

gTest Learning: Um Jogo para Ensino Básico de Teste de Software

Cleibson Gomes Silva¹, Ivaldir de Farias Junior²,
Marcelo Teixeira³, Cristiane de Aquino³

¹Unibratec – Recife - PE

²Faculdade Estácio Recife – Recife – PE

³UFRPE - Recife - PE

{Cleibongomessilva, ivaldirjr, marcelo.ufrpe.br, crisaquino.minds}
@gmail.com

Abstract. *Today, games are part of the daily lives of most people. There has been an effort to extract more of the games and end the old idea that they serve only as a hobby, making them learning tools. Using techniques that guarantee fun and motivation, it is possible to provide the players a satisfactory learning process through the game. Thus, the purpose of gTest Learning is to develop a 2D platform game based on the classic Mario Word, whose purpose is to run small fixation tests for the player. These challenges are designed to force the search for artifacts that will help in solving the problems, focusing on software testing based on the ISTQB Foundation Certification.*

Resumo. *Hoje, os games fazem parte do cotidiano da maioria das pessoas. Têm-se buscado extrair mais dos games acabando com a ideia que eles servem apenas para passar o tempo, tornando-os ferramentas de aprendizagem. Usando técnicas que garantem diversão e motivação, é possível proporcionar ao jogador um processo de aprendizado satisfatório através do jogo. Sendo assim, o objetivo do gTest Learning é desenvolver um game de plataforma 2D baseado no clássico Mario Word, cuja finalidade é executar pequenos testes de fixação para o jogador. Estes desafios têm como objetivo forçar a busca por artefatos que venham a ajudar na resolução dos problemas, focando em testes de software com base na certificação Foundation do ISTQB.*

1. Introdução

O desenvolvimento de jogos para ensino e treino de conhecimentos vem sendo bastante investigado, nas mais vastas áreas. Um alvo comum para este tipo de abordagem é o nível inicial de escolaridade onde pode-se motivar e segurar a atenção das crianças para aprendizagem [Tânia 2014], [Kapp 2012]. Também existem exemplos de jogos para adultos nas mais diversas áreas como turismo, saúde, defesa, gestão de empresas e etc. eles são chamados de jogos sérios, buscando transmitir um conteúdo de formação educativa [Tânia 2014], [Kapp 2012].

Jogos sérios foram desenvolvidos para o ensino ou para prática dos conhecimentos técnicos essenciais ao desenvolvimento profissional, incluindo

elementos de entretenimento para segurar a atenção e incentivar o envolvimento do utilizador na aprendizagem.

É preciso ter uma grande gama de técnicas e conceitos para se ensina testes de software, O cenário encontrado pelos profissionais da área de testes de software não é fácil de simular em um ambiente de ensino [Fernandes e Sousa 2010]. E mesmo simulando os estudantes não conseguem destingir como e quando devem aplicá-los no mundo "real". A falta de conhecimento prático mesmo no final da formação acadêmica, não considerada com suficiente para os empregadores contratá-los [Fernandes e Sousa 2010].

O uso de jogos para aprendizagem desta matéria tem funcionado como uma boa alternativa, visto que em várias experiências comprovam que o aluno consegue assimilar o que realmente lhe é ensinado, no jogo o aluno deixa do lado os seu medos e ansiedades, facilitando a concentração que aumenta a quantidade de informações assimiladas [Schneiderman 2005].

O gTest Learning não vai substituir o ensino tradicional, mais fornece uma opção de carácter complementar para facilitar e aumentar o interesse dos estudantes da área, além de ter objetivo de ensina os estudantes teste de software usando como base a certificação Foundation do ISTQB (International software Testing Qualification Board).

A qualificação de nível *Foundation* destina-se a indivíduos envolvidos em testes de softwares, independente da função, tais como: analistas, consultores, engenheiros, consultores e ambos da área de teste de software, A qualificação de nível *Foundation* também é sugerida para obter conhecimentos básicos em teste software [ISTQB 2011].

Na seção 2 é descrito sobre teste de softwares e gTest Learning. Na seção 3 trabalhos relacionados e na seção 4 a conclusão e trabalhos futuros.

