

UNIVERSIDADE FEDERAL DE PERNAMBUCO
GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA

2008.1



Método de Seleção de Casos de Teste de
Regressão Baseado em Risco
Trabalho de Graduação

Aluna: Aline Cristina Arruda de Medeiros (acam@cin.ufpe.br)
Orientador: Alexandre Marcos Lins de Vasconcelos (amlv@cin.ufpe.br)

Recife, Junho de 2008

UNIVERSIDADE FEDERAL DE PERNAMBUCO
GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA

2008.1

Método de Seleção de Casos de Teste de
Regressão Baseado em Risco
Trabalho de Graduação

Monografia apresentada ao Centro de Informática da
Universidade Federal de Pernambuco, como requisito parcial
para obtenção do Grau de Engenharia da Computação.

Aluna: Aline Cristina Arruda de Medeiros (acam@cin.ufpe.br)
Orientador: Alexandre Marcos Lins de Vasconcelos (amlv@cin.ufpe.br)

Assinaturas

Este Trabalho de Graduação é resultado dos esforços da aluna Aline Cristina Arruda de Medeiros, sob a orientação do professor Alexandre Marcos Lins de Vasconcelos, sob o título de “Método de Seleção de Casos de Teste de Regressão Baseado em Risco”. Todos abaixo estão de acordo com o conteúdo deste documento e os resultados deste Trabalho de Graduação.

Aline Cristina Arruda de Medeiros

Alexandre Marcos Lins de Vasconcelos

Agradecimentos

Agradeço primeiramente a Deus, por ter guiado meus caminhos para que eu pudesse viver este momento.

A meu orientador Alexandre Vasconcelos, pela oportunidade da realização deste trabalho de graduação e a Elifrancis Soares, por todo seu apoio e confiança depositada em mim.

Agradeço também a toda minha família, em especial aos meus pais, por serem exemplos de esforço e dedicação ao trabalho, por terem me dado uma ótima base educacional e por terem me ensinado a ser uma pessoa com valores e princípios que me acompanharão para o resto de minha vida. E um agradecimento especial também às minhas irmãs, que estiveram sempre ao meu lado, compartilhando momentos bons e ruins.

Não poderia deixar de agradecer aos meus companheiros e amigos que fiz na universidade, com os quais tive a oportunidade de compartilhar cinco anos de muito esforço e dedicação. Um agradecimento em especial à turma de Engenharia, por transformarem nossos momentos de estudos em uma verdadeira diversão!

Às minhas amigas do Colégio Damas, Carlinha, Mari, Rafidows e Roby, por estarem presentes em todos os momentos de minha vida, desde a infância, pelas maravilhosas conversas sobre a vida, pelos conselhos, e por me conhecerem às vezes até mais do que eu mesma. Tenho sorte por vocês fazerem parte de minha vida.

Aos meus amigos em geral, que souberam entender a minha ausência tantas vezes e que me proporcionaram muitos momentos de felicidade. Em especial, agradeço ao pessoal do AJACA, por todos os fins

de semanas que estivemos juntos, compartilhando alegrias, filmes, reflexões, fondues, cachaças, carnavais, sushis e Geraldinhos.

A todos os professores e funcionários do CIn, e a Renato Viana e Erick Lopes, por terem sido dois monitores maravilhosos.

Agradeço, enfim, a todas as pessoas que de alguma forma contribuíram e se envolveram na realização deste trabalho e que participaram da minha vida acadêmica, me dando forças e apoio até hoje.

Resumo

Teste de regressão é o processo de validar o software modificado para fornecer a confiança de que as partes mudadas do software se comportem como pretendidas e que as partes inalteradas do software não tenham sido afetadas pela modificação. No entanto, testar completamente todas as possibilidades nem sempre é possível, já que esta tarefa exige um grande esforço e normalmente o cronograma é apertado.

Este trabalho estabelece um método de priorização de casos de teste de regressão, visando definir um escopo menor e mais efetivo, podendo assim diminuir custos e ganhar tempo no ciclo de teste. A efetividade deste método é avaliada através do experimento realizado em uma organização de testes de software.

Palavras-chave

Teste de Software, Teste de Regressão, Seleção de Testes.

SUMÁRIO

1.	INTRODUÇÃO	9
1.1	OBJETIVOS E CONTEXTO	10
1.2	VISÃO GERAL DO DOCUMENTO.....	10
2.	SELEÇÃO DE TESTES: TRABALHOS RELACIONADOS	11
2.1	TÉCNICAS DE SELEÇÃO DE TESTES DE REGRESSÃO	11
2.1.1	<i>Retest All</i>	11
2.1.2	<i>Regression Test Selection</i>	11
2.1.3	<i>Test Suíte Reduction</i>	12
2.1.4	<i>Test Case Priorization</i>	12
2.2	TESTES BASEADOS EM RISCOS	13
2.2.1	<i>Bach</i>	14
2.2.2	<i>Amland</i>	16
2.2.3	<i>Chen</i>	16
2.2.4	<i>Schaefer</i>	17
2.2.5	<i>Gerrard</i>	18
2.2.6	<i>Besson</i>	19
2.3	CONSIDERAÇÕES FINAIS.....	20
3.	MÉTODO PARA SELEÇÃO DE CASOS DE TESTE	21
3.1	PREMISSAS ADOTADAS	21
3.2	PONTUAÇÃO DOS CRITÉRIOS.....	22
3.3	CRITÉRIOS DE PRIORIZAÇÃO	22
3.3.1	<i>Fator de Exposição de Risco</i>	23
3.3.2	<i>Testes Automatizados</i>	28
3.3.3	<i>Tipo de Testes (Testes Positivos ou Negativos)</i>	29
3.4	CRITÉRIOS AUXILIARES.....	30
3.4.1	<i>Tempo de Execução e Avaliação do Teste</i>	30
3.4.2	<i>Recursos Disponíveis</i>	31
3.4.3	<i>Dependência dos Casos de Teste</i>	31
3.5	MÉTODO PROPOSTO	32
3.6	CONSIDERAÇÕES FINAIS.....	37
4.	EXPERIMENTO	38
4.1	CONTEXTO E PROCESSO ATUAL	38
4.2	RELATO DO EXPERIMENTO.....	39
4.3	ANÁLISE DE RESULTADOS.....	40
4.4	CONSIDERAÇÕES FINAIS.....	42
5.	CONCLUSÃO	43
5.1	PRINCIPAIS CONTRIBUIÇÕES	43
5.2	TRABALHOS FUTUROS.....	44
	ANEXO I	45
	REFERÊNCIAS	46

Tabelas

Tabela 1 – Exemplo do cálculo do risco, adaptado de Shaefer [21]	17
Tabela 2 – Exemplo de pontuação da análise de risco x eficiência do teste, adaptada de Gerrard [18].....	19
Tabela 3 – Critérios de Priorização de Casos de Teste.....	22
Tabela 4 – Impacto decorrente da falha de uma função.....	24
Tabela 5 – Percentual de falhas identificadas	25
Tabela 6 – Severidade da falha encontrada.....	26
Tabela 7 – Classificação de $P(f)$ a partir de $F \times S$	26
Tabela 8 – Exemplo de Probabilidade de Severidade por Caso de Teste	27
Tabela 9 – Exemplo do cálculo de Fator de Exposição de Risco	27
Tabela 10 – Pesos para classificação dos testes.....	29
Tabela 11 – Pesos para Tipos de Teste	30
Tabela 12 – Lista de prioridades antes	32
Tabela 13 – Lista de prioridades depois	32
Tabela 14 – Critérios de Priorização e Pesos.....	33
Tabela 15 – Cálculo do Fator de Exposição de Risco	34
Tabela 16 – Cálculo da Prioridade dos Casos de Teste	34
Tabela 17 – Lista de Casos de Teste ordenados por Prioridade	35
Tabela 18 – Tempo médio por CT medido em minutos.....	35
Tabela 19 – Lista de prioridades após tratamento de dependência	37
Tabela 20 – Planilha Eletrônica utilizada em experimento	40
Tabela 21 – Comparação entre a lista de prioridades resultante com os CTs selecionados da feature A no último ciclo de testes	41

1. Introdução

A construção de softwares está se tornando uma tarefa cada vez mais desafiadora, já que sistemas computacionais estão ganhando um papel cada vez mais importante na sociedade. Falhas encontradas em um sistema podem causar danos catastróficos. Com isso, é grande a pressão em cima dos profissionais da área de tecnologia para que os softwares sejam construídos com qualidade.

A atividade de Testes é um componente vital no processo de Qualidade de Software [12]. O SWEBOK [13] define teste como “uma atividade executada para avaliação da qualidade do produto, identificando defeitos e problemas para sua melhoria. É uma verificação dinâmica do comportamento de um programa através de uma seqüência finita de casos de teste, adequadamente retirada de um domínio usualmente infinito de execuções”.

Durante o desenvolvimento de um software, muitas modificações são realizadas por diversas razões, como correção de erros, inclusão de novos requisitos, mudança de ambiente e melhoria de desempenho. Após as mudanças terem sido realizadas, o sistema precisa ser testado para garantir que ele está se comportando como pretendido e que as modificações não causaram nenhum impacto adverso na qualidade do software. Estes testes realizados após as mudanças são chamados de Testes de Regressão [8].

