

# Evolução e um Estudo Exploratório do Jogo InspectorX

Henrique Pötter<sup>1</sup>, Marcelo Schots<sup>1,2</sup>, Vera Maria B. Werneck<sup>1</sup>

<sup>1</sup>Instituto de Matemática e Estatística – Universidade do Estado do Rio de Janeiro  
Caixa Postal 15.064, Rio de Janeiro, RJ, 91.501-970

<sup>2</sup>Programa de Engenharia de Sistemas e Computação – COPPE/UFRJ  
Caixa Postal 68.511, Rio de Janeiro, RJ, 21945-970

henriquepotter.hp@gmail.com, schots@cos.ufrj.br, vera@ime.uerj.br

***Abstract.** This paper describes the evolution of the InspectorX game, and presents an exploratory study with Computer Science students, from the disciplines of Software Quality and Systems Modeling, in order to evaluate the game. Results indicate that the impact on didactics is related to the quality of the question and clarity of the taxonomy used in the classification of defects.*

***Resumo.** Este artigo descreve a evolução do jogo InspectorX e apresenta um estudo exploratório com alunos de Ciência da Computação, das disciplinas Qualidade de Software e Modelagem de Sistemas, visando avaliar o jogo. Os resultados mostram que o impacto na didática está relacionado à qualidade da questão e da clareza da taxonomia usada na classificação dos defeitos.*

## 1. Introdução

Experiências comprovam que o uso de inspeções contribui para a melhoria da qualidade do produto, bem como para a redução do custo relacionado a testes, embora não elimine a necessidade de realizá-los. No entanto, o aprendizado do processo não é trivial, e requer treinamentos e potencialização das habilidades de futuros inspetores. Foi então proposto o InspectorX, um jogo para o treinamento e aprendizado em inspeção de software, visando à assimilação dos conceitos de forma lúdica [Pötter e Schots 2011].

Este artigo apresenta uma evolução do jogo e um estudo exploratório, sendo organizado da seguinte forma: a Seção 2 descreve as características e a evolução do jogo InspectorX; a Seção 3 apresenta um estudo realizado com alunos do curso de Ciência da Computação da UERJ; por fim, a Seção 4 contém as considerações finais.

## 2. O Jogo InspectorX

O jogo InspectorX [Pötter e Schots 2011] tem como objetivo prover uma maneira prática, eficiente e lúdica de treinar a percepção dos defeitos mais comuns, por parte de inspetores e desenvolvedores, permitindo desta forma o aprendizado de inspeção e, paralelamente, evitando a recorrência dos defeitos no futuro, conduzindo assim a uma melhoria nas práticas de desenvolvimento de software. O objetivo do jogador é obter o maior número de pontos na identificação (níveis intermediário e avançado) e classificação dos defeitos em artefatos de software. As pontuações dos jogadores são apresentadas por nível de dificuldade, usando a competitividade como um incentivo no jogo. Além disso, considerando que é possível utilizar diferentes taxonomias para a classificação de defeitos, especialmente quando se trata de artefatos de diferentes tipos

(e.g., código e requisitos), fez-se necessário possibilitar no jogo a customização das taxonomias utilizadas para as classificações de defeitos [Pötter e Schots 2011].

O InspectorX se encaixa na categoria de *serious game*, pois é um jogo que possui um objetivo educacional, e não apenas de entretenimento [Abt 1987]. Jogos nesta categoria usam o potencial imersivo e de engajamento dos jogos em geral, com o intuito de desenvolver novas habilidades para fins específicos. Além disso, eles possibilitam que os usuários (jogadores) vivenciem situações que seriam difíceis ou impossíveis por questões de custo, tempo, logística e segurança [Corti 2006].

Devido à complexidade inerente ao processo de inspeção, o InspectorX foi evoluído por meio da criação de dois modos de jogo (*Defect Crawler* e *Full Inspection Process*) para flexibilizar a jogabilidade, permitindo partidas mais casuais ou mais imersivas conforme a necessidade de treinamento.

No modo de jogo *Defect Crawler*, o foco é a atividade de *detecção* dos defeitos dos artefatos de software, sem contemplar as demais fases do processo de inspeção. O papel do jogador é identificar os defeitos e classificá-los de acordo com a taxonomia pré-estabelecida. Como não há uma pré-seleção dos artefatos a serem inspecionados, as questões (artefato com defeito) cadastradas podem surgir aleatoriamente (pois são embaralhadas pela versão moderna do algoritmo de Fisher-Yates [Durstfeld 1964]).

