

# Adaptando o Design Thinking para a Definição e Desenvolvimento de um Jogo Educacional Não Digital no Ensino de Gerenciamento de Riscos

Sebastião Santos<sup>1</sup>, Yandson Costa<sup>1</sup>, Flávia Carvalho<sup>2</sup>, Davi Viana<sup>3</sup>, Luis Rivero<sup>3</sup>

<sup>1</sup>Departamento de Informática (DEINF)  
Universidade Federal do Maranhão (UFMA) – São Luís, MA – Brasil

<sup>2</sup>Departamento de Desenho e Tecnologia (DEDET)  
Universidade Federal do Maranhão (UFMA) – São Luís, MA – Brasil

<sup>3</sup>Programa de Pós-Graduação em Ciência da Computação (PPGCC)  
Universidade Federal do Maranhão (UFMA) – São Luís, MA – Brasil

{sebastiaohns, yandson.jesus, flvdcrvlh}@gmail.com,  
davi.viana@lsdi.ufma.br, luisrivero@nca.ufma.br

**Abstract.** *An alternative to traditional teaching is the use of educational games that can motivate students. However, there are still difficulties in the development of educational games when selecting their content and when designing their dynamics. This paper presents the experience report of the adaptation of the design thinking methodology in the development process of the non-digital educational game: Risking (Arriscando). Design Thinking combines the focus on the end user with multidisciplinary collaboration and iterative improvement to produce innovative products. A course content analysis was applied to define the subject of the game, as well as brainstorming and validation meetings with the students.*

**Resumo.** *Uma alternativa para o ensino tradicional é o uso de jogos educacionais que podem motivar os alunos. No entanto, ainda há dificuldade no desenvolvimento de jogos educacionais ao selecionar os conteúdos presentes e ao projetar a dinâmica do jogo. Este trabalho apresenta o relato de experiência da adaptação da metodologia design thinking no processo de desenvolvimento do jogo educacional não digital: Arriscando. O Design Thinking combina o foco no usuário final com colaboração multidisciplinar e aprimoramento iterativo para produzir produtos inovadores. Foi aplicada uma análise de conteúdos de disciplinas para definir o assunto do jogo, assim como reuniões de brainstorming e validação com os discentes.*

## 1. Introdução

Na disciplina de Engenharia de Software (ES), o professor preocupa-se não apenas com a transmissão de conceitos e teorias, mas principalmente com a associação do campo teórico e prático para o desenvolvimento eficiente de projetos de software [Prikladnicki et al. 2009]. Sendo assim, ao mesmo tempo que é extremamente relevante o conteúdo a ser ensinado dentro da sala de aula, também é notada a necessidade de manutenção dos aspectos didáticos e pedagógicos no ensino [Savi et al. 2011].

Uma alternativa para o ensino da ES de forma inovadora é a partir do uso de objetos de aprendizagem (jogos, mídias, ferramentas, entre outros) como forma de auxiliar o ensino, satisfazendo a necessidade de aprendizagem de maneira efetiva [Prikladnicki et al. 2009]. Transformando aulas que, comumente, são somente expositivas, em dinâmicas que envolvam toda a turma, para permitir a vivência de situações reais que podem ser encontradas no mercado de trabalho.

Apesar da existência de metodologias para o desenvolvimento de jogos educacionais [de Oliveira et al., 2014], poucas metodologias detalham o processo de identificação dos tópicos a serem abordados nos jogos, além da discussão de como pode ser definida a dinâmica de jogo. Nesse contexto, este trabalho apresenta o processo de desenvolvimento do jogo Arriscando, um jogo de cartas que motiva o aluno a relacionar riscos de projetos de software com possíveis soluções através de uma dinâmica de ataque e defesa, utilizando a metodologia *Design Thinking* (DT) [Plattner et al. 2010]. Nas próximas seções é apresentado o objeto de aprendizagem desenvolvido e os passos realizados no seu processo de desenvolvimento.

## **2. Processo de Desenvolvimento**

Para o desenvolvimento do jogo educacional no apoio ao ensino de engenharia de software foi utilizado o processo de desenvolvimento *Design Thinking* (DT), uma metodologia centrada nas pessoas que integra conhecimentos de design, ciências sociais, engenharia e negócios. A metodologia combina o foco no usuário final com colaboração multidisciplinar e aprimoramento iterativo para produzir produtos, sistemas e serviços inovadores [Plattner et al. 2010]. O modelo de DT da d.School da Universidade de Stanford [Doorley et al. 2018] é caracterizado por cinco fases iterativas e conectadas: Empatizar, Definir, Idealizar, Prototipar e Testar. Para estas fases, são apresentadas as adaptações e atividades executadas no contexto de desenvolvimento de um jogo educacional para ensino de engenharia de software.

### **2.1. Empatizar**

O desenvolvimento do jogo foi iniciado na primeira etapa da metodologia do DT, Empatizar. Nessa etapa foram identificados os tópicos ensinados nas disciplinas introdutórias à ES, através da análise das ementas dos 10 cursos de computação que foram mais bem avaliados no país segundo o ranking da Revista Folha. O ranking inclui as seguintes universidades em ordem de classificação: UNICAMP, USP, UFRGS, UFMG, UFRJ, UFPE, UFSC, UFSCAR, PUC-RIO, UNB. Também foram levantados os tópicos ensinados na disciplina de Engenharia de Software I da Universidade Federal do Maranhão (UFMA), de modo a comparar a ementa da disciplina nos cursos de computação desta instituição e analisar que tópicos poderiam ser melhor explorados. Os principais tópicos identificados foram: Processo e Produto de Software, Estimativa de Software, Engenharia de Requisitos, Design (Projeto de Software), Implementação de Software, Qualidade de Software, Teste de Software, Gerência de Configuração, Gerenciamento de Riscos e Gerência de Projetos.

Para entender melhor o público alvo do jogo foi utilizado o estudo de Mendes et al. (2019) como fonte para a identificação das expectativas e dificuldades de alunos de graduação no ensino de ES. O trabalho visou entender as necessidades dos alunos iniciantes em ES através da aplicação de um questionário quanto à experiência na disciplina introdutória de ES. Os resultados mostraram que o ensino com abordagem

teórica não corresponde às expectativas dos alunos para a disciplina. Mostra-se uma insatisfação sobre a didática tradicional e a dificuldade de aplicar os conteúdos na prática ou de forma dinâmica [Mendes et al. 2019].

Com relação aos jogos educacionais existentes, foi realizado um mapeamento sistemático que caracterizou os jogos educacionais para o ensino da ES, permitindo identificar os pontos fortes (abordagem dos conteúdos de forma dinâmica e interativa) e fracos (a indisponibilidade dos jogos para replicação) dos jogos disponíