Um Mapeamento Sistemático Sobre Ensino de Teste de Software

Pedro Henrique D. Valle[1], **Ellen F. Barbosa**[1], **José C. Maldonado**[1]

[1]Instituto de Ciências Matemáticas e de Computação (ICMC/USP)
São Carlos/SP, Brasil, 13560-970

{pedrohenriquevalle@usp.br, francine@icmc.usp.br, jcmaldon@icmc.usp.br}

Abstract. Context*: Software testing is an important activity to ensure quality for software products. However, there is a lack of qualified professionals and a lack of motivation to work with software testing.* ***Objective:*** *To identify the state of art about teaching software testing.* ***Method****: We performed a systematic mapping based on digital libraries and manual search.* ***Results****: We identified the main approaches of teaching software testing, as well as how to develop and evaluate them. Futhermore, we idenfied the languages addressed to teaching and the testing phases considered in these approaches.* ***Conclusion****: We characterized the state of art about teaching software testing approaches, observing that the most used ones are educational games and teaching software testing combined programming.*

Resumo. Contexto*: Teste é uma atividade importante na garantia da qualidade de produtos de software. No entanto, ainda há uma grande carência de profissionais qualificados e uma desmotivação para trabalhar com teste de software.* ***Objetivo:*** *Identificar o estado da arte referente ao ensino de teste de software.* ***Método:*** *Para isso, realizou-se a condução de um mapeamento sistemático utilizando máquinas de busca e a busca manual.* ***Resultados:*** *Identificaram-se as principais abordagens para o ensino de teste de software, bem como a forma de desenvolvê-las e avaliá-las. Além disso, identificaram-se as linguagens alvo e as fases de teste de software consideradas nessas abordagens.* ***Conclusões:*** *Caracterizou-se o estado da arte referente ao ensino de teste, observando que as abordagens mais utilizadas são jogos educacionais e ensino de teste com programação.*

1. Introdução

O teste de software tem por objetivo executar programas ou produtos intermediários com entradas específicas analisando se esses produtos comportam-se de acordo com o esperado. O teste envolve basicamente quatro fases: o planejamento de teste, o projeto de casos de teste, a execução de testes e a avaliação dos resultados dos testes [Delamaro et al. 2007].

No entanto, a indústria de teste de software defronta-se com a carência de mão-de-obra especializada nessa área [de Souza et al. 2012]. Isso deve-se, sobretudo, à dificuldade e ineficiência em ensinar teste de software por meio de aulas teóricas, pois ainda se fazem necessários ambientes práticos que motivem os estudantes a realizarem atividades relacionadas com o teste de software. Outro fator que pode contribuir para a

DOI: 10.5753/cbie.sbie.2015.71

carência de profissionais qualificados é a desvalorização dos profissionais de teste que, em geral, são pouco reconhecidos pelo trabalho realizado [Smith et al. 2012].

Diante desse cenário, percebe-se que é imprescindível a utilização de diferentes abordagens para auxiliar o processo de ensino e aprendizagem de teste de software. Nesse contexto, entende-se como abordagem as metodologias e os instrumentos utilizados para auxiliar o ensino de algum conteúdo, em um dado domínio de conhecimento.

O objetivo deste trabalho é identificar o estado da arte referente ao ensino de teste de software. Para isso, serão analisadas por meio de um mapeamento sistemático quais são as abordagens utilizadas para auxiliar o ensino de teste, as fases de teste de software contempladas no ensino dessa atividade, as tecnologias utilizadas para o desenvolvimento das abordagens, bem como as linguagens alvo e, por fim, verificar como foram realizadas as avaliações das abordagens propostas para auxiliar o ensino de teste de software.

De acordo com os resultados obtidos neste trabalho, o ensino conjunto de teste de software com programação e jogos educacionais são as duas abordagens mais utilizadas para auxiliar o ensino de teste de software. Essa caracterização do estado da arte referente ao ensino de teste poderá auxiliar na identificação e no direcionamento de passos futuros para a realização de novas pesquisas relacionadas com esse tema, pois algumas novas direções podem ser seguidas a partir dos resultados obtidos neste mapeamento.