2. Teste de Software

Teste de software é considerado por IEEE [IEEE 2004], “A verificação dinâmica do funcionamento de um programa utilizando um conjunto finito de casos de teste, adequadamente escolhido dentro de um domínio de execução, em geral, infinito, contra seu comportamento esperado”. Neste sentido, é importante destacar que o teste exaustivo é inviável na pratica e que o mesmo compara o comportamento real do sistema contra o comportamento esperado.

Neste contexto, torna-se necessário descrever as diferenças entre casos de testes e procedimentos de testes. A especificação de casos de teste documenta as entradas, saídas esperadas e as condições de execução [IEEE 1998]. Vale a pena destacar que os procedimentos de teste são sequências de instruções para configuração, execução e avaliação de resultados de um caso de teste [IEEE 1990].

Testes podem ser classificados quanto ao objetivo. Testes funcionais verificam se o sistema está se comportando de acordo com as especificações. Teste de usabilidade analisa o quão fácil é para o usuário usar o software. Teste de desempenho verificam os requisitos de desempenho do software. Outros tipos de teste e mais detalhes podem ser encontrados em [IEEE 2004].

Testes também podem ser classificados quanto nível. Segundo Muller et al.[Müller, Graham, Friedenber e Veendental 2007] existem quatro níveis de teste:

- 1) Unidade, no qual procura-se defeitos em partes do software que podem ser testadas separadamente;
- 2) Integração, onde estão os testes caracterizados por testar as interfaces entre os componentes;
- 3) Sistema, no qual os testes verificam o comportamento de todo o sistema;
- 4) Aceitação, que abrange os testes que são frequentemente feitos pelo usuário final e que tem por objetivo estabelecer a confiança no sistema.

É importante ressaltar que a execução de testes é bastante incomoda por que requer tempo, conhecimento, planejamento infraestrutura e pessoal especializado [Myers 2004]. Sendo assim algumas ferramentas são de grande importância para reduzir os custos envolvidos:

- JMeter [JMeter 2014];
- Mantis [Mantis 2015];
- TestLink [TestLink 2015];
- Selenium [Selenium 2015];

Em geral, os processos de testes apresentam as seguintes atividades:

- Planejamento e Controle.;
- Análise e Modelagem (Projeto);
- Implementação e execução;
- Avaliação do critério de saída e relatórios;
- Atividades de encerramento de testes.

2.1. O Jogo gTest Learning

O gTest Learning é um jogo sério educacional para ensino de teste de software com o principal objetivo nas informações importantes sobre a teoria e prática de teste de software com base certificação *Foundation* do ISTQB (*International Software Testing Qualifications Board*).

Foi incluído no jogo diversas características para que se torne atraente para os utilizadores, focando sempre no teste de software com base na certificação *Foundation* do ISTQB, com este jogo sério educacional pretende-se:

- Novos desafios com questões sempre atuais de teste de software;
- A possibilidade de testar seus conhecimentos em qualquer lugar e no tempo que lhe for conveniente;
- Gerar ranking entre os jogadores, incentivando os mesmos a sempre buscar maior pontuação que resulta e mais conhecimento;
- Através de diálogos fixar item importantes para maturar o conhecimento do jogador.

Porém existem alguns pré-requisitos para que o jogador se sinta confortável dentro do jogo, é necessário ter conhecimentos no mínimo básicos nos seguintes tópicos:

1. Engenharia de Software: é preciso conhecer as fases do desenvolvimento do software e a importância de teste de software em cada fase na concepção de um software;
2. Teste de Software, ter uma base de conhecimento relativa aos conceitos de software.

O conteúdo de questões está baseado na estrutura de capítulos do programa de certificação *Syllabus* na versão de 2011 [ISTQB. 2011]. Segue abaixo os capítulos:

- Fundamentos de testes;
- Teste durante do Ciclo de Vida de Software;
- Técnicas Estáticas;
- Técnicas de Modelagem de testes;
- Gerenciamento de testes;
- Ferramentas de Suporte aos testes.

O software está disponível para acesso através de browser, para ser acessado das mais variadas plataformas, possibilitando o acesso por múltiplos jogadores.