Os testes de regressão são essenciais para garantir a qualidade e a confiabilidade de um sistema. Eles são caros, porém, quando executados, podem reduzir à metade o custo da manutenção do software [10, 11]. Entretanto, os testadores dispõem de pouco tempo para executar os testes de regressão devido à pressão para liberação de um release. Sendo assim, um teste de regressão completo não pode ser sempre realizado durante modificações e atualizações freqüentes de um sistema por exigir uma grande quantidade de esforço e por ocupar uma fração de tempo significativa no ciclo de vida de um software [9].

A organização deve estruturar seu processo de teste de modo que possa entregar o software no tempo e dentro do orçamento, e igualmente satisfaça as exigências do cliente. Uma alternativa é a seleção de testes de regressão, que é a repetição de apenas um subconjunto da suíte de testes inicial. Porém, é preciso ter muito cuidado na priorização dos casos de teste, já que casos de teste que poderiam revelar falhas podem ser rejeitados, reduzindo a eficácia do ciclo de teste.

1.1 Objetivos e Contexto

O objetivo deste trabalho é propor uma técnica de seleção de casos de teste de regressão, a partir de um conjunto de testes já existente, visando diminuir o escopo do ciclo de teste, de forma que os casos de teste mais relevantes sejam priorizados. A priorização é dada de acordo com a pontuação que cada caso de teste recebe, a partir do estabelecimento de alguns critérios. O método aproveita as informações recolhidas em execuções precedentes dos casos de teste para obter a lista de prioridades. Além disso, o julgamento de especialistas também é considerado como uma das principais premissas na técnica descrita neste documento. A efetividade deste método é avaliada através do experimento realizado em uma organização de testes de software.

1.2 Visão Geral do Documento

Este documento encontra-se estruturado em cinco capítulos, sendo este o Capítulo 1, que apresentou uma visão geral sobre a motivação do trabalho e seu objetivo. O Capítulo 2 irá mostrar um conjunto de propostas encontradas na literatura de teste de software que enfocam métodos de seleção de casos de teste de regressão. O Capítulo 3 descreve o método proposto neste trabalho. O Capítulo 4 apresenta a aplicação da técnica realizada em uma organização de testes de software. E por fim, as conclusões e os trabalhos futuros são descritos no Capítulo 5 deste trabalho.

2. Seleção de Testes: Trabalhos relacionados

Este capítulo tem como objetivo principal fornecer o embasamento teórico com relação aos conceitos empregados neste trabalho e o estado da arte em trabalhos relacionados. Serão apresentadas as principais técnicas relacionadas à seleção de casos de teste para testes de regressão, destacando também a abordagem de teste baseado em risco (Risk-Based Testing), um assunto que é alvo de pesquisas por muitos autores.

2.1 Técnicas de Seleção de Testes de Regressão

À medida que um sistema de software evolui em funcionalidades, novas versões precisam ser testadas por meio de testes de regressão. O teste de regressão é um processo caro usado para validar versões novas do software e para detectar se as falhas novas foram adicionadas no código. Sendo assim, reutilizar todos os casos de testes da suíte no teste de regressão, pode ser considerada uma atividade inviável ou extremamente custosa.

Visando aumentar o custo-benefício do teste de regressão, muitos pesquisadores têm se dedicado em definir técnicas de seleção de testes. Rothermel [14], em seu trabalho, considera quatro técnicas que reusam a suíte de testes da versão original do software: retest-all, regression test selection, test suite reduction, e test case prioritization.

2.1.1 Retest All

Reutilização na execução do software modificado de todos os casos de teste previamente desenvolvidos. Casos de testes obsoletos devem ser reformulados ou descartados. Essa é a técnica mais utilizada atualmente. Porém, essa técnica pode ser muito cara, já que executar todas as situações de teste exige muito tempo e grande esforço humano.

2.1.2 Regression Test Selection

O método de seleção de testes de regressão (RTS) também reusa casos de teste, porém, seletivamente, centrando-se sobre subconjuntos de suítes de

teste existentes. Diversos estudos ([23], [24], [25]) têm mostrado que esta técnica pode ter um bom custo-benefício.

Uma técnica de seleção de testes de regressão segura deve garantir que os casos de testes que não foram selecionados não revelem falhas na nova versão. Uma forma de garantir a segurança é executar um grande número de casos de testes, mas isso faz com que a técnica ganhe em segurança e perca em eficiência. Encontrar o equilíbrio entre segurança e eficiência da técnica é um grande desafio.

2.1.3 Test Suíte Reduction

A redução do custo do teste de regressão é dada eliminando casos de teste da suíte de testes de forma permanente. À medida que o software evolui, novos casos de testes são criados para validar novas funcionalidades. Com isso, alguns testes podem se tornar redundantes. A técnica de redução da suíte de testes aumenta a eficiência do conjunto de testes eliminando os testes redundantes. Estudos comprovam ([26], [27]) que esta técnica traz bastante economia no custo de detecção de falhas, porém ela pode igualmente reduzir a eficácia dos testes.

2.1.4 Test Case Priorization

Esta técnica estabelece uma ordenação dos casos de testes de modo que aqueles com prioridade mais elevada, de acordo com algum critério, sejam executados mais cedo no ciclo de teste de regressão do que os casos de testes com baixa prioridade. Por exemplo, os testadores podem querer priorizar os testes utilizando critérios de cobertura de código, tempo de execução, funcionalidades mais utilizadas, probabilidade de detecção de erro, entre outros.

A priorização de caso de teste pode ser usada no teste inicial do software ou durante a fase de teste da regressão. O benefício de usar a priorização durante o teste de regressão é que os resultados da análise precedentes podem ser usados para dar a melhor prioridade às situações de teste futuras [15].

Segundo Rothermel [15], quando o tempo requerido para executar uma suíte de testes por completo é pequeno, a priorização pode não trazer uma boa relação de custo x benefício e a suíte pode ser executada em qualquer ordem. Por outro lado, quando a execução da suíte de testes demanda muito tempo, a priorização é benéfica porque os objetivos do teste podem ser atingidos mais cedo, o que possibilita que o início das atividades de correção também aconteça mais cedo.

Como esta técnica de priorização não rejeita casos de testes, ela pode possuir vantagens sobre as técnicas de Regression Test Selection e Test Suit Redution, reduzindo os riscos de perdas na cobertura devido à remoção dos casos de teste.

Após analisar as quatro técnicas, percebe-se que nos casos onde 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. Esta alternativa será utilizada para o estabelecimento de um método de seleção de casos de teste, escopo deste trabalho.

A proposta de seleção de casos de testes descrita neste trabalho utilizará a abordagem de priorização de testes baseado em riscos (Risk-Based Testing). Esta técnica visa, através da identificação de áreas de maior risco, identificar os casos de testes mais prioritários, de forma a aumentar a eficiência dos testes de regressão diante de restrições de tempo comumente existentes nesta fase [1].

2.2 Testes Baseados em Riscos

De acordo com PMBOK 2004 [16], "risco de projeto é um evento ou condição incerta que, se ocorrer, tem um positivo ou um negativo efeito no objetivo de um projeto. Um risco tem uma causa e, se ocorre, uma consequência no custo, no cronograma ou na qualidade do projeto."

Risco não é um problema, mas pode se tornar se ações preventivas não forem tomadas. Isso pode ser feito determinando ações para melhorar

oportunidades e reduzir ameaças para os objetivos do projeto. Em Engenharia de Software, testar é uma maneira de reduzir o risco. Testar reduz a incerteza e pode encontrar as falhas críticas que podem ser corrigidas [17].

A atividade de execução de testes normalmente é realizada no fim do projeto, quando não se tem muito tempo e recursos disponíveis. Se o sistema for entregue a tempo, muitas funcionalidades poderão não ser testadas. A maioria das organizações escolhe entregar o sistema no tempo certo ao invés de um sistema que funciona perfeitamente [18].

Identificando e priorizando riscos, é possível garantir que o tempo e os recursos limitados serão usados para testar as áreas mais importantes do sistema. A técnica de priorização de testes baseados em riscos é uma prática que decide o que testar usando risco como uma maneira de medir o impacto da não realização dos testes de algumas funcionalidades do sistema.

Essa abordagem de testes baseada em risco (Risk-Based Testing) tem sido alvo de pesquisas por diversos autores ([2], [3], [18], [19], [20], [21]). Estes trabalhos serão descritos a seguir.

2.2.1 Bach

Bach [19] faz uma análise heurística de testes baseados em risco. Esta análise é realizada com dois enfoques: Inside-out e Outside-in.

O Inside-Out, verifica os detalhes de cada situação e identifica os riscos associados a elas. O testador deverá tentar responder, junto ao desenvolvedor, a questão “o que pode dar errado?” em cada funcionalidade. Para cada parte do produto, o testador deve procurar vulnerabilidades, ameaças e vítimas.

No enfoque de Outside-in, uma lista de riscos pré-definidos é consultada e usada para identificar os riscos de cada funcionalidade. Bach usa três tipos de listas: Categorias dos critérios de qualidade, lista de riscos genéricos e catálogo de riscos.