Este trabalho está organizado da seguinte forma: Na Seção 2 apresenta-se o planejamento deste mapeamento sistemático. Na Seção 2.1 apresentam-se as informações sobre a coleta dos dados, demonstrando a quantidade de trabalhos selecionados para análise e o ano de publicação dos mesmos. Na Seção 2.2 apresentam-se as respostas para as questões de pesquisa. Na Seção 3 discutem-se as ameaças à validade deste estudo. Na Seção 4 apresentam-se os principais conceitos das duas abordagens mais utilizadas para auxiliar o ensino de teste de software. Por último, na Seção 5, apresentam-se as considerações finais e os trabalhos futuros.

2. Planejamento do Mapeamento Sistemático

O mapeamento sistemático é um tipo de pesquisa muito utilizada quando há um cenário abrangente e que tem por objetivo reunir o máximo de informações disponíveis sobre uma determinada área do conhecimento [Kitchenham e Charters 2007]. Para a condução deste trabalho realizou-se um planejamento do mapeamento sistemático que contém as questões de pesquisa, *string* de busca, estratégias de busca, critérios de inclusão e exclusão. Devido à limitação de espaço, o protocolo deste mapeamento sistemático e os trabalhos selecionados para análise estão disponíveis em: (goo.gl/dTmSRR).

As questões de pesquisa para a realização do Mapeamento Sistemático são as seguintes:

- **Q1** - Quais são os tipos de abordagens que têm sido utilizadas para auxiliar o ensino teste de software?
- **Q2** - Quais são as fases de teste de software que têm sido contempladas no ensino de teste de software?
- **Q3** - Quais são tecnologias que têm sido utilizadas no desenvolvimento das abordagens identificadas na Q1 e quais são as linguagens alvo utilizadas para auxiliar o ensino de teste de software?

- **Q4** - Quais são as avaliações que têm sido realizadas para a validação das abordagens utilizadas para apoiar o ensino de teste de software?

Na Tabela 1 apresenta-se a *string* de busca utilizada neste mapeamento sistemático.

Tabela 1. *String* de busca genérica gerada para o Mapeamento Sistemático

Idioma	*String* de Busca
Inglês	(“teaching” OR “learning” OR “training” OR “education”) AND (“software testing” OR “software test” OR “software evaluation”)

Para a inclusão dos trabalhos na análise, utilizaram-se os seguintes critérios de inclusão: (i) Os trabalhos devem estar escritos em inglês ou português; (ii) Os trabalhos devem conter as palavras chaves utilizadas na *string* de busca, no resumo e/ou título e/ou nas palavras-chave do artigo selecionado; e (iii) Os trabalhos devem apresentar abordagens que auxiliem o ensino de teste de software. A negação dos critérios apresentados anteriormente foi utilizada para exclusão dos trabalhos encontrados. O conjunto completo de critérios de exclusão está disponível em: (goo.gl/dTmSRR).

2.1. Coleta dos Dados

Na Tabela 2 apresenta-se a quantidade de trabalhos selecionados em cada passo de acordo com o planejamento estabelecido. É importante observar que grande parte dos trabalhos selecionados para análise foram identificados a partir das bases de dados IEEE Xplore, ACM e também pela busca manual. A busca manual foi realizada já que trabalhos importantes ou recentes poderiam não ter sido considerados na análise.