2.2. Tecnologia do Jogo

O jogo foi desenvolvido usando as seguintes tecnologias IDE Construct 2, PHP, bando de dados MySQL, AJAX, API do Facebook, que serão descritas abaixo:

- Construct 2 [Scirra 2015] é um software de desenvolvimento de jogos 2D em *HTML5*, conhecido por não necessitar de conhecimento em programação, uma vez que usa uma técnica de desenvolvimento através de *drag-and-drop*, a lógica é baseada em comportamento. A ferramenta apresenta um editor de *JavaScript* permitindo assim a criação de comportamentos mais complexos para programadores experientes. O sistema fornece comportamentos flexíveis, pré-visualização instantânea, efeitos visuais aliciantes, publicação na maioria das plataformas e fácil extensibilidade através da agregação de diversos *plugins*.
- O PHP [Lynn e Michael 2009] (um acrônimo recursivo para PHP: Hypertext Preprocessor) é uma linguagem de script open source de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML, considerada por muito pela sua produtividade na hora de desenvolver sistema web.
- O MySQL [Lynn e Michael 2009] é SGBD muito usado no mundo e consolidado no mercado pela sua robustez e facilidade tanto para uso como gerenciamento.
- AJAX [Brett e Michael 2006] (acrônimo em língua inglesa de Asynchronous Javascript And XML) é o uso metodológico de tecnologias como JavaScript e XML, providas por navegadores, para tornar páginas Web mais interativas com o usuário, utilizando-se de solicitações assíncronas de informações.

- API Facebook possibilita a captura de informações de usuários do Facebook através de tecnologia PHP.

2.3. Modelagem do Jogo

Na figura 1 é apresentado o fluxo de atividades do jogo, que inicia na apresentação de um pequeno sumario do assunto que irá ser visto durante o estágio e em seguida ao e o, após este diálogo é feito perguntas relacionadas ao assunto discutido anteriormente.

As perguntas e diálogos são feitas com interface de RPG, para maior fixação do conteúdo, cada oponente tem 3 perguntas a serem feitas a exceto que é só uma, no final do turno de pergunta se quantidade de acertos for menor que erros o jogador perde o jogo, se não ele continua somando pontos até que chegue no final do estágio. Todas as questões foram obtidas de simulados do *ISTQB*, no final o jogador perdendo ou finalizando o estágio é exibido um ranking.

A infraestrutura e de pequeno porte visto que não há necessidade de grande estrutura para sustentar um sistema básico de consultas e *insert's*, como mostra na figura 2 o DER e a figura 3 a arquitetura de infraestrutura.

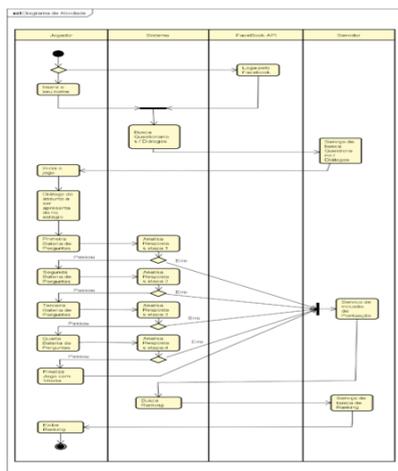


Figura 1 – Diagrama

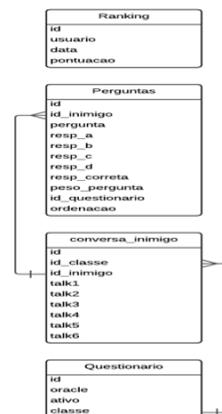


Figura 2 – DER

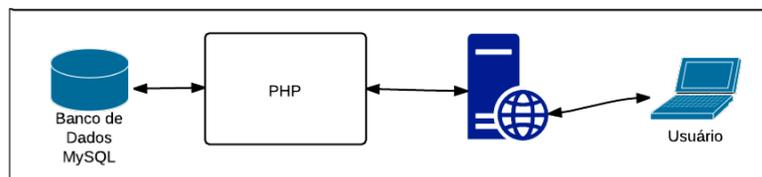


Figura 3 – Arquitetura

2.4. Dinâmica do Jogo

O gTest Learning é um jogo *single-player* (jogo para somente um jogador), em estilo *Plataforma* onde o jogador é desafiado a derrotar inimigos através de pequenos

questionários em fase de turno, adicionando uma técnica de diálogo usada em jogos de *RPG*.