A lista de categorias dos critérios de qualidade foi projetada para verificar alguns requisitos, como usabilidade, desempenho, manutenibilidade, compatibilidade, suportabilidade, entre outros.

A lista de riscos genéricos pode ser aplicada a todos os sistemas. Os principais riscos genéricos levados em consideração são: complexidade, nova funcionalidade, falha recente, modificações, dependências, precisão, requisito estratégico, popular, crítico, etc.

O catálogo de riscos é uma lista de riscos pertencente a um domínio particular. Cada linha da lista é um complemento da frase “Nós poderemos ter problemas com...”.

As três listas podem ser usadas para:

- Decidir que componente ou função deverá ser analisada
- Determinar a escala do interesse
- Recolher informações sobre o que deve ser analisado com mais cuidado
- Determinar a importância de cada risco
- Anotar outros riscos que não estão na lista
- Anotar ocorrências desconhecidas que impactarem a análise de riscos
- Verificar a distribuição de risco

São sugeridas três maneiras de organizar os testes baseados em riscos:

Lista de riscos assistida: Lista de riscos revisada periodicamente.

Matriz de tarefas: Os riscos são ordenados de acordo com sua importância. Cada risco tem uma lista de tarefas para minimizar o impacto do risco.

Matriz de risco do componente: O sistema é quebrado em componentes e uma matriz é montada com informações sobre escala de interesse do risco (baixo, normal e alto) e uma lista de riscos heurísticos

(usabilidade, complexidade, desempenho, nova função...). Durante o teste, os componentes são testados com foco nos riscos informados na matriz.

2.2.2 Amland

O processo de priorização proposto por Amland [2] tem como ponto principal a avaliação de riscos, utilizando elementos de custo da falha sob o ponto de vista do vendedor e do cliente e a probabilidade da falha ocorrer.

A seleção dos casos de teste é realizada através do ranking gerado com o cálculo da exposição de risco (Risk Exposure) de cada função $RE(f)$ usando a seguinte fórmula:

$$Re(f) = P(f) * \frac{C(c) + C(v)}{2}$$

Onde:

$RE(f)$ = Fator de exposição de risco de uma função

$P(f)$ = Probabilidade de ocorrência de falha em uma função

$C(c)$ = Custo decorrente da falha na visão do cliente

$C(v)$ = Custo decorrente da falha na visão do vendedor

Amland [2] realizou um estudo de caso com uma aplicação bancária depois que percebeu que ele não tinha recursos suficientes para testar todas as possibilidades. Os resultados mostraram que este método consumiu menor esforço do que a estimativa original baseada na abordagem tradicional de testes.

2.2.3 Chen

Chen [3] foi quem introduziu a metodologia de Testes Baseados em Risco em Testes de Regressão.

A abordagem de Chen [3] é focada em casos de testes que testam áreas arriscadas. A fórmula do fator de exposição de risco $RE(f)$ proposta por Amland [2] é utilizada por Chen [3] apenas com uma mudança no que diz ao aspecto de probabilidade $P(f)$. O elemento probabilidade do modelo original é alterado

para probabilidade da severidade, que combina a severidade do erro encontrado com a probabilidade do teste falhar.

2.2.4 Schaefer

Shaefer [21] foca a priorização dos testes avaliando as partes mais importantes e as mais críticas do produto. As partes do sistema são julgadas mais importantes ao utilizar fatores como custo decorrente da falha, áreas de grande visibilidade (grande número de usuários afetados se houver falhas), e áreas que tenham suas funções utilizadas de forma intensiva. As partes mais críticas do sistema são determinadas utilizando aspectos como complexidade, áreas que sofrem mudanças constantes, novas tecnologias, soluções e métodos, número de pessoas envolvidas, pressão de tempo em cima dos desenvolvedores e fatores locais.

Pesos são atribuídos para cada fator relevante, em cada parte do sistema. Valores altos significam que aquelas áreas são as mais importantes ou as mais críticas. Esses valores são multiplicados pelos seus pesos e adicionados. O valor mais elevado se refere à parte do sistema mais arriscada e esta precisará ser priorizada. Um exemplo do cálculo do risco adaptado de Shaefer [21] é mostrado na tabela.

Áreas	Crítica	Visibilidade	Complexa	Modificações frequentes	RISCO
Pesos	3	10	3	3	
Ordem de registro	2	4	5	1	828
Faturamento	4	5	4	2	1116
Odem estatística	2	1	3	3	288
Gestão de relatórios	2	1	2	4	288
Desempenho de ordem registro	5	4	1	1	330
Desempenho das estatísticas	1	1	1	1	78
Desempenho do faturamento	4	1	1	1	132

Tabela 1 – Exemplo do cálculo do risco, adaptado de Shaefer [21]

2.2.5 Gerrard

Uma abordagem de testes baseada em riscos para sistemas de negócios on-line é proposta por Gerrard [18], porém este método pode ser aplicado também para outros tipos de sistemas. O processo consiste de cinco estágios: identificação do risco, análise do risco, resposta do risco, escopo do teste e definição do processo de teste.

Identificação do risco: Um inventário de falhas é preparado. Eles são organizados em listas de verificação, similares às usadas por Bach [19], como descrito na seção 2.2.1. Quando as falhas são identificadas, é possível separar falhas complexas e descartar duplicações.

Análise de risco: Neste estágio, uma reunião entre os representantes do negócio, do desenvolvimento, do suporte e dos testadores é realizada com a intenção de considerar os riscos e avaliar a probabilidade e a consequência das falhas. A exposição do risco é calculada.

Resposta do risco: Se o risco é testável, ele pode ser transformado em um teste cujo objetivo utiliza a descrição do risco. Uma escala de 1 a 5 relata a confiança do testador na eficiência do teste, na garantia que o risco pode ser solucionado através do teste. Um valor de um indica que o testador não tem nenhuma confiança. Cinco é dado se o testador tem alta "confiança de que o teste irá encontrar falhas e fornecer provas de que o risco foi abordado" [18]. O produto da exposição do risco pela eficácia do teste irá dar o número de prioridade do teste. Um exemplo desta pontuação é dado na tabela abaixo.

ID	Risco	Probabilidade	Consequência	Exposição do Risco	Eficiência do Teste	Prioridade do Teste
Plataforma do Cliente						
04	HTML inválido ou incorreto.	4	4	16	5	80
05	O HTML funciona em um navegador, mas não funciona em outros.	4	4	16	4	64
Funcionalidade dos componentes						
13	Os erros de digitação nas páginas da Web podem irritar usuários e prejudicar nossa credibilidade.	5	4	20	5	100
Integração						
19	Os links para outras páginas que não funcionam tornam o site inutilizado.	4	4	16	5	80

Tabela 2 – Exemplo de pontuação da análise de risco x eficiência do teste, adaptada de Gerrard [18]

Escopo do teste: Neste momento é realizada uma atividade de revisão, com a participação de todas as partes interessadas e equipe de funcionários com conhecimento técnico. A reunião discute e concorda com o total de testes definidos para o escopo. A tabela gerada pelas fases anteriores e o custo estimado para executar o teste são usados para definir o escopo do teste.

Definição do processo de teste: Com o escopo de teste definido, é possível cuidar de dependências, realizar estimativa de tempo para cada fase do teste e definir cada estágio do processo de teste. O passo-a-passo do processo deve ser documentado.

2.2.6 Besson

Besson [8] descreve como realizar o teste baseado em risco priorizando as funções mais importantes do sistema. O método começa com a identificação das funções vitais que podem ter falhas de severidade elevada. A severidade é definida medindo o impacto negativo que uma falha pode causar ao negócio: alta, média ou baixa. Se o usuário desperdiça tempo e dinheiro por causa de uma falha no sistema, a severidade do erro é considerada alta.

As possíveis maneiras de avaliar a severidade são entrevistar o usuário final do sistema ou usar estatísticas do registro das versões anteriores do sistema. É importante também observar as funções usadas com mais frequência, já que estas têm maiores risco de falhar. Os casos de testes são

projetados e classificados em ordem de esforço, com o caso de teste com menor esforço primeiro. Os testes são executados na ordem definida.

2.3 Considerações Finais

Teste de regressão é o processo de validar o software modificado para fornecer a confiança de que as partes mudadas do software se comportem como pretendidas e que as partes inalteradas do software não tenham sido afetadas pela modificação [22]. No entanto, testar completamente todas as possibilidades nem sempre é possível, já que esta tarefa exige um grande esforço e normalmente o cronograma é apertado.

Este capítulo apresentou algumas técnicas que têm sido aplicadas para seleção de testes de regressão. Também foram apresentadas diversas abordagens relativas aos Testes Baseados em Risco (Risk-Based Testing).

3. Método para Seleção de Casos de Teste

O método descrito a seguir, proposto neste trabalho, tem como objetivo estabelecer critérios para a priorização de testes de regressão, viabilizando uma cobertura mais efetiva e minimizando o esforço nos ciclos de regressão. Além disso, serão apresentados alguns critérios auxiliares, que são utilizados somente após a lista de prioridades estar definida e que ajudam tanto na seleção da quantidade de casos de teste que farão parte do escopo do ciclo de regressão, como também na reordenação da lista de prioridades.