Base de Dados	**Passo 1**	**Passo 2**	**Passo 3**
Scopus	105	33	4
ACM	47	12	5
Compendex	228	36	3
IEEE	360	22	5
Busca Manual	—	—	8
TOTAL	740	103	25

Tabela 2. Quantidade de Trabalhos Selecionados

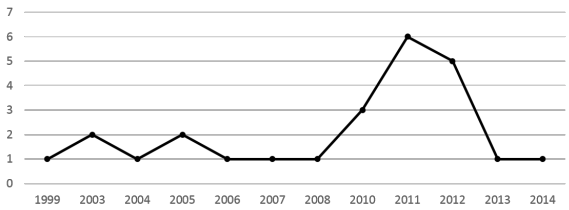

Figura 1. Estudos por Ano

Na Figura 1 apresenta-se a distribuição temporal dos trabalhos analisados neste mapeamento sistemático. Mesmo não havendo padrão ou tendência clara sobre a distribuição dos trabalhos ao longo dos anos, observa-se que houve um aumento no interesse dos pesquisadores com relação ao processo de ensino de aprendizagem de conteúdos relacionados com o teste de software a partir do ano de 2010.

A maior parte dos trabalhos selecionados foram publicados nas seguintes conferências: Software Engineering Education and Training (CSEE&T), Frontiers in Education Conference (FIE), Technical Symposium on Computer Science Education (SIGCSE), International Conference on Software Engineering (ICSE), Fórum de Educação em Engenharia de Software (FEES) e Simpósio Brasileiro de Informática na Educação (SBIE).

2.2. Análise dos Resultados

Por meio das respostas obtidas realizou-se a caracterização do estado da arte referente ao ensino de teste de software. É importante ressaltar que essa caracterização poderá auxiliar no desenvolvimento de novas pesquisas, pois há algumas decisões que podem ser tomadas, a partir dos resultados obtidos neste mapeamento sistemático, para o desenvolvimento de novos trabalhos que tenham como objetivo criar novas abordagens e/ou ambientes que auxiliem o ensino de teste de software.

2.2.1. Respostas para a Questão 1

A primeira questão de pesquisa diz respeito sobre quais são os tipos de abordagens utilizadas para auxiliar o ensino teste de software. Na Figura 2 apresenta-se um gráfico de bolha que ilustra o cenário do ensino de teste de software, apresentando as principais abordagens utilizadas, o método de validação dessas abordagens e as técnicas de teste de software contempladas em cada uma das abordagens.

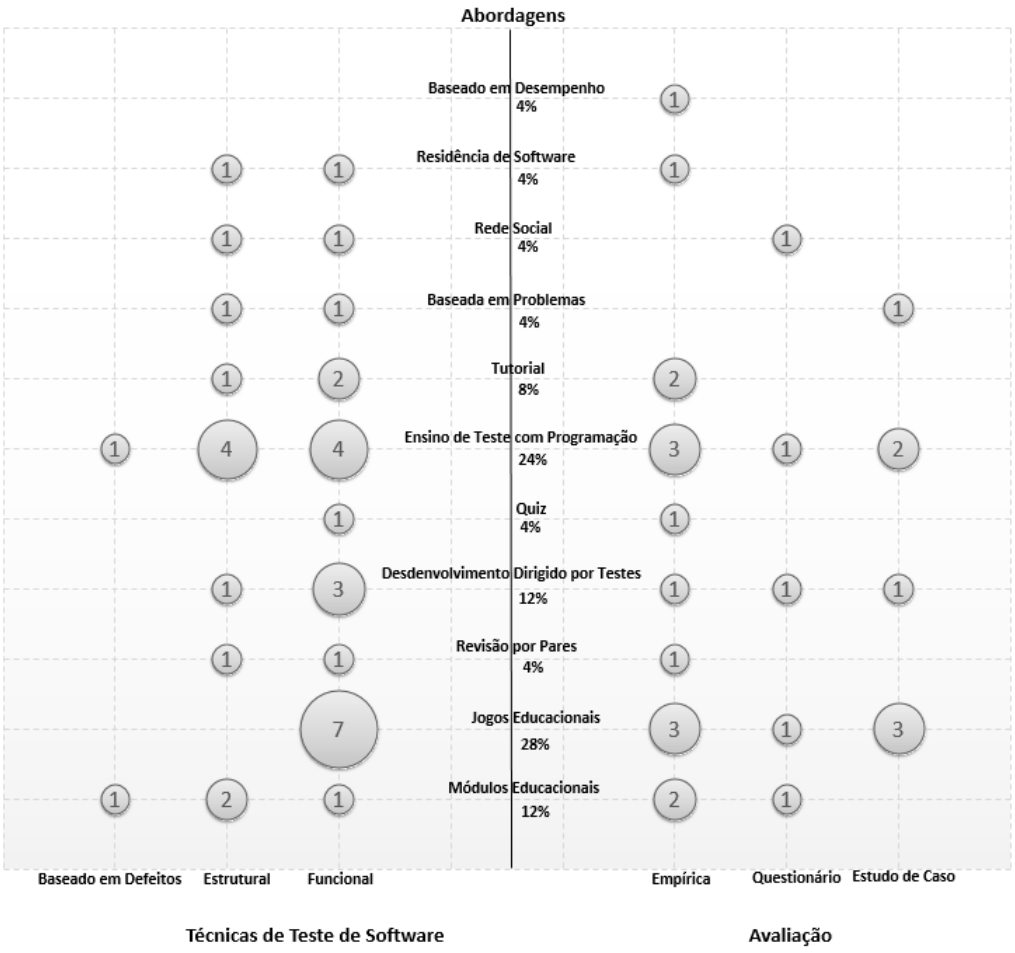

Figura 2. Abordagens para Auxiliar o Ensino de Teste de Software

A seguir apresentam-se as descrições das abordagens identificadas para auxiliar o ensino de teste de software.

- **Módulos Educacionais:** São unidades concisas de estudo, compostas por conteúdos teóricos combinados com atividades práticas e avaliações, apoiadas por recursos tecnológicos e computacionais [Barbosa e Maldonado 2011];
- **Jogos Educacionais:** São jogos que proporcionam práticas educacionais atrativas e inovadoras, nos quais os usuários podem aprender de forma mais ativa, dinâmica e motivadora [Farias et al. 2012];
- **Ensino Conjunto de Teste com Programação:** É o ensino conjunto de conceitos básicos de programação e de teste de software. Muitas pesquisas demonstram que o ensino conjunto dessas disciplinas possui benefícios [de Souza et al. 2012];
- **Quiz:** É uma forma interativa de avaliar o conhecimento de usuários, por meio de questionários [Mustakerov e Borissova 2005];
- **Revisão por Pares:** Utilizar essa técnica para auxiliar o ensino de teste de software, permite um aprendizado lúdico e competitivo, no qual os estudantes aprendem uns com os outros [Smith et al. 2012];
- **Desenvolvimento Dirigido por Testes (TDD):** Nessa técnica o desenvolvedor escreve um caso de teste e em seguida produz um código que possa ser validado pelo teste [Edwards 2003];
- **Tutorial:** É uma ferramenta de ensino e aprendizagem que pode ser um programa de computador ou um texto que contém ou não imagens que auxiliam no processo de aprendizagem, demonstrando o passo a passo para a realização de alguma atividade [Liu et al. 2010];
- **Rede Social:** Essa abordagem permite que usuários comuniquem-se uns com outros trocando experiências no decorrer do processo de aprendizagem [Clarke et al. 2011];
- **Modelo de Residência de Software:** Essa abordagem inclui o ensino tradicional de conceitos relevantes de um determinado conteúdo e em seguida a realização de atividades práticas de com profundidade e/ou com especialização em algum assunto específico [Sampaio et al. 2005];
- **Aprendizagem Baseada em Problemas:** Essa abordagem permite que os estudantes trabalhem em equipe com o intuito de resolverem problemas, incentivando o desenvolvimento de habilidades como atitudes, auto iniciativa e cooperação [Figuerêdo et al. 2011];
- **Aprendizagem Baseada em Desempenho:** Essa abordagem utiliza a medição do desempenho dos usuários para esclarecer os principais objetivos e demonstrar as necessidades da aprendizagem individual do usuário [Wang et al. 2011];

Foram identificadas 11 abordagens diferentes para auxiliar o ensino de teste de software. As duas abordagens mais utilizadas são os **Jogos Educacionais** e o **Ensino de Teste com Programação**, as quais são discutidas na Seção 4. Juntas essas abordagens foram utilizadas em mais de 50% dos trabalhos analisados. É importante ressaltar que alguns trabalhos utilizaram mais de uma abordagem para apoiar o ensino de teste de software. A partir da Figura 2 percebe-se que o Teste Funcional é a técnica mais abordada.