No modo de jogo *Full Inspection Process*, buscou-se simular o processo de inspeção de software como um todo, para tornar o ambiente do jogo mais próximo do cenário real. Os jogadores podem assumir dois papéis: (i) moderador, atuando nas atividades de *planejamento* (escopo da inspeção) e *avaliação* (análise dos defeitos apontados e definição se foram identificados corretamente, nos casos em que divergem da resposta cadastrada no jogo); (ii) inspetor, atuando na atividade de *detecção* (análise e relato de defeitos no escopo definido pelo moderador). O inspetor é avaliado quanto à identificação e classificação dos defeitos; já o moderador é avaliado por meio de uma auditoria externa, realizada por um conhecedor do domínio e da taxonomia, quanto à correta análise dos defeitos apontados (em termos de falsos positivos).

### **3. Avaliação do Jogo**

Visando avaliar a proposta didática do InspectorX, foi feito um estudo com alunos de Ciência da Computação das disciplinas Qualidade de Software e Modelagem de Sistemas. Buscou-se avaliar se o uso de jogos pode apoiar o aprendizado em inspeção e se as características do jogo InspectorX tornam o jogo mais atrativo ao jogador.

Para este estudo, foi utilizado apenas o modo *DefectCrawler*, por ter sido o primeiro modo desenvolvido e testado, sendo, portanto, mais estável. Foram elaboradas 20 questões com trechos de código fonte em Java e Portugol, sendo 10 para o nível de dificuldade básico e 10 para o nível intermediário (o nível avançado não foi avaliado). Cada questão possui uma descrição do contexto associado, caso necessário ou aplicável.

O estudo contou com 12 participantes, que enviaram por e-mail o questionário de caracterização. A análise do perfil incluiu a experiência em desenvolvimento de software, inspeções de software e Java. Todos os participantes foram treinados no jogo e na taxonomia utilizada [Jones 2009]. As respostas foram gravadas em um mecanismo de *logging* do jogo, permitindo obter por participante o tempo gasto e a pontuação por

questão. Por fim, um questionário *follow-up* permitiu a análise qualitativa do jogo.

Devido a alguns participantes terem chegado ao local do estudo com atraso, a divisão em grupos precisou ser refeita com base em um balanceamento entre o pré-estabelecido conforme o perfil e a ordem de chegada. A Tabela 1 mostra a distribuição final dos grupos e as categorias. É possível verificar que a heterogeneidade foi ponderada.

**Tabela 1. Alocação dos grupos versus categorias**

Grupo	Formação Acadêmica	Média da Experiência em Engenharia de software (0-4)	Média da Experiência em Qualidade de software (0-4)	Média da Experiência em Inspeção de software (0-4)	Média da Experiência em Desenvolvimento em Java (0-4)
1	Graduação em andamento	2,5	3	2	2,25
2	Mestrado em andamento	2,8	2,6	1,2	2,6
	Graduação em andamento				
3	Graduação em andamento	3	1,666	1,6667	2,6667

O grupo 2 possui algumas particularidades: além de possuir menor média de experiência em inspeção, um participante é mestrando. Assim, este grupo jogou tanto o nível fácil quanto o intermediário, para se analisar os resultados de forma independente no decorrer do aprendizado. A Tabela 2 correlaciona grupos e níveis de dificuldade.

**Tabela 2. Divisão dos grupos versus níveis de dificuldades**

	Nível fácil	Nível intermediário
Grupo 1		X
Grupo 2	X	X
Grupo 3	X	

As métricas usadas para descrever o desempenho dos participantes são eficiência (relação entre pontuação por questão e tempo gasto) e eficácia (porcentagem de acertos em relação ao total de questões). Os grupos 2 e 3, que jogaram o nível fácil, tiveram desempenho inferior ao grupo 1, que jogou apenas o intermediário (vide Tabela 3). Uma possível explicação está na alocação dos grupos (o grupo 1 tem mais experiência em inspeção); além disso, o grupo 2 indicou que o defeito estar destacado no nível fácil acabou dificultando a análise, pois tirou a atenção do resto do código.