Na seção 3.1, serão apresentadas as premissas adotadas no estabelecimento do método. A seção 3.2 descreve como serão feitas as pontuações dos critérios. Em seguida, a seção 3.3 irá mostrar os critérios de priorização com alguns exemplos. A seção 3.4 descreve os critérios auxiliares. O método proposto é finalmente apresentado na seção 3.5. E por fim, as considerações finais são descritas na seção 3.6.

3.1 Premissas Adotadas

Viana [1] adota em seu trabalho algumas premissas, com o objetivo de estabelecer um cenário para a viabilidade e aplicabilidade do método de seleção de testes de regressão para automação. O método proposto neste trabalho fará uso de algumas dessas premissas:

Acervo existente: Como um dos objetivos deste trabalho é selecionar casos de teste para reduzir o escopo do ciclo de testes, então deve existir um acervo previamente criado de casos de teste.

Documentação: Para viabilizar o processo de análise e priorização, é importante que os casos de teste estejam organizados em suítes e catalogados de forma centralizada.

Julgamento de Especialistas: São os testadores que irão julgar os testes de acordo com os critérios de priorização, portanto estes devem ter conhecimento aprofundado nos testes e conhecimento do sistema, pois é sua experiência que contribuirá com as informações relevantes para a priorização.

Histórico: Devem existir informações históricas sobre tempo de execução dos testes, quantidade de falhas encontradas e severidade destas falhas.

3.2 Pontuação dos critérios

Schaefer [21] afirma em seu trabalho que o método geral para calcular a priorização dos testes das áreas do sistema é atribuindo pesos. Esta atribuição pode ser realizada de maneiras muito elaboradas, porém isto tomará muito tempo. A escolha de três pesos é a opção ideal.

Viana [1] sugere em seu trabalho que estes três pesos sejam os valores 1, 3 e 9, sendo 1 para o fator menos importante, 3 para o fator com influência normal e 9 para o fator com influência muito forte.

Neste trabalho, da mesma forma que Viana [1], serão utilizados os valores iguais a 1, 3 e 9.

3.3 Critérios de Priorização

Da mesma maneira proposta por Viana [1], o método descrito neste trabalho é baseado em critérios de priorização. Os casos de teste serão pontuados de acordo com estes critérios e os que obtiverem maiores pontuações, serão os primeiros a serem testados. Os critérios de priorização que serão levados em consideração estão listados na tabela abaixo:

Critérios	
Sigla	Nome
FER	Fator de Exposição de Risco
TAU	Testes Automatizados
TPN	Tipo de Testes (Testes Positivos ou Negativos)

Tabela 3 – Critérios de Priorização de Casos de Teste

Os detalhes de cada critério serão descritos a seguir, juntamente com alguns atributos que são usados como base para o cálculo do valor de cada critério.

3.3.1 Fator de Exposição de Risco

Amland [2] apresenta uma técnica simples de seleção de casos de testes baseada em riscos com apenas dois elementos: Probabilidade e Custo (Impacto).

Como foi apresentado no capítulo anterior, com esses elementos, Amland [2] criou uma maneira de calcular o fator de exposição de risco (Risk Exposure) que classifica os casos de teste através da fórmula:

$$RE(f) = P(f) \times C(f)$$

Onde:

$RE(f)$ = Fator de exposição de risco de uma função

$P(f)$ = Probabilidade de ocorrência de falha em uma função

$C(f)$ = Média entre o impacto decorrente da falha de uma função na visão do cliente e na visão do vendedor.

Estes mesmos elementos também foram abordados por Chen [3] e Gerrard [18] e também são utilizados no método de priorização proposto neste trabalho.

A seguir, cada um destes dois elementos será detalhado.

- Impacto

A consequência que uma falha pode trazer para o sistema também deve ser observada. O impacto decorrente de uma falha em uma funcionalidade crítica de um sistema será muito maior do que em uma área que não afeta diretamente o funcionamento básico da aplicação. Este impacto deve ser avaliado do ponto de vista do cliente (por exemplo, perda de espaço no mercado, não cumprimento de regulamentos governamentais) e do ponto de vista do vendedor do serviço (por exemplo, custo de manutenção elevado do software por causa de uma função com falhas). Amland [2] admite que o

impacto ao cliente é igualmente importante na análise de risco ao impacto ao vendedor. Sendo assim, o valor de $C(f)$ é calculado a partir de uma média aritmética entre os valores da consequência de uma falha avaliada pela visão do cliente e pela visão do vendedor.

No método proposto, o impacto será pontuado pelos valores 1, 3 e 9, onde 1 representa o impacto mínimo em caso de uma falha em uma determinada função. Da mesma maneira, o valor 9 representa o de maior impacto. Esta definição é mostrada na Tabela 4:

Impacto decorrente da falha de uma função	$C(f)$
Baixo	1
Médio	3
Alto	9

Tabela 4 – Impacto decorrente da falha de uma função

- Probabilidade

Quanto maior o número de erros no sistema, maior a probabilidade de que mais erros não detectados existam. Se um componente tem defeitos em um ciclo de teste, é muito provável que mais defeitos sejam encontrados neste componente em um teste de regressão, já que a correção destes defeitos causa alterações no código, podendo assim inserir novos defeitos. Sendo assim, componentes com falhas devem ser tratados com atenção em um ciclo de regressão.

A partir da análise do histórico de resultados do teste, o percentual de falhas identificadas pelo teste em relação ao seu total de execuções é calculado. Um teste que identificou falha na maioria das vezes em que foi executado é melhor candidato para priorização que um teste que quase não falhou em suas execuções. Viana [1] utilizou este critério para selecionar casos de teste para automação e a escala percentual proposta por ela é mostrada na Tabela 4:

Percentual de falhas identificadas	F
Menor que 25%	0
Maior ou igual a 25% e menor que 50%	1
Maior ou igual a 50% e menor que 75%	3
Maior ou igual a 75%	9

Tabela 5 – Percentual de falhas identificadas

Após executar toda a suíte de teste, podemos obter informações sobre a severidade dos testes falharam. Ter a severidade destes defeitos (quão sério esse defeito é) é de grande importância para a qualidade do software. Chen [3] propõe uma mudança no modelo proposto por Amland [2] no que diz respeito ao aspecto de probabilidade $P(f)$. O elemento probabilidade do modelo original é alterado para probabilidade da severidade, que combina a severidade do erro encontrado com a probabilidade do teste falhar.

Com relação à severidade das falhas encontradas, podemos defini-las da seguinte maneira [7]:

Alta: Erros que bloqueiam o trabalho de desenvolvedores ou de testes ou que provocam panes, perdas de dados, problemas severos de memória.

Média: Muita(s) perda(s) de funcionalidade(s).

Baixa: Menor perda de funcionalidade, algum problema ou alteração que necessita de pouco trabalho, problema ou mudança cosmética, como palavra grafada incorretamente ou texto desalinhado.

Podemos concluir que um erro classificado com severidade “Alta” tem maior prioridade que um erro com severidade “Baixa”. Sendo assim, o peso atribuído a uma falha do tipo “Alta” é bem maior que os pesos atribuídos às demais classificações. A tabela abaixo mostra esta distribuição:

Severidade da falha encontrada	S
Baixa	1
Média	3
Alta	9

Tabela 6 – Severidade da falha encontrada

Com essa avaliação, podemos estimar a probabilidade da severidade para cada caso de teste, fazendo uma relação de valores a partir dos resultados da multiplicação entre os critérios de Percentual de falhas identificadas (F) e Severidade da falha encontrada (S). Esta relação que classifica a probabilidade da severidade é dada da seguinte forma: $P(f)$ será igual a 9 quando $F \times S$ for igual a 81 ou 27, $P(f)$ será igual a 3 quando $F \times S$ for igual a 9 ou 3 e $P(f)$ será igual a 1 quando $F \times S$ for igual a 1 ou 0. A tabela abaixo mostra essa regra:

$F \times S$	$P(f)$
0	1
1	1
3	3
9	3
27	9
81	9

Tabela 7 – Classificação de $P(f)$ a partir de $F \times S$

A Tabela 8 mostra $P(f)$ para alguns casos de teste como exemplo:

Caso de Teste	F	S	F x S	P(f)
CT01	0	3	0	1
CT02	1	1	1	1
CT03	1	9	9	3
CT04	3	1	3	3
CT05	3	9	27	9
CT06	9	9	81	9

Tabela 8 – Exemplo de Probabilidade de Severidade por Caso de Teste

Combinando os resultados das Tabelas 4 e 8, podemos finalmente calcular o Fator de Exposição de Risco ($RE(f)$) para cada caso de teste, como mostrado na tabela a seguir:

Caso de Teste	C(f)	P(f)	$RE(f) = C(f) \times P(f)$
CT01	1	1	1
CT02	3	1	3
CT03	3	3	9
CT04	9	3	27
CT05	1	9	9
CT06	9	9	81

Tabela 9 – Exemplo do cálculo de Fator de Exposição de Risco

Tendo as informações atualizadas referentes à Impacto e Probabilidade de cada caso de teste, é possível fazer o cálculo do Fator de Exposição de Risco de forma automática.