2.2.2. Respostas para a Questão 2

A segunda questão de pesquisa diz respeito sobre quais são as fases de teste de software contempladas no ensino de teste de software. Na Figura 3 ilustram-se as fases de teste de software identificadas e as porcentagens de trabalhos que as contemplaram.

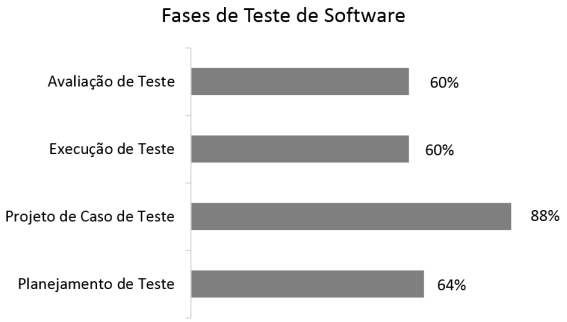

Figura 3. Fases de teste de software contempladas nos trabalhos para auxiliar o ensino de teste

Como pode ser observado, aproximadamente 60% dos trabalhos analisados abordaram conteúdos relacionados com todas as fases de teste de software. Nos trabalhos analisados, a fase de projeto de caso de teste é a fase mais abordada no ensino dos conteúdos relacionados com o teste de software, sendo utilizado em aproximadamente 84% dos trabalhos selecionados. Além disso, há trabalhos que abordam apenas uma fase de teste de software como o trabalho proposto por [Farias et al. 2012], que propôs um jogo educacional para auxiliar o ensino de teste de software, especificamente para a fase de planejamento de teste.

2.2.3. Respostas para a Questão 3

A terceira questão de pesquisa diz respeito sobre quais são as tecnologias utilizadas no desenvolvimento das abordagens identificadas na Q1 e quais são as linguagens alvo utilizadas para auxiliar o ensino de teste de software, ou seja, as linguagens de programação utilizadas para escrever os códigos a serem testados.

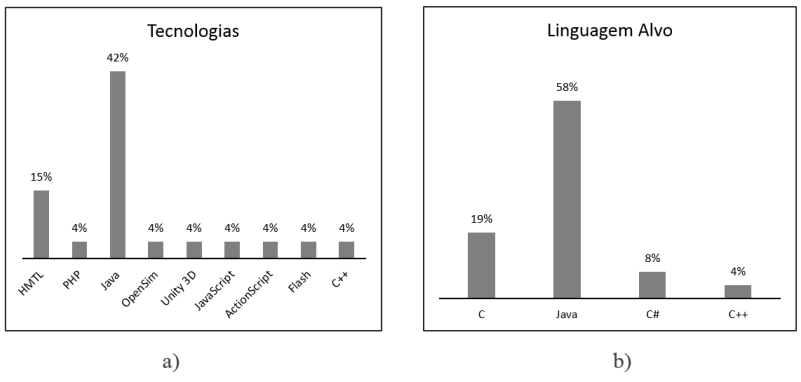

Figura 4. a) Linguagens Alvo e b) Linguagens para o desenvolvimento das abordagens

Na Figura 4 a) observam-se as 9 tecnologias identificadas para auxiliar o desenvolvimento das abordagens identificadas na Q1, e as porcentagens de trabalhos que as utilizaram. A linguagem de programação Java foi a mais utilizada, sendo que aproximadamente 48% dos trabalhos a utilizaram. Isso pode estar relacionado com o fato

da linguagem Java ser bem conhecida entre os desenvolvedores de produtos de software e por prover a característica de portabilidade.

A segunda tecnologia mais utilizada foi a linguagem HTML, sendo utilizada em 15% dos trabalhos analisados. Por fim, as tecnologias PHP, OpenSim, Unity 3D, JavaScript, Flash e C++ foram utilizadas em aproximadamente 4% dos trabalhos analisados cada uma. É importante observar que alguns trabalhos utilizaram mais de uma linguagem de programação para o desenvolvimento das abordagens propostas. Por outro lado, outros não deixaram claro quais linguagens de programação foram utilizadas.

Na Figura 4 b) observam-se as linguagens alvo utilizadas pelas abordagens identificadas na Q1 para auxiliar o ensino de teste de software, a saber: C, Java, C# e C++, e a porcentagem de trabalhos que as abordaram. Aproximadamente 58% dos trabalhos analisados utilizaram a linguagem de programação Java como linguagem alvo para auxiliar o ensino de teste de software. Isso pode estar relacionado com o fato da linguagem Java ser bem disseminada no meio acadêmico e na indústria e também por haver muitas ferramentas de teste de software que auxiliam o teste de códigos escritos nessa linguagem. As outras linguagens alvo identificadas, tais como: C, C# e C++, foram utilizadas em 19%, 8%, 4% dos trabalhos analisados, respectivamente.

2.2.4. Respostas para a Questão 4

A quarta questão de pesquisa diz respeito sobre quais os tipos de avaliações que foram realizadas para validação das abordagens propostas para auxiliar o ensino de teste de software. Na Figura 5 ilustram-se os tipos de avaliações identificadas e a porcentagem de trabalhos que as utilizaram.

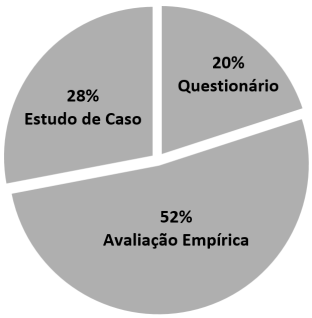

Figura 5. Avaliações realizadas para validação das abordagens propostas

As avaliações realizadas por meio de questionários foram utilizadas em 20% dos trabalhos analisados, como o trabalho proposto [de Oliveira 2013] que realizou a avaliação do Jogo Educacional proposto por meio de questionários com 127 avaliadores (jogadores). As avaliações por meio de estudo de caso correspondem a 28% dos trabalhos analisados. Porém, poucos detalhes sobre a realização dos estudos de caso foram disponibilizados. A avaliação empírica foi o tipo de avaliação mais utilizada para validação das abordagens propostas, sendo utilizada por 52% dos trabalhos. Nesse tipo de avaliação foram observadas as mudanças de atitude dos estudantes após a utilização das abordagens propostas, não havendo uma avaliação formal e sistemática. Esse tipo de

avaliação é interessante para avaliar o comportamento dos alunos ao utilizar as abordagens propostas.

3. Ameaças à Validade

Neste trabalho foram identificados três tipos de ameaças à validade que são:

1. **Validade Externa**: Refere-se à capacidade de generalizar os resultados do experimento. Apesar da pequena quantidade de trabalhos contemplados nesse mapeamento sistemático, os trabalhos contemplados são considerados relevantes na área.
2. **Validade Interna**: Refere-se à capacidade de tirar conclusões sobre os resultados obtidos. A busca automatizada dos trabalhos pode ser classificada neste tipo de ameaça, pois há a possibilidade de estudos importantes ou recentes não serem considerados. Para minimizar esse problema foram realizadas buscas manuais nos principais anais de congressos da área e buscas no Google Scholar.
3. **Validade de Construção**: Está relacionada, em geral, com a má definição da base teórica ou com a definição do processo de experimentação. A eficiência da *string* de busca pode ser classificada neste tipo ameaça. A validação da *string* de busca utilizada neste trabalho foi realizada por meio de trabalhos controle. Além disso, formularam-se *strings* de busca alternativas com outros sinônimos das palavras utilizadas.

4. Ensino de Teste de Software com Programação e Jogos Educacionais

A partir dos resultados obtidos, observa-se que as duas abordagens mais utilizadas para auxiliar o ensino de teste de software são: i) jogos educacionais; e ii) ensino de teste de software com programação. O ensino de teste de software é pouco explorado no contexto de cursos de graduação, sendo pouco abordado na maioria dos cursos de computação e apenas ensinado no final da graduação. Entretanto, pesquisas indicam que o teste de software deve ser ensinado o mais cedo possível para que os estudantes desenvolvam suas habilidades de compreensão e análise [Edwards 2004, de Souza et al. 2012, de Souza et al. 2014].

Para resolver esses problemas, algumas iniciativas foram investigadas, dentre elas destaca-se o ensino de teste de software com programação. Essa abordagem sugere que o teste de software seja ensinado em paralelo com fundamentos de programação. O ensino conjunto dessas disciplinas pode ajudar o desenvolvimento de habilidades de compreensão e análise nos estudantes, pois para testar os estudantes precisam entender o comportamento de seus programas. Neste mapeamento sistemático identificaram-se algumas ferramentas para auxiliar o ensino de teste de software com programação, tais como: Web-CAT [Edwards 2003] e ProgTest [de Souza et al. 2012].

Jogos educacionais, por sua vez, têm sido muito utilizados para auxiliar o ensino de teste de software. Muitos pesquisadores afirmam que essas ferramentas podem facilitar a aprendizagem de conceitos e ideias, já que jogos motivam e proporcionam prazeres aos jogadores. Sendo assim, os usuários são incentivados a dedicarem parte de seu tempo e de seus esforços para vencerem os obstáculos propostos [Neto e da Fonseca 2013]. Portanto, percebe-se que os jogos educacionais possibilitam que seus usuários obtenham conhecimentos combinados com a diversão. Alguns jogos educacionais no domínio

de teste de software foram identificados neste mapeamento sistemático, tais como: *iTest Learning* [Farias et al. 2012], Jogo das 7 Falhas [Diniz e Dazzi 2011], TestEG [de Oliveira 2013], entre outros.

Os trabalhos analisados que utilizaram os jogos educacionais para auxiliar o ensino de teste de software contemplaram apenas a técnica de Teste Funcional. Desta forma, seria interessante a criação de jogos educacionais que abordassem conteúdos relacionados com o Teste Estrutural e o Baseado em Defeitos. Portanto, seria pertinente criar uma ferramenta que combine o ensino de teste de software com programação e jogos educacionais para apoiar o ensino de teste de software incremental. Uma alternativa para isso seria a criação de um jogo educacional que abordasse conteúdos de teste de software em conjunto com fundamentos de programação.

5. Conclusões e Trabalhos Futuros

Por meio do mapeamento sistemático realizado neste trabalho, observou-se que há uma carência de trabalhos que abordam o ensino de teste de software. Para a realização dessa análise foram selecionados 25 trabalhos. A partir dos estudos selecionados identificaram-se 11 abordagens diferentes para auxiliar o ensino de teste de software, dentre elas: jogos educacionais e ensino de teste com programação, as mais utilizadas.

Aproximadamente 60% dos trabalhos analisados contemplaram as quatro fases de teste de software. Para uma melhor formação de profissionais de teste seria interessante que uma maior quantidade de estudos abordassem todas essas fases, pois assim os mesmos contemplaram o ensino de teste de forma geral. Para o desenvolvimento das abordagens identificadas foram utilizadas diferentes tecnologias, sendo que a linguagem Java foi utilizada em 42% dos trabalhos analisados. Além disso, a mesma foi utilizada em 58% dos trabalhos como linguagem alvo. Para validação das abordagens a avaliação empírica foi a mais utilizada, aproximadamente de 52% dos trabalhos a utilizaram. Porém, poucos detalhes foram apresentados sobre a condução dessas avaliações.

Neste cenário, percebe-se uma carência de ambientes virtuais que motivem o ensino de teste de software. Sendo assim, como trabalhos futuros pretende-se desenvolver um jogo educacional que aborde o ensino conjunto de teste de software com programação. Neste cenário o mapeamento sistemático desenvolvido contribuiu para identificar os principais elementos para o desenvolvimento desse jogo. Esse jogo educacional contemplaria as quatro fases de teste. A linguagem Java poderia ser utilizada para o desenvolvimento do jogo, bem como linguagem alvo. Por fim, para a validação seria realizada uma avaliação empírica por meio de experimentos controlados.

6. Agradecimentos

Os autores agradecem à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES)(PROEX-9035919/M) pelo apoio concedido a este trabalho.

Referências

Barbosa, E. e Maldonado, J. (2011). *E-Infrastructures and Technologies for Lifelong Learning: Next Generation Environments*, chapter Collaborative development of educational modules: a need for lifelong learning.

Clarke, P. J., Pava, J., Wu, Y., e King, T. M. (2011). Collaborative web-based learning of testing tools in se courses. In *Technical symposium on Computer science education*.

de Oliveira, B. C. (2013). Testeg-um software educacional para o ensino de teste de software. In *Universidade Federal de Lavras*.

de Souza, D. M., da Silva Batista, M. H., e Barbosa, E. F. (2014). Avaliação de qualidade de um ambiente de apoio ao ensino de programação. *Revista Novas Tecnologias na Educação*.

de Souza, D. M., Maldonado, J. C., e Barbosa, E. F. (2012). Aspectos de desenvolvimento e evolução de um ambiente de apoio ao ensino de programação e teste de software. In *Simpósio Brasileiro de Informática na Educação*.

Delamaro, M. E., Maldonado, J. C., e Jino, M. (2007). *Introdução ao Teste de Software*. Elsevier.

Diniz, L. L. e Dazzi, R. L. (2011). Jogo para o apoio ao ensino do teste de caixa-preta. In *Simpósio Brasileiro de Informática na Educação*.

Edwards, S. H. (2003). Teaching software testing: Automatic grading meets test-first coding. In *Conference on Object-oriented Programming, Systems, Languages, and Applications*. ACM.

Edwards, S. H. (2004). Using software testing to move students from trial-and-error to reflection-in-action. In *Technical Symposium on Computer Science Education*. ACM.

Farias, V., Moreira, C., Coutinho, E., e Santos, I. S. (2012). itest learning: Um jogo para o ensino do planejamento de testes de software. In *Fórum de Educação em Engenharia de Software*. Simpósio Brasileiro de Engenharia de Software.

Figuerêdo, C. d. O., dos Santos, S. C., Borba, P. H., e Alexandre, G. H. (2011). Using pbl to develop software test engineers. In *International Conference on Computers and Advanced Technology in Education*.

Kitchenham, B. e Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical report, Keele University and Durham University Joint Report.

Liu, H., Kuo, F.-C., e Chen, T. Y. (2010). Teaching an end-user testing methodology. In *Software Engineering Education and Training*.

Mustakerov, I. e Borissova, D. (2005). A conceptual approach for development of educational web-based e-testing system. *Expert Systems with Applications*.

Neto, J. F. B. e da Fonseca, F. d. S. (2013). Jogos educativos em dispositivos móveis como auxílio ao ensino da matemática. *Revista Novas Tecnologias na Educação*, 11.

Sampaio, A., Albuquerque, C., Vasconcelos, J., Cruz, L., Figueiredo, L., e Cavalcante, S. (2005). Software test program: a software residency experience. In *International Conference on Software Engineering*.

Smith, J., Tessler, J., Kramer, E., e Lin, C. (2012). Using peer review to teach software testing. In *International Conference on International Computing Education*. ACM.

Wang, M., Jia, H., Sugumaran, V., Ran, W., e Liao, J. (2011). A web-based learning system for software test professionals. *IEEE Transactions on Education*.