O nível de dificuldade é balanceado através dos inimigos, sendo quatro inimigos por estágio, três perguntas para o primeiro, segundo e terceiro, no caso do quarto inimigo ele é considerado o BOSS do estágio, sendo assim só existe uma pergunta para enfrentá-lo.

Como mostra na Figura 4 o Oracle passa um resumo do assunto a ser trabalhado no estágio. Na Figura 5 o diálogo sempre acontece antes das perguntas serem feitas pelos inimigos, logo após finalizar o diálogo vamos para as perguntas que o inimigo irá fazer ao jogador representado na Figura 6. Existem itens a serem coletados que dar dicas para o jogador sobre as perguntas do estágio, no final do jogo é exibido o ranking. Figura 7, onde o jogador verificar sua colocação.



Figura 4 – Resumo



Figura 5 – Diálogo

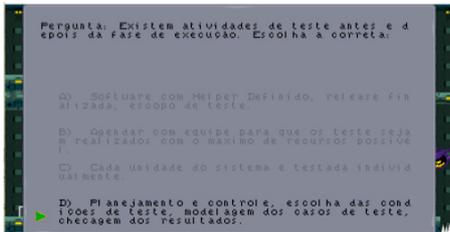


Figura 6 – Perguntas

1	rob.test1322	11/2015	623
2	rob.test1338	11/2015	623
3	rbo.test1340	11/2015	574
4	rebato	11/2015	547
5	hhhhh	11/2015	513
6	rob.test	11/2015	300

Figura 6 – Ranking

No mais o jogo é repleto de desafios, com vários inimigos e armadilhas para todo lado, no intuito que jogador possa se sentir dentro um jogo completo, a todo momento o jogador será desafiado, tanto para testes de software com base na certificação Foundation do ISTQB quanto para sair ileso das armadilhas e inimigos comuns (Que não possuem interação de conhecimento). O jogador irá pode usar pequenas bombas contra os inimigos comuns espalhados pelo estagio, quando mais acertar questões mais bombinhas ele irá ganhar.

Seguindo a curva de aprendizagem do jogo, todo cuidado foi tomado para que o jogador comece sem se incomodar com a dificuldade do mesmo [Jesus 2007]. A proposta da dinâmica do jogo é prender o jogador em um ambiente divertido e desafiador.

3. Trabalhos Relacionados

Vários jogos foram desenvolvidos com o proposito para apoiar o ensino de Engenharia de Software [Monsalve, Werneck e Leite 2010], [Gonçalves, Thiry e Zoucas 2011].

Para a criação de um novo jogo educacional é de suma importância conhecer e estudar os jogos que já desenvolvidos e disponíveis no mercado ou na academia, de modo a melhorar as metodologias a serem aplicadas visando alcançar um resultado mais satisfatório para o público alvo [Jesus 2007]. Contudo, aos que focam no ensino da disciplina de teste de software, foram identificados na literatura poucos jogos educacionais [Silva, Müller e Bernardi 2011a], [Diniz e Dazzi 2011], [Elbaum, Person e Dokulil 2007], [Silva 2010] os quais alguns deles estão descritos a seguir logo após os jogos uma tabela visualizando alguns itens comuns entre eles.

3.1. SimSE

O SimSE, Segundo [Navarro e Hoek 2005], é um jogo educacional focado em um ambiente de simulação, dedicando-se no conhecimento de engenharia de software chegando próximos da praticada nas empresas de desenvolvimento de projetos.

O jogo se passa em um ambiente *single-player* (jogo para somente um jogador), que o mesmo assume o papel do gesto de projeto de uma equipe de programadores. Todo ambiente virtual é visualizada de forma gráfica, interativa e divertida em um cenário que recebe uma avaliação direta do seu desempenho. Recebendo pontuações positivas se o projeto no prazo e dentro do custo.

O ponto muito forte do jogo é a qualidade gráfica, que mostra um escritório virtual com qualidade muito próximo da realidade, com computadores, mesas, cadeiras e até sala de reuniões, na mesma tela onde é visível o escritório virtual. Existem informações sobre produtividades de cada funcionário, estado de desenvolvimento das tarefas, nível de satisfação dos clientes, e status do projeto como todo, contendo orçamento e prazo do projeto. O SimSE é capaz de suportar diferentes modelos de processos aplicados em várias empresas. Logo abaixo está a figura 8 capturada da interface gráfica do jogo:

3.2. U-Test

O U-Test, é um jogo de simulação com foco em teste de unidade, abordando teoria e prática [Silva 2010]. Neste jogo é esperado que os jogadores tenham no mínimo conhecimento básico nos conceitos de programação, de engenharia de software e testes de software, este jogo tem como alvo que o jogador escreva testes unitários e focando nas técnicas de caixa-preta. Durante o jogo o jogador é apresentado a algumas etapas descritas abaixo:

- Definir valores-limite para as classes identificadas;
- Identificar as classes de equivalência;
- Selecionar o valor correspondente ao valor identificado;
- Configurar um gráfico de causa-efeito ou de árvore de decisão;

No final de cada etapa o jogador recebe a avaliação do seu desempenho em forma de gráficos com comentários, e no final e mostrado uma tabela de pontuação dos jogadores.

3.3. SE-RPG

O jogo SE-RPG [Benitti e Molléri 2008] simula um ambiente de desenvolvimento de software utilizado um cenário com diversos personagens, para o jogador interagir. É apresentado uma breve descrição do projeto selecionado pelo jogador, contendo a informações necessárias para o seguimento do desenvolvimento, como prazos e orçamentos, fica também a cargo do jogador escolher a metodologia de desenvolvimento que vai utilizar dentre a duas disponíveis (modelo cascata ou modelo iterativo), assim como a linguagem de programação a ser utilizada.

No próximo passo o jogador deve escolher a equipe de programadores para o desenvolvimento do projeto com base nos atributos de cada personagem, depois desse passo o jogado pode começar a distribuir atividades aos elementos de sua equipe recém-formada, isso tudo confrontando uma lista de requisitos disponíveis. Depois desta últimas ações o jogador já pode controlar o desenvolvimento do projeto verificando tempo gasto e o orçamento disponível.

A qualquer momento do projeto o jogado pode adicionar novos recursos, é claro que analisando seus atributos ou até mesmo demitir recursos, é possível ver o desempenho de cada desenvolvedor para facilitar no gerenciamento. No final com o projeto concluído, o software é entregue ao cliente e o jogador recebe seu resultado final, levando em consideração prazo, custo e metas alcançadas. A figura 10 foi capturada do jogo em execução.

3.4. Bug Hunt

The O Bug Hunt [Elbaum, Person e Dokulil 2007], foi desenvolvido para conquistar interesse dos alunos na aprendizagem de técnicas de teste de software, imagem capturada na figura 11. É uma aplicação *web* onde o aluno inicialmente tem acesso aos objetivos e a uma série de guias para ajuda [Elbaum, Person e Dokulil 2007]. Então, o aluno passa por um conjunto de lições sobre testes caixa-preta, caixa-branca, automação e eficiência de testes. Esta aplicação possui também um conjunto de exercícios e as respectivas soluções. Ao final, um sumário é exibido. O jogo tem diversas características que atraem tanto alunos quanto professores:

- Incorpora desafio por aula com avaliação imediata do desempenho na pratica dos conhecimentos fundamentais de testes de software;
- O aluno escolhe em qual ritmo vai resolver os desafios;
- A solução é confiável seguindo os requisitos de cada professor;
- Relatório completo para os professores para cada aluno.

Cada exercício é composto por um conjunto de componentes:

- Instruções – descrevem o desafio da aula e as atividades de teste;
- Artefatos – fornecem a informação necessária para a resolução da aula;
- Resultados imediatos – adaptados de acordo com o tipo de teste que está a ser realizado;
- Testes – submetidos durante a aula, um de cada vez, na área de casos de teste;
- Ajuda – dicas para ajudar o aluno na resolução do exercício.

No final de cada exercício o estudante recebe os resultados com o número de falhas encontradas, assim ajudando o estudante a entender melhor as questões que não conseguiu resolver antes e assim saber uma direção em que item deve focar nos estudos. A figura abaixo mostra a estrutura do Bug Hunt pela perspectiva do aluno.

3.5. Análise

A tabela 1 logo abaixo apresenta um comparativo entre os trabalhos encontrados na literatura x gTest Learning.

Tabela 1 – Comparação entre jogos educacionais na literatura

	Simulador	Multi-Plataforma	Intuitivo	Bom retorno de aprendizagem
SimSE	SIM	SIM	SIM	SIM
U-Test	SIM	SIM	NÃO	SIM
SE-RPG	NÃO	SIM	SIM	NÃO
Byg Hunt	NÃO	NÃO	NÃO	NÃO
gTest Learning	NÃO	SIM	SIM	SIM

Ambos os jogos encontrados na literatura são de um adicional imensurável, visto que quando mais ferramentas para o ensino de tecnologias e quanto mais agradar ao público alvo mais resultados será obtido com as técnicas de jogos sérios.

4. Conclusões e Trabalhos Futuros

Neste artigo, foi apresentado um game web para ensino na área de teste de software com base na certificação *Foudation* do ISTQB, buscando o método de treinamento com aprendizagem visto que sempre o game está oferecendo novas dicas sobre os assuntos testados.

O gTest Learning foca no teste de software, dentro do jogo, espera-se que os jogadores possam através de dicas e pequenas informações sedimentar seus conhecimentos sobre a teste de software, imerso dentro de um processo contínuo de reciclagem de questões sempre balanceando a aprendizagem, sempre sendo renovada a cada estágio do jogo.

Como trabalhos futuros, pretende-se criar vários estágios aumentando a dificuldade do game, e conseqüentemente aumentar a base de dados em relação a questões. Está previsto uma avaliação do jogo feita por alunos de Curso de Engenharia de Software e Game Design, para validar a capacidade de aprendizagem e teste de conhecimentos.

Referências

Monsalve, E. S., Werneck, V. M. B., Leite, J. C. S. P. (2010). “SimulES-W: Um Jogo para o Ensino de Engenharia de Software”. In: III Fórum em Educação de Engenharia de Software (FEES)., Salvador.

Gonçalves, R. Q., Thiry, M., Zoucas, A. (2011). “Avaliação da Aprendizagem em Experimentos com Jogo Educativo de Engenharia de Requisitos”. In: X Simpósio Brasileiro de Qualidade de Software (SBQS). Curitiba.

Silva, T. G.; Müller, F. M.; Bernardi, G. (2011a). “Panorama do Ensino de Engenharia de Software em Cursos de Graduação Focado em Teste de Software: Uma Proposta de Aprendizagem Baseada em Jogos”. In RENOTE - Revista Novas Tecnologias na Educação., ISSN 1679-1916.

Diniz, L. L., Dazzi, R. L. S. (2011). “Jogo Digital para o Apoio ao Ensino do Teste de Caixa-Preta”. In: X Simpósio Brasileiro de Qualidade de Software (SBQS)., Curitiba.

Elbaum, S., Person, S. e Dokulil, J. (2007). “Bug hunt: making early software testing lessons engaging and affordable”. In: 29th International Conference on Software Engineering (ICSE'07)., Minneapolis, p. 688 – 697.

Silva, A. C. (2010). “Jogo Educacional para Apoiar o Ensino de Técnicas para Elaboração de Testes de Unidade”. Dissertação de Curso de Mestrado, Computação Aplicada, UNIVALI, São José.

Myers, G. (2004). “The Art of Software Testing”. John Wiley & Sons., 2ª edição.

Wangenheim, C. G.; Silva, D. A. (2009). “Qual conhecimento de engenharia de software é importante para um profissional de software?” In Anais do Fórum de Educação em Engenharia de Software., Fortaleza

Andrade, A. F., Madeira, C. A. G., & Melo, H. H. A. (2013). “Batalha de Vetores Virtual: uma proposta de jogo pedagógico para o ensino de biociências”. XVIII Conferência Internacional sobre Informática na Educação (TISE)., Porto Alegre, Brasil.

Reis, S. C., Panciera, R. J., Gomes, A. F., & Menezes, V. P. (2013). “Da pesquisa à ação: conectando pressupostos teóricos e pedagógicos no desenvolvimento de um jogo de Inglês interdisciplinar em 3D”. XVIII Conferência Internacional sobre Informática na Educação (TISE)., Porto Alegre, Brasil.

Furtado, A., Vallerius, D., & Barone, D. (2013). “O Jogo Digital como Motivador do Interesse pela Literatura Brasileira em Alunos do Ensino Médio”. XVIII Conferência Internacional sobre Informática na Educação (TISE). Porto Alegre., Brasil.

Silveira, J. L.; Thiry, M.; Zoucas, A. C. (2013). “SPI City: Jogo Educacional para Apoiar o Ensino de Melhoria de Processo de Software”. In: XII Simpósio Brasileiro de Qualidade de Software – SBQS., 2013, Salvador, BA. p. 5165.

Pietruchinski, M. H.; Neto, J. C.; Malucelli, A.; Reinehr, S. (2011). “Os jogos educativos no contexto do SBIE: uma revisão sistemática de Literatura”. Anais do Simpósio Brasileiro de Informática na Educação – SBIE.

Müller, T.; Graham, D.; Friedenber, D. e Veendendal, E. (2007). “Base de Conhecimento para Certificação em Teste - Foundation Level Syllabus”. ISTQB - Comissão Internacional para Qualificação de Teste de Software.

Dempsey, J.; Rasmussen, K; Lucassen, B. (1994). “Instructional Gaming: implication for instructional technology”. Annual Meeting of Association for Educational Communications and Technology, Nashville.

Tânia, P. B. R. (2014). iLearnTest: Jogo Educativo para Aprendizagem de Testes de Software

K. Kapp. (2012). “The gamification of learning and instruction: game-based methods and strategies for training and education”. United States of America: Pfeiffer.

J. Fernandes e S. Sousa. (2010). PlayScrum - A Card Game to Learn the Scrum Agile Method. Em: Second International Conference on Games and Virtual Worlds for Serious Applications.

B. Schneiderman. (2005). Designing for Fun: Can we design user interfaces to be more fun?.

Scirra. *Construct 2*. Web Page. 2015. URL: www.scirra.com.

ISTQB. (2011). Certified Tester, Foundation Level Syllabus. Generic.

Lynn B., Michael M. (2009). Head First - PHP & MySQL.

Brett. M., Michael M. (2006). Head Rush - AJAX.

Jesus, P. A. (2007) Artes do Videogames - Conceito e Técnicas.

E. Navarro e A. van der Hoek. (2005) Software Process Modeling for an Educational Software Engineering Simulation Game». Em: Software Process Improv Pract. F. Benitti e J. Molléri. (2008).

Utilização de um RPG no Ensino de Gerenciamento e Processo de Desenvolvimento de Software.

IEEE (2004). SWEBOK - Guide to the software engineering body of knowledge. IEEE Computer Society.

IEEE (1998). IEEE Standard for Software Test Documentation, IEEE Std 829-1998, IEEE Computer.

IEEE (1990). IEEE Standard Glossary of Software Engineering Terminology, IEEE Std.610.12-1990, IEEE Computer Society.

Selenium (2015). “Ferramenta que automatiza testes funcionais ou de aceitação”. <http://docs.seleniumhq.org/>. Dezembro.

JMeter (2014). “Ferramenta que automatiza a execução de testes de desempenho”. <https://jmeter.apache.org/>. Dezembro.

Mantis (2015). “Ferramenta para o registro de Bugs (defeitos)”. <http://www.mantisbt.org/>. Dezembro.

TestLink (2015). “Ferramenta para documentação de casos de testes”. <http://testlink.org/>. Dezembro.