3.3.2 Testes Automatizados

Os testes podem ser classificados de três maneiras quanto a sua forma de testar: Manuais, Semi-automáticos e Automáticos.

Os testes manuais são testes executados manualmente pelos testadores, seguindo os passos do Projeto de Teste. Projeto de Teste é um conjunto predefinido de passos e resultados esperados, baseados nas especificações do projeto. Essa forma de testar é a mais utilizada pelas empresas. Os testes manuais podem ser classificados como "Testes Exploratórios", que é a criação e a execução do teste ao mesmo tempo.

Os testes automáticos são aqueles que fazem uma geração automática da massa de dados de testes, bem como a execução dos mesmos. Eles se baseiam em ferramentas computacionais para avaliar o comportamento da aplicação, fornecendo entradas e analisando as saídas de forma a identificar a presença de defeitos no software. Sua utilização é fundamental para acelerar o ciclo de desenvolvimento e permitir a adoção de abordagens incrementais na produção de software [1].

Os testes semi-automáticos podem ser considerados como uma mistura dos outros dois tipos. São testes automáticos que precisam de intervenção humana durante a execução dos mesmos.

Fazendo uma comparação entre as formas de testar, é fácil perceber que os testes automáticos podem ser executados de forma paralela ou em períodos noturnos, reduzindo o tempo de realização dos testes e liberando os testadores para atividades onde a intervenção humana seja realmente necessária e útil [1]. Por isso, os testes automáticos devem ser priorizados.

Sendo assim, podemos classificar as formas de testar de acordo com a tabela abaixo:

Testes Automatizados	Peso (A)
Manuais	1
Semi-automáticos	3
Automáticos	9

Tabela 10 – Pesos para classificação dos testes

3.3.3 Tipo de Testes (Testes Positivos ou Negativos)

Teste de software pode ser definido como um tratamento de uma entrada, uma ação, e uma saída. A ação atua em cima da entrada para derivar alguma saída. Os testes podem ser classificados como Positivos ou Negativos. Nyman [4] conceitua esses testes da seguinte maneira:

Testes Positivos: Teste que tenta demonstrar que um determinado módulo dado de uma aplicação faz aquilo que ele deve fazer; verifica se o aplicativo funciona como ele realmente deve funcionar. Como exemplo, podemos citar uma aplicação que exibe uma mensagem de erro quando realmente deva ser exibida.

Testes Negativos: Teste que tenta demonstrar que o módulo não faz nada que ele não deva fazer, são testes que são introduzidos com a intenção de “quebrar” o sistema. O comportamento inesperado conseguido a partir de um teste negativo é quando o sistema não exibe uma mensagem de erro quando deve ou exibe uma mensagem de erro quando não deve.

Segundo Desikan [5], ao selecionar casos de testes para o ciclo de regressão, não se deve enfatizar os testes que tenham nenhuma (ou pouca) relevância no reparo dos erros. Dasikan [5] afirma que se deve dar preferência aos testes positivos, já que uma falha neste tipo de teste pode criar alguma confusão. É recomendável, porém, que os ciclos regulares de teste, antes do teste de regressão, devam ter uma mistura de casos de teste positivos e negativos.

Considerando estes conceitos, é aconselhável utilizar esta classificação como critério de priorização de casos de teste. Serão atribuídos pesos aos tipos

de teste de forma que os testes positivos obtenham maior pontuação que os testes negativos. A tabela abaixo apresenta esses pesos:

Tipo de Teste	Peso (T)
Positivo	3
Negativo	1

Tabela 11 – Pesos para Tipos de Teste

3.4 Critérios Auxiliares

Além dos critérios de priorização, definidos na seção anterior, outros três critérios são usados na nossa proposta: Tempo de Execução, Recursos Disponíveis e Dependência dos Casos de Teste.

Diferentemente dos critérios citados anteriormente, os critérios auxiliares não interferem no cálculo de priorização de casos de teste. Eles serão utilizados somente após a lista de prioridades estar definida, ajudando tanto na seleção da quantidade de casos de teste que serão escopo no ciclo de regressão, como também na reordenação da lista de prioridades.

A seguir, cada um desses critérios auxiliares serão descritos.

3.4.1 Tempo de Execução e Avaliação do Teste

O tempo para execução de um teste engloba o tempo gasto na preparação dos dados necessários para a sua realização, normalmente chamado tempo de setup, mais o tempo da execução do teste propriamente dito, bem como o tempo gasto em alguma atividade realizada após a execução do teste para retornar o software testado à sua condição inicial [1]. O tempo para avaliação de um teste é o tempo reservado para analisar o erro e reportá-lo à equipe de desenvolvimento.

A partir de dados históricos de ciclos anteriores é possível obter uma média do tempo gasto para a execução e avaliação de um caso de teste. Tendo esse valor, podemos, de acordo com o tempo total disponível para execução

dos testes, selecionar os N primeiros casos de teste da lista de prioridades resultante do cálculo feito com base nos critérios de priorização. Se uma build de um sistema é liberada para ser testada em 2 dias, porém existe uma suíte de testes que para ser testada por completo levaria 5 dias, então serão selecionados os casos de teste com maiores pontuações, onde somados os seus tempos de execução e avaliação, não ultrapassem os 2 dias que se tem disponível.

Dessa forma, a estimativa da quantidade de casos de teste a serem testados será precisa e serão priorizados os casos de teste de maiores riscos.

3.4.2 Recursos Disponíveis

O tempo total gasto em um ciclo de testes está intimamente ligado aos recursos disponíveis, tanto os recursos humanos como também recursos materiais. A duração da maioria das atividades será significativamente influenciada pelos recursos a elas designadas. Por exemplo, duas pessoas trabalhando juntas podem ser capazes de completar uma atividade na metade do tempo que levariam para fazê-lo individualmente, enquanto uma pessoa trabalhando meio expediente em uma atividade levará geralmente, no mínimo, duas vezes o tempo que a mesma pessoa levaria trabalhando o expediente completo [6].

Desse modo, o tempo destinado aos trabalhos de execução e avaliação de um ciclo de teste será calculado de acordo com a quantidade de pessoas disponíveis bem como a quantidade de computadores ou de sistemas embarcados (nos casos de testes para celulares, impressoras, etc.) disponíveis e necessários para a realização dos testes.

3.4.3 Dependência dos Casos de Teste

Em alguns casos, em uma suíte de testes, existem casos de teste que são dependentes de outros. Um exemplo desta situação é um fluxo de "alteração". Para alterar um elemento, este deve ter sido incluído anteriormente. Portanto testes referentes à alteração devem vir depois dos testes referentes à inclusão.

Se na lista de prioridades resultante do método proposto tiver um caso de teste (CT04 no exemplo abaixo) onde sua execução depende da saída de outro caso de teste (CT02 no exemplo) que está em uma posição mais abaixo dele, este segundo deverá subir até ocupar uma posição na frente daquele que tem a dependência.

Para ilustrar a explicação, seguem as tabelas abaixo:

Lista de prioridades
CT06
CT04
CT05
CT03
CT02
CT01

Tabela 12 – Lista de prioridades antes

Lista de prioridades
CT06
CT02
CT04
CT05
CT03
CT01

Tabela 13 – Lista de prioridades depois

3.5 Método Proposto

O método de priorização de testes de regressão proposto neste trabalho utiliza todos os critérios que foram estabelecidos nas seções anteriores. Os critérios de priorização montam uma lista de prioridades sendo os de maiores pontuações os primeiros a serem testados. Já os critérios auxiliares servem para reajustar a lista de prioridades (se necessário) e ajudar na seleção da quantidade de casos de teste que serão escopo no ciclo de regressão.

Os critérios de priorização não têm a mesma relevância no processo de seleção. Sendo assim, alguns pesos serão atribuídos aos critérios de acordo com sua importância, como mostra a Tabela 14:

Critérios		
Sigla	Nome	Peso
FER	Fator de Exposição de Risco	3
TAU	Testes Automatizados	2
TPN	Tipo de Testes (Testes Positivos ou Negativos)	1

Tabela 14 – Critérios de Priorização e Pesos

A partir desses pesos, podemos montar uma equação para calcular a pontuação de cada caso de teste:

$$PCT = FER \times P_{FER} + TAU \times P_{TAU} + TPN \times P_{TPN}$$

Onde:

PCT = Prioridade do Caso de Teste t

FER = Fator de Exposição de Risco

PFER = Peso de FER

TAU = Testes Automatizados

PTAU = Peso de TAU

TPN = Tipo de Testes (Testes Positivos ou Negativos)

PTPN = Peso de FAT

Com os resultados do PCT de cada caso de teste, uma lista será montada, sendo o primeiro caso de teste da lista o candidato cujo teste tem maior importância, e o último da lista aquele cujo teste pode esperar um próximo ciclo.

O exemplo abaixo demonstra a aplicação do método proposto, considerando uma suíte de testes com 15 casos de teste. A primeira tabela apresenta o cálculo do Fator de Exposição de Risco de cada função (RE(f)). A segunda, mostra o cálculo da Prioridade do Caso de Teste (PCT), que representa a pontuação final de cada caso de teste. E por fim, a lista de prioridades resultante, ordenada da pontuação maior para a menor.

Caso de Teste	Custo	Probabilidade				FER
		$C(f)$	F	S	F x S	$P(f)$
CT01	1	0	9	0	1	1
CT02	3	1	1	1	1	3
CT03	9	3	1	3	3	27
CT04	1	1	3	3	3	3
CT05	3	3	3	9	3	9
CT06	9	9	9	81	9	81
CT07	9	1	3	3	3	27
CT08	3	9	3	27	9	27
CT09	9	0	3	0	1	9
CT10	1	9	9	81	9	9
CT11	3	0	3	0	1	3
CT12	9	1	9	9	3	27
CT13	1	0	1	0	1	1
CT14	9	9	1	9	3	27
CT15	9	3	9	27	9	81

Tabela 15 – Cálculo do Fator de Exposição de Risco

Caso de Teste	FER	TAU	TPN	PCT
	Peso 3	Peso 2	Peso 1	
CT01	1	1	3	8
CT02	3	1	3	14
CT03	27	1	1	84
CT04	3	3	1	16
CT05	9	1	1	30
CT06	81	1	3	248
CT07	27	1	3	86
CT08	27	3	1	88
CT09	9	1	3	32
CT10	9	9	1	46
CT11	3	1	1	12
CT12	27	9	1	100
CT13	1	3	1	10
CT14	27	9	3	102
CT15	81	9	1	262

Tabela 16 – Cálculo da Prioridade dos Casos de Teste

Caso de Teste	PCT
CT15	262
CT06	248
CT14	102
CT12	100
CT08	88
CT07	86
CT03	84
CT10	46
CT09	32
CT05	30
CT04	16
CT02	14
CT11	12
CT13	10
CT01	8

Tabela 17 – Lista de Casos de Teste ordenados por Prioridade

Tendo informações históricas do tempo médio gasto para executar e avaliar cada caso de teste, podemos limitar o tamanho da suíte de testes do próximo ciclo. No nosso exemplo, o tempo (em minutos) de cada caso de teste é mostrado na tabela abaixo:

Caso de Teste	Tempo (min)
CT15	8
CT06	23
CT14	6
CT12	7
CT08	11
CT07	21
CT03	12
CT10	5
CT09	21
CT05	19
CT04	10
CT02	19
CT11	26
CT13	6
CT01	25

Tabela 18 – Tempo médio por CT medido em minutos

Supondo que o ciclo de regressão deve ser realizado em 1 hora (60 minutos), deveremos selecionar apenas os primeiros cinco casos de teste, já que a soma de seus tempos resulta em 55 minutos. Os 5 minutos que sobram, podem ser deixados como contingência no caso de algum caso de teste extrapolar o tempo previsto. Ou então, a equipe pode incluir na suíte um próximo caso de teste da lista que tenha um tempo médio de execução e avaliação menor ou igual a 5 minutos. No exemplo, podemos incluir na suíte o CT10.

Supondo que a equipe que realizará os testes é composta por dois testadores e são disponibilizados recursos suficientes para que os dois possam testar independente um do outro, supondo ainda que a produtividade dos dois é a mesma, podemos então considerar que cada um terá 1 hora para realizar os testes em paralelo, que somando equivale a 2 horas. Sendo assim, podemos considerar 120 minutos para o ciclo de regressão. Neste caso, serão selecionados os nove primeiros casos de teste, já que a soma de seus tempos resulta em 114 minutos. Com os 6 minutos que sobram, podemos incluir o CT13 na suíte de testes ou deixá-los como tempo de sobra para plano de contingência.

Sendo a execução do CT06 dependente do resultado proveniente do CT01, é necessário fazer um ajuste na lista de prioridades, trazendo o CT01 para frente do CT06:

Caso de Teste	Tempo (min)
CT15	8
CT01	25
CT06	23
CT14	6
CT12	7
CT08	11
CT07	21
CT03	12
CT10	5
CT09	21
CT05	19
CT04	10
CT02	19
CT11	26
CT13	6

Tabela 19 – Lista de prioridades após tratamento de dependência

3.6 Considerações Finais

Este capítulo propôs e detalhou cada fase do método de seleção de casos de teste sugerida, considerando aspectos relevantes para esta seleção.

Baseando-se em referências técnicas de Testes de Regressão bem como em Testes Baseados em Risco, critérios foram definidos e a eles foram atribuídos pesos. A partir disso, cada caso de teste recebeu uma pontuação e uma lista de prioridades foi montada, para que depois, baseando-se em critérios auxiliares como tempo, recurso e dependência, a seleção fosse feita da maneira mais eficiente possível.

4. Experimento

O objetivo deste capítulo é apresentar o experimento realizado com a finalidade de avaliar a aplicação da abordagem proposta neste trabalho. O experimento foi realizado com a cooperação de uma equipe que faz parte de uma organização de testes de software.

Este capítulo inicia com uma descrição da organização bem como o processo atual utilizado para priorização dos casos de testes em ciclos de regressão. Em seguida, será apresentado o relato do experimento e logo após, uma avaliação dos resultados obtidos.

4.1 Contexto e Processo Atual

A avaliação da efetividade do método foi realizada através de um experimento em uma organização de testes de software. Esta organização é formada por uma equipe de nove pessoas, sendo um coordenador, um gerente, três engenheiros de teste, três estagiários de teste e um técnico administrativo.

A aplicação utilizada no experimento possui cerca de 520 casos de testes, distribuídos em 20 features. Os ciclos de teste de regressão têm uma média de 180 casos de testes para serem executados normalmente em cerca de sete dias.

Ao longo do tempo, a equipe de testes da organização em questão estava trabalhando com testes de regressão total. À medida que o projeto foi crescendo, novos casos de testes foram criados, porém, os recursos humanos e o tempo disponibilizado para execução do ciclo de teste permaneceram os mesmos. Nos últimos ciclos, a organização realizou uma seleção de casos de testes baseada em alterações conhecidas no software e pela identificação dos casos de testes falharam em ciclos anteriores. A partir disso, foram selecionados os casos de teste não executados anteriormente.

Cientes de que uma seleção mais efetiva pode trazer um retorno maior e economia de esforço para a organização, os líderes da equipe perceberam a necessidade de se utilizar um critério de seleção mais robusto. A seleção

realizada pela organização não contribuíram significativamente na redução do esforço da equipe além de não ter trazido benefícios relevantes, já que alguns casos de testes de alto impacto não eram selecionados por não terem falhado anteriormente e outros não eram escolhidos por não fazerem parte diretamente dos módulos que sofreram alterações no código. Não foram considerados fatores importantes para priorização tais como o impacto que um erro pode trazer para o sistema, severidade da falha encontrada, testes automáticos, testes positivos e outros critérios estabelecidos na proposta deste trabalho.

4.2 Relato do Experimento

Para realizar o experimento, foram selecionadas três features, denominadas neste trabalho por A, B e C. A feature A tem um total de 69 casos de testes, a B tem 144 e a C tem 59, resultando em um total de 272 casos de testes. Estas features foram escolhidas para a realização do experimento por serem as que possuem o maior número de casos de testes e por testarem as partes mais importantes do sistema.

A organização tem um processo de testes bem definido. Um caso de teste, ao ser elaborado, é classificado de acordo com algumas características que foram úteis para a realização do experimento. Informações como tipos de testes (positivos e negativos), se são ou não automatizados e o impacto decorrente de um erro no sistema já existiam e foram utilizadas. Dados históricos sobre os ciclos de testes também são guardados, como o tempo de execução de cada caso de teste, a quantidade de vezes que ele falhou e a severidade do erro.

Para a realização do experimento, foram utilizados os dados referentes ao último ciclo de testes realizado pela organização. Após ter coletado todos os dados necessários para aplicação do método, uma planilha eletrônica foi criada de acordo com o modelo apresentado abaixo:

Caso de Teste	C (f)	Percentual de Falhas	F	Severidade	S	F x S	P (f)	FER	Testes Automatizados	TAU	Tipo de Teste	TPN	PCT	Tempo de Execução

Tabela 20 – Planilha Eletrônica utilizada em experimento

A pontuação que indica a prioridade de cada caso de teste foi calculada. Quanto maior o valor de PCT, maior a prioridade. A lista de casos de testes foi organizada pelo valor resultante do cálculo do PCT, em ordem decrescente de prioridade. Um exemplo de uma planilha preenchida com os dados detalhados do experimento pode ser encontrado no Anexo I.

4.3 Análise de Resultados

No ciclo de teste analisado, foi selecionado na feature A um total de 23 casos de testes. Fazendo uma comparação entre a lista de prioridades resultante da aplicação do método e os casos de teste selecionados pela organização, apenas 10 desses casos de teste estão entre os 23 primeiros da lista, ou seja, 56% dos casos de teste selecionados não são prioritários. A tabela abaixo mostra esta comparação, destacando os 10 casos de teste:

Feature A	
Lista de Prioridades	CTs Seleccionados
A_CT39	A_CT39
A_CT40	A_CT40
A_CT41	A_CT41
A_CT50	A_CT50
A_CT27	A_CT27
A_CT38	A_CT22
A_CT44	A_CT42
A_CT48	A_CT43
A_CT52	A_CT44
A_CT54	A_CT45
A_CT01	A_CT47
A_CT10	A_CT48
A_CT13	A_CT59
A_CT22	A_CT25
A_CT30	A_CT26
A_CT33	A_CT32
A_CT34	A_CT46
A_CT35	A_CT64
A_CT36	A_CT65
A_CT37	A_CT66
A_CT42	A_CT67
A_CT43	A_CT68
A_CT58	A_CT69

Tabela 21 – Comparação entre a lista de prioridades resultante com os CTs seleccionados da feature A no último ciclo de testes

Na comparação realizada com a feature B, dos casos de testes seleccionados, 36% não eram prioritários de acordo com a lista resultante do método proposto neste trabalho. Na feature C a diferença foi maior, 71% dos casos de teste não eram prioritários.

Com relação aos critérios auxiliares, é possível fazer as seguintes considerações:

- O tempo disponibilizado para a realização dos testes da feature A foi suficiente para testar 23 casos de teste. Analisando a lista de prioridades resultante da aplicação do método proposto neste trabalho, coincidentemente os 23 primeiros casos de testes deveriam ser seleccionados, de acordo com o tempo disponível para testar esta feature.

- Com relação ao tempo disponibilizado para os testes da feature B, só seria possível selecionar 34 casos de teste ao invés de 74, como aconteceu no ciclo de testes analisado no experimento. O mesmo acontece com a feature C, onde a quantidade de casos de teste selecionados para o teste no ciclo é maior que a quantidade sugerida pelo método, ao considerar o tempo médio de cada caso de teste.

- A organização onde o experimento foi realizado não tem nenhum caso de teste com dependência. Portanto, uma reorganização na lista de prioridades não foi necessária.

4.4 Considerações Finais

Este capítulo apresentou o experimento realizado em uma organização de testes de software com o objetivo de avaliar a eficiência do método proposto para seleção e priorização de casos de teste. Este experimento foi realizado com a colaboração de um time de teste, o que permitiu a avaliação da abordagem proposta na prática.

Com base na observação deste experimento, podemos perceber a importância da utilização de um método para priorização de casos de teste, de forma a reduzir o escopo dos testes sem comprometer os resultados.

5. Conclusão

Com a massiva utilização do software, nas mais diversas atividades, este começou a ter, uma cobrança e exigência cada vez maior por parte dos usuários, principalmente em relação a garantias da sua qualidade [28]. Para continuarem competitivas e manterem sua fatia de mercado duramente conquistada, as empresas desenvolvedoras de software devem estar antenadas com as constantes novidades do mundo informatizado, visando sempre beneficiar o cliente para que ele se torne assíduo e satisfeito.

Em face da demanda por qualidade de software, surgiram e surgem a cada dia, alternativas que englobam a melhoria do processo. A atividade de testes é uma das mais importantes no que diz respeito à garantia de qualidade.

Com esse contexto, o trabalho apresentado neste documento propõe um método de seleção de casos de teste para testes de regressão. Com o intuito de avaliar o funcionamento da estratégia proposta, um experimento foi realizado em uma organização de testes de software.

Este capítulo descreve as principais contribuições deste trabalho, bem como propostas de evoluções futuras para sua consolidação.

5.1 Principais contribuições

A principal motivação para realização deste trabalho é aumentar a efetividade no processo de seleção de casos de testes para ciclo de regressão. O método apresentado permite selecionar e priorizar para execução os casos de teste com melhor relação custo-benefício, levando-se em consideração critérios como o tempo e recursos disponíveis para execução e as dependências dos casos de teste.

O método sugerido pode ser adaptado e empregado em qualquer organização. Além disso, o procedimento apresentado neste trabalho não adiciona custos extras ao processo de teste, visto que, são utilizados registros de dados históricos e o julgamento dos especialistas, que uma vez realizado, são necessárias apenas atualizações periódicas.

5.2 Trabalhos Futuros

Com a conclusão deste trabalho, surgem várias oportunidades que o complementam. A seguir, temos uma lista de alguns desses trabalhos propostos:

- Realizar experimentos em outros ciclos de testes de regressão passados
- Verificação dos resultados após aplicação do método proposto em novos ciclos de regressão
- Desenvolvimento de um software para automatização do método
- Realização de estudos experimentais e estudos de casos em diferentes organizações
- Realizar estudos com uma suíte de testes que possui dependências entre seus casos de teste.
- Comparação do método proposto com os já existentes (descritos no capítulo 2)
- Prover embasamento formal para a abordagem apresentada

