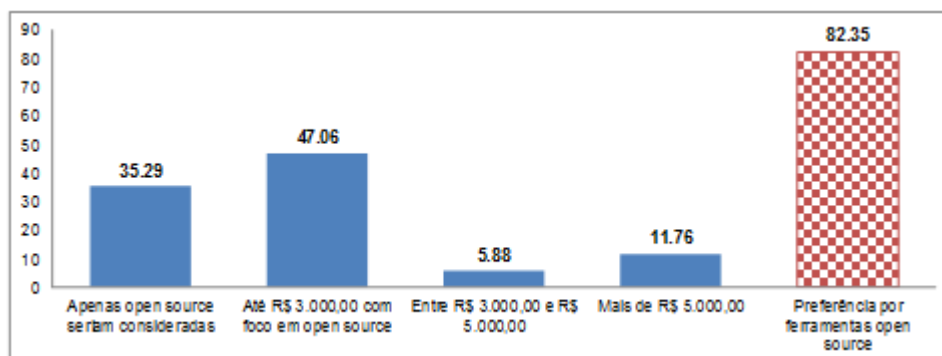


Figura 8 – Investimento em ferramentas de teste por faixa de valores

O gráfico apresentado na Figura 8 mostra o percentual de empresas por faixa de investimento considerado. A coluna com preenchimento quadriculado representa o somatório das duas primeiras colunas e indica que 82,35% das empresas tem preferência pelo uso de ferramentas open source. O uso de soluções open source não implica na ausência total de custos, pois é necessário considerar o tempo de institucionalização da nova ferramenta, que pode afetar negativamente a produtividade, e também o custo de treinamento e capacitação dos funcionários.

Apesar do cenário de baixo investimento efetivo, é importante destacar que 54,83% das empresas pesquisadas admitem que o investimento feito por elas em testes é baixo ou mediano; o que não deixa de ser alentador, pois mesmo aquelas que investem pouco tem consciência de que o investimento realizado é aquém do ideal, apesar de ainda não tomarem ações concretas para alterar tal situação.

5 Considerações Finais

Testes de regressão são essenciais durante o desenvolvimento de um novo sistema, já que visam certificar que alterações não impactaram negativamente no produto que será entregue ao final de cada projeto. Com isso, fica claro que esta atividade tem como objetivo garantir que tudo que está sendo realizado segue as especificações e anseios dos clientes.

Diante disso, neste trabalho foram identificadas as principais e mais conhecidas técnicas para se realizar tal atividade, segundo os principais autores da área. Dessa forma pode-se averiguar como as empresas estão testando seus sistemas, além de como aplicam a teoria em benefício dos seus produtos e serviços.

A pesquisa realizada foi essencial para destacar a relevância dada pelas empresas para as atividades relacionadas a testes de software, e principalmente aos testes de regressão.

Foram também apontadas as vantagens e desvantagens da aplicação desta atividade. Viu-se que ao fazer uso dos testes de regressão, as empresas se beneficiam bastante já que influenciam diretamente na qualidade dos produtos, na confiança que o cliente tem na empresa e na prospecção de futuros projetos, pois ao entregar produtos com alta qualidade, a empresa passa a ser bem vista no mercado.

Com a aplicação do questionário proposto, foi possível ver como as empresas de Uberlândia-MG e região do triângulo mineiro entendem, praticam e trabalham com os testes de regressão. Foi possível verificar que esta é uma atividade reconhecida, de importância comprovada e que se aplicada da forma correta, com base na teoria proposta, gera um ganho relevante e importante aos envolvidos.

Este trabalho não teve como objetivo esgotar todos os conceitos e argumentos disponíveis na literatura sobre o tema proposto, mas sim servir como uma fonte de consulta e/ou um ponto de partida para outros trabalhos que possam abordar de forma mais profunda cada um dos pontos levantados.

Referências

- ALMEIDA, C. Introdução ao teste de software. *Linha de Código*, 2010. Disponível em: <<http://www.linhadecodigo.com.br/artigo/2775/introducao-ao-teste-de-software.aspx>>. Citado na página 7.
- BASTOS, A. et al. *Base de Conhecimento em teste de software*. 2ª edição. ed. [S.l.]: São Paulo: Editora Martins, 2007. Citado 2 vezes nas páginas 15 e 24.
- BECK, K. *Test Driven Development: By Example*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., 2002. Citado na página 9.
- BOEHM, B. W. Software engineering: R & d trends and defense needs. In: (ORG.), P. W. (Ed.). *Software Technology*. [S.l.]: Cambridge, Mass.: MIT Press, 1979. p. 1–9. Citado na página 12.
- BOEHM, B. W. A spiral model of software development and enhancement. *Computer*, v. 21, n. 5, p. 61–72, 5 1988. ISSN 0018-9162. Citado na página 8.
- BOEHM, B. W.; BASILI, V. R. Software defect reduction top 10 list. *Computer*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 34, n. 1, p. 135–137, 1 2001. ISSN 0018-9162. Citado 2 vezes nas páginas 8 e 9.
- BOEHM, B. W.; HANSEN, W. J. *Spiral Development: Experience, Principles, and Refinements*. [S.l.], 2000. Special Report CMU/SEI-2000-SR-008. Citado 2 vezes nas páginas 8 e 9.
- BUSE, R. P.; ZIMMERMANN, T. Analytics for software development. In: *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*. New York, NY, USA: ACM, 2010. (FoSER '10), p. 77–80. ISBN 978-1-4503-0427-6. Disponível em: <<http://research.microsoft.com/pubs/136301/MSR-TR-2010-111.pdf>>. Citado na página 9.
- CHEN, T. Y.; YU, Y.-T. On the expected number of failures detected by subdomain testing and random testing. *IEEE Transactions on Software Engineering*, v. 22, n. 2, p. 109–119, 1996. Citado na página 9.
- CHEON, Y.; LEAVENS, G. T. A simple and practical approach to unit testing: The JML and JUnit way. In: *Boris Magnusson (ed.), 16th European Conference on Object-Oriented Programming — ECOOP'02, Malaga, Spain*. [S.l.]: Springer-Verlag, 2002. (Lecture Notes in Computer Science, 2374), p. 231–255. Citado 2 vezes nas páginas 8 e 9.
- CLAUDIO, A. Artigo engenharia de software - introdução a teste de software. *Canal Engenharia de Software*, 2008. Disponível em: <<http://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-teste-de-sofSilva/8035>>. Citado na página 8.
- DIJKSTRA, E. W.; DAHL, O.-J.; HOARE, C. A. R. *Structured Programming*. [S.l.]: Londres: Academic Press, 1972. Citado na página 13.

- DURAN, J. W.; NTAFOFOS, S. C. An evaluation of random testing. *IEEE Transactions on Software Engineering*, v. 10, n. 4, p. 438–444, 1984. Citado na página 9.
- GIL, A. C. *Métodos e Técnicas de Pesquisa Social*. [S.l.]: São Paulo: Editora Atlas, 2010. Citado na página 19.
- HARTMANN, D.; DYMOND, R. Appropriate agile measurement: using metrics and diagnostics to deliver business value. In: *Agile Conference, 2006*. Minneapolis, MN, USA: IEEE, 2006. p. 126–134. Disponível em: <<http://dx.doi.org/10.1109/agile.2006.17>>. Citado na página 14.
- HETZEL, W. *Program test methods*. [S.l.]: Prentice Hall, 1973. Citado 2 vezes nas páginas 12 e 15.
- IEC, I. *Software engineering - Product quality Part1: Quality model*. [S.l.], 2003. Citado na página 23.
- KAN, S. H. *Metrics and models in software quality engineering*. Boston, MA, USA: Addison Wesley Longman Publishing Co, 2002. Citado na página 15.
- KULIK, P. A practical approach to software metrics. *IT Professional*, v. 2, n. 1, p. 38–42, 2000. Citado 2 vezes nas páginas 8 e 14.
- LEDRU, Y. et al. Test purposes: adapting the notion of specification to testing. In: *Proceedings of the 16th International Conference on Automated Software Engineering, San Diego, november 2001*. [S.l.]: IEEE Computer Society Press, 2001. Citado na página 9.
- MAHESHWARI, S.; JAIN, D. C. A comparative analysis of different types of models in software development life cycle. *International Journal of Advanced Research in Computer Science and Software Engineering*, v. 2, n. 5, p. 285–290, 5 2012. ISSN 2277-128X. Citado na página 7.
- MORAES, E. M. de; SILVA, E. P. da; VASCONCELLOS, F. P. Benefícios da implantação de uma equipe de teste de software em uma empresa desenvolvimento. *Revista Pensar Tecnologia*, v. 3, n. 2, jul 2014. Disponível em: <http://revistapensar.com.br/tecnologia/pasta_upload/artigos/a80.pdf>. Citado 2 vezes nas páginas 7 e 9.
- MYERS, G. J. *The art of software testing*. [S.l.]: John Wiley and Sons, 1979. Citado 2 vezes nas páginas 12 e 15.
- NTAFOS, S. C. On comparisons of random, partition, and proportional partition testing. *IEEE Transactions on Software Engineering*, v. 27, n. 10, p. 949–960, July 2001. Citado na página 9.
- PRESSMAN, R. S. *Engenharia de Software*. [S.l.]: São Paulo: Makron Books, 2005. Citado 5 vezes nas páginas 9, 12, 15, 23 e 24.
- PRESSMAN, R. S. *Engenharia de Software*. 6ª. ed. [S.l.]: São Paulo: McGraw Hill, 2006. Citado 3 vezes nas páginas 7, 8 e 14.
- RAGLAND, B. Measure, metrics or indicator: what's the difference? In: CROSSTALK (Ed.). [S.l.], 1995. v. 8, n. 3. Citado na página 14.

- ROTHERMEL, G. et al. On test suite composition and cost-effective regression testing. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, v. 13, p. 277–331, 2004. Citado 2 vezes nas páginas 15 e 16.
- ROTHERMEL, G.; HARROLD, M. J. A framework for evaluating regression test selection techniques. *Proc. of the 16th Int'l. Conf. on Softw. Eng.*, p. 201–210, 1994. Citado na página 17.
- ROTHERMEL, G.; HARROLD, M. J. Empirical studies of a safe regression test selection technique. *IEEE Transactions on Software Engineering*, v. 24, p. 401 – 419, 1998. Citado na página 17.
- SIGNIFICADOS. Significado de pme. *Portal Significados*, 2013. Disponível em: <<http://www.significados.com.br/pme/>>. Citado na página 19.
- SILVA, M. J. Ontologias& terminologias: Perspectivas da engenharia. *EdV-Linguatca*, 2006. Citado na página 20.
- SOMMERVILLE, I. *Engenharia de Software*. 9ª. ed. [S.l.]: São Paulo: Pearson Prentice Hall, 2011. Citado 2 vezes nas páginas 7 e 9.
- TIINSIDE. Triângulo mineiro ganha polo tecnológico. *Portal TI Inside Online*, 2011. Citado na página 23.
- TÉCNICAS, A. B. de N. *NBR ISO/IEC 9004. Sistemas de gestão da qualidade - Diretrizes para melhorias de desempenho*. [S.l.], 2004. Citado na página 9.
- WONG, W. E. et al. Effect of test set minimization on fault detection effectiveness. *Software: Practice and Experience*, v. 28, p. 347–369, 1998. Citado na página 16.
- YIN, R. K. Estudo de caso - planejamento e métodos. *Bookman Companhia Editorial*, 2010. Citado na página 19.