**Tabela 3. Desempenho dos grupos (em termos de eficiência e eficácia)**

	Grupo 1	Grupo 2 (Fácil)	Grupo 2 (Intermediário)	Grupo 3
Eficiência	0.405	0.081	0.674	0.214
Eficácia	48.75%	18.00%	39.00%	23.33%

Na análise qualitativa, o jogo teve boa recepção pelos jogadores, que afirmaram que a ferramenta tem potencial para ser utilizada no aprendizado em inspeção. Alguns pontos que podem ter afetado o desempenho foram evidenciados: (i) um dos jogadores apontou a necessidade de melhoria na explicação de algumas questões; segundo ele, alguns enunciados pareciam não estar claros; (ii) um jogador do grupo 1 e um do grupo 2 apontaram a necessidade de utilizar artefatos de código mais completos para maior imersão no papel de inspetor; (iii) quase todos os jogadores apontaram a necessidade de melhoria na explicação da taxonomia utilizada. Os tipos de defeito que geraram mais confusão foram o de comissão (código cuja implementação é diferente do que foi especificado) e o de computação (um valor é definido erroneamente para uma variável).

Os participantes que afirmaram possuir experiência na indústria e conhecimento

elevado nas áreas da Tabela 1 mostraram, em geral, melhor desempenho no jogo, em ambos os níveis. Outro ponto notado é o impacto da qualidade da questão na qualidade do aprendizado (uma questão mal elaborada pode distorcer a compreensão dos conceitos) [Pötter e Schots 2011]. Por fim, as seguintes ameaças à validade foram detectadas:

- Validade Interna: (i) não é possível garantir que as informações do questionário de caracterização condizem com a realidade; (ii) o desempenho dos participantes foi afetado pela descrição das questões; no entanto, todas as questões tiveram porcentagem razoável de acertos; (iii) a taxonomia foi ensinada no mesmo dia do estudo, assim não é possível saber se esta foi assimilada e se isto impactou o desempenho; (iv) apesar dos cuidados na codificação do jogo e testes em pequena escala, possíveis defeitos na implementação do jogo podem ter afetado os resultados.
- Validade Externa: (i) com exceção de um participante, todos os demais são alunos de graduação, o que limita o escopo de análise na didática do InspectorX no ensino de inspeção; porém, nota-se que muitos dos participantes trabalham na indústria; (ii) o estudo foi realizado usando somente questões com artefatos de código, o que limita o universo de aplicação do InspectorX; (iii) apenas o modo *Defect Crawler* foi avaliado, i.e., os resultados podem ser diferentes no modo *Full Inspection Process*.
- Validade de conclusão: o tamanho da amostra não é significativo para evidenciar padrões consistentes que permitam resultados conclusivos.
- Validade de construção: (i) como o jogo foi acessado via web e o cálculo do tempo de duração de cada questão foi feito no servidor, não há como prever se problemas na rede afetaram o tempo de resposta no computador do participante; (ii) não foi analisada a dificuldade inerente ao trecho de código (e.g., algumas questões podem ser mais difíceis em função da medida de sua complexidade); (iii) o atraso de alguns participantes pode impactar os resultados, apesar de ser o mesmo treinamento.

#### 4. Considerações Finais

Este artigo apresentou a evolução do jogo InspectorX e a realização de um estudo exploratório para observar a utilização do jogo em sala de aula. Foi possível notar a importância do cuidado na elaboração das questões e como a dificuldade decorrente da complexidade do código pode influenciar na identificação e classificação dos defeitos. Além disso, observou-se que o jogo pode ter impactos didáticos práticos positivos.

#### Agradecimentos

Agradecemos aos participantes do estudo pela disponibilidade e pelas contribuições, e à colaboração e incentivo da profa. Letícia Duboc (IME/UERJ) na realização do estudo.

#### Referências

- Abt, C. *Serious Games*. Washington, D.C.: University Press of America, 1987.
- Corti, K., 2006. *Games-based Learning; a serious business application*, White Paper, PIXELearning Limited, February.
- Durstenfeld, R., 1964. "Algorithm 235: Random permutation". *CACM*, 7 (7), p. 420.
- Jones, C., 2009. *Software Engineering Best Practices*, McGraw-Hill Inc., NY, USA.
- Pötter, H., Schots, M., 2011. InspectorX: Um Jogo para o Aprendizado em Inspeção de Software. *IV Fórum de Educação em Engenharia de Software (FEES'11)*, São Paulo.