ANEXO I

Caso de Teste	C (f)	Percentual de Falhas	F	Severidade	S	F x S	P (f)	FER	Testes Automatizados	TAU	Tipo de Teste	TPN	PCT	Tempo de Execução
B_CT01	9	38%	1	Alta	9	9	3	27	Manual	1	Positivo	3	86	0:12:12
B_CT02	9	13%	0	Alta	9	0	1	9	Manual	1	Positivo	3	32	0:05:45
B_CT03	9	0%	0		0	0	1	9	Manual	1	Positivo	3	32	0:10:11
B_CT04	9	0%	0		0	0	1	9	Manual	1	Positivo	3	32	0:07:02
B_CT05	9	0%	0		0	0	1	9	Manual	1	Positivo	3	32	0:08:35
B_CT06	9	0%	0		0	0	1	9	Manual	1	Positivo	3	32	0:08:28
B_CT07	9	0%	0		0	0	1	9	Manual	1	Positivo	3	32	0:07:40
B_CT08	9	0%	0		0	0	1	9	Manual	1	Positivo	3	32	0:19:10
B_CT09	9	0%	0		0	0	1	9	Manual	1	Positivo	3	32	0:03:50
B_CT10	9	0%	0		0	0	1	9	Manual	1	Positivo	3	32	0:04:48
B_CT11	9	13%	0		0	0	1	9	Manual	1	Positivo	3	32	0:05:07
B_CT12	9	0%	0		0	0	1	9	Manual	1	Positivo	3	32	0:03:21
B_CT13	9	0%	0		0	0	1	9	Manual	1	Positivo	3	32	0:03:23
B_CT14	9	25%	0	Alta	9	0	1	9	Manual	1	Positivo	3	32	0:14:45
B_CT15	9	0%	0		0	0	1	9	Manual	1	Positivo	3	32	0:12:04
B_CT16	9	0%	0		0	0	1	9	Manual	1	Positivo	3	32	0:14:30
B_CT17	9	0%	0		0	0	1	9	Manual	1	Positivo	3	32	0:13:36
B_CT18	9	0%	0		0	0	1	9	Manual	1	Positivo	3	32	0:11:24
B_CT19	9	0%	0		0	0	1	9	Manual	1	Positivo	3	32	0:10:00
B_CT20	9	20%	0	Média	3	0	1	9	Manual	1	Positivo	3	32	0:08:00
B_CT21	9	0%	0		0	0	1	9	Manual	1	Positivo	3	32	0:08:00
B_CT22	3	0%	0		0	0	1	3	Automático	9	Positivo	3	30	0:09:11
B_CT23	3	0%	0		0	0	1	3	Automático	9	Positivo	3	30	0:13:59
B_CT24	3	0%	0		0	0	1	3	Automático	9	Positivo	3	30	0:09:27
B_CT25	3	0%	0		0	0	1	3	Automático	9	Positivo	3	30	0:20:30
B_CT26	3	0%	0		0	0	1	3	Automático	9	Positivo	3	30	0:17:33
B_CT27	3	0%	0		0	0	1	3	Automático	9	Positivo	3	30	0:12:17
B_CT28	3	0%	0		0	0	1	3	Automático	9	Positivo	3	30	0:08:46
B_CT29	3	0%	0		0	0	1	3	Automático	9	Positivo	3	30	0:09:05
B_CT30	3	0%	0		0	0	1	3	Automático	9	Positivo	3	30	0:18:12
B_CT31	3	0%	0		0	0	1	3	Automático	9	Positivo	3	30	0:11:05
B_CT32	3	0%	0		0	0	1	3	Automático	9	Positivo	3	30	0:17:07
B_CT33	3	0%	0		0	0	1	3	Automático	9	Positivo	3	30	0:17:10
B_CT34	3	0%	0		0	0	1	3	Automático	9	Positivo	3	30	0:26:02
B_CT35	3	0%	0		0	0	1	3	Automático	9	Positivo	3	30	0:22:31
B_CT36	3	0%	0		0	0	1	3	Automático	9	Positivo	3	30	0:17:39
B_CT37	3	0%	0		0	0	1	3	Automático	9	Positivo	3	30	0:12:00
B_CT38	3	0%	0		0	0	1	3	Automático	9	Positivo	3	30	0:08:43
B_CT39	9	0%	0		0	0	1	9	Manual	1	Negativo	1	30	0:12:06
B_CT40	9	0%	0		0	0	1	9	Manual	1	Negativo	1	30	0:10:00
B_CT41	9	0%	0		0	0	1	9	Manual	1	Negativo	1	30	0:06:45
B_CT42	9	20%	0	Média	3	0	1	9	Manual	1	Negativo	1	30	0:05:36
B_CT43	9	25%	0	Baixa	1	0	1	9	Manual	1	Negativo	1	30	0:08:00
B_CT44	9	0%	0		0	0	1	9	Manual	1	Negativo	1	30	0:08:40
B_CT45	9	20%	0	Baixa	1	0	1	9	Manual	1	Negativo	1	30	0:08:12
B_CT46	9	25%	0	Média	3	0	1	9	Manual	1	Negativo	1	30	0:08:10
B_CT47	9	0%	0		0	0	1	9	Manual	1	Negativo	1	30	0:09:36
B_CT48	9	20%	0	Média	3	0	1	9	Manual	1	Negativo	1	30	0:07:00
B_CT49	9	0%	0		0	0	1	9	Manual	1	Negativo	1	30	0:05:45
B_CT50	9	0%	0		0	0	1	9	Manual	1	Negativo	1	30	0:08:15
B_CT51	9	0%	0		0	0	1	9	Manual	1	Negativo	1	30	0:03:15
B_CT52	9	0%	0		0	0	1	9	Manual	1	Negativo	1	30	0:05:15
B_CT53	9	20%	0	Baixa	1	0	1	9	Manual	1	Negativo	1	30	0:07:10
B_CT54	3	0%	0		0	0	1	3	Automático	9	Negativo	1	28	0:12:23
B_CT55	3	0%	0		0	0	1	3	Automático	9	Negativo	1	28	0:17:22
B_CT56	3	0%	0		0	0	1	3	Automático	9	Negativo	1	28	0:19:57
B_CT57	3	0%	0		0	0	1	3	Automático	9	Negativo	1	28	0:12:59
B_CT58	3	0%	0		0	0	1	3	Automático	9	Negativo	1	28	0:16:15
B_CT59	3	0%	0		0	0	1	3	Automático	9	Negativo	1	28	0:18:20
B_CT60	3	0%	0		0	0	1	3	Automático	9	Negativo	1	28	0:22:24
B_CT61	3	0%	0		0	0	1	3	Automático	9	Negativo	1	28	0:15:01
B_CT62	3	0%	0		0	0	1	3	Automático	9	Negativo	1	28	0:09:45
B_CT63	3	0%	0		0	0	1	3	Automático	9	Negativo	1	28	0:04:51
B_CT64	1	0%	0		0	0	1	1	Automático	9	Positivo	3	24	0:09:37
B_CT65	1	0%	0		0	0	1	1	Automático	9	Positivo	3	24	0:08:25
B_CT66	1	0%	0		0	0	1	1	Automático	9	Positivo	3	24	0:09:36
B_CT67	1	0%	0		0	0	1	1	Automático	9	Positivo	3	24	0:08:16
B_CT68	1	0%	0		0	0	1	1	Automático	9	Positivo	3	24	0:11:47
B_CT69	1	0%	0		0	0	1	1	Automático	9	Positivo	3	24	0:04:50
B_CT70	1	0%	0		0	0	1	1	Automático	9	Positivo	3	24	0:01:54
B_CT71	1	0%	0		0	0	1	1	Automático	9	Negativo	1	22	0:08:33
B_CT72	1	0%	0		0	0	1	1	Automático	9	Negativo	1	22	0:08:17
B_CT73	1	0%	0		0	0	1	1	Automático	9	Negativo	1	22	0:09:09
B_CT74	1	0%	0		0	0	1	1	Automático	9	Negativo	1	22	0:19:38
B_CT75	1	0%	0		0	0	1	1	Automático	9	Negativo	1	22	0:05:11
B_CT76	1	8%	0	Baixa	1	0	1	1	Automático	9	Negativo	1	22	0:07:04
B_CT77	1	0%	0		0	0	1	1	Automático	9	Negativo	1	22	0:08:00
B_CT78	1	0%	0		0	0	1	1	Automático	9	Negativo	1	22	0:02:01
B_CT79	1	0%	0		0	0	1	1	Automático	9	Negativo	1	22	0:01:51
B_CT80	1	0%	0		0	0	1	1	Automático	9	Negativo	1	22	0:01:43
B_CT81	1	0%	0		0	0	1	1	Automático	9	Negativo	1	22	0:02:11
B_CT82	1	0%	0		0	0	1	1	Automático	9	Negativo	1	22	0:01:31
B_CT83	1	0%	0		0	0	1	1	Automático	9	Negativo	1	22	0:02:10

Referências

- [1] Viana, Virginia M. A. Um Método para Seleção de Testes de Regressão para Automação. Dissertação de Mestrado, Universidade Federal de Pernambuco, 2006.
- [2] AMLAND, Ståle. Risk Based Testing and Metrics, 5th International Conference EuroSTAR '99, November 8 - 12, 1999, Barcelona, Spain.
- [3] CHEN, Yanping, Specification-based Regression Test Selection with Risk Analysis, Dissertação de Mestrado, University of Ottawa, Canada, December 2002.
- [4] NYMAN, Jeff, Positive and Negative Testing, GlobalTester, TechQA, <http://www.sqatester.com/methodology/PositiveandNegativeTesting.htm>, último acesso: 15/05/2008.
- [5] Desikan, Srinivasan, A test methodology for an effective regression testing, www.stickyminds.com, acesso em 15/05/2008.
- [6] Project Management Institute (2004), A Guide to the Project Management Body of Knowledge (PMBOK Guide).
- [7] Walter Gorlitz, Defects Severity and Priority, <http://www.sqatester.com/bugsfixes/defectparameters.htm>, ultimo acesso: 18/05/2008.
- [8] CHITTIMALLI, Pavan K., HARROLD, Mary J., Regression Test Selection on System Requirements, Proceedings of the 1st conference on India software engineering conference, ISEC '08, February 2008, Hyderabad, India.
- [9] AGRAWAL, H., HORGAN, J., KRAUSER, E. and LONDON, S., Incremental Regression Testing, Proc. Conf. Software Maintenance—1993, Sept. 1993.
- [10] H. K. N. Leung and L. White. Insights Into Regression Testing. In Proceedings of the Conference on Software Maintenance, October 1989.
- [11] B. Beizer. Software Testing Techniques. Van Nostrand Reinhold, 1990.
- [12] BURNSTEIN, Ilene. Practical Software Testing. Springer Professional Computing.
- [13] SWEBOK - Guide to the Software Engineering Body of Knowledge, 2004.
- [14] ROTHERMEL, Gregg; ELBAUM, Sebastian; MALISHEVSKY, Alexey; KALLAKURI, Praveen; QIU, Xuemei. On Test Suite Composition and Cost-

Effective Regression Testing, Agosto de 2003

[15] ELBAUM S., MALISHEVSKY A., ROTHERMEL G. Prioritizing test cases for regression testing. In Proceedings of the International Symposium on Software Testing and Analysis. Agosto de 2001.

[16] PMBOK 2004, Um Guia do Conjunto de Conhecimentos em Gerencia de Projetos. Edição 2004.

[17] JØRGENSEN, Lars Kristoffer Ulstein. A Software Tool for Risk-based Testing. Autumn 2004.

[18] Paul Gerrard, Neil Thompson. Risk-Based E-Business Testing, 2002.

[19] BACH, James. Heuristic Risk-Based Testing. Software Testing and Quality Engineering Magazine, 11/99.

[20] BESSON, Stephane. A Strategy for Risk-Based Testing, <http://www.stickyminds.com>, último acesso: 08/06/2008.

[21] SCHAEFER Hans. Strategies for Prioritizing Tests against Deadlines Risk Based Testing. Software Test Consulting, Noruega, <http://home.c2i.net/schaefer/testing/risktest.doc> , acesso em 08/06/2008.

[22] HARROLD, Mary Jean; JONES, James A.; LI, Tongyu; LIANG, Donglin; Regression Test Selection for Java Software, Proceedings of the ACM Conference on OO Programming, Systems, Languages, and Applications (OOPSLA '01), 2001.

[23] G. Rothermel and M. J. Harrold. Empirical studies of a safe regression test selection technique. IEEE Transactions on Software Engineering, June 1998.

[24] G. Rothermel, M.J. Harrold, and J. Dedhia. Regression test selection for C++ programs. Journal of Software Testing, Verification, and Reliability, June 2000.

[25] Y.F. Chen, D.S. Rosenblum, and K.P. Vo. TestTube: A system for selective regression testing. In Proceedings of the 16th International Conference on Software Engineering, May 1994.

[26] G. Rothermel, M.J. Harrold, J. Ostrin, and C. Hong. An empirical study of the effects of minimization on the fault detection capabilities of test suites. In Proceedings of the International Conference on Software Maintenance, November 1998.

[27] W. E. Wong, J. R. Horgan, S. London, and A. P. Mathur. Effect of test set minimization on fault detection effectiveness. *Software Practice and Experience*, April 1998.

[28] Durante, Felipe. Qualidade como Subproduto do Negócio. *Testexpert*. <http://www.testexpert.com.br/?q=node/330>. Último acesso: 18/06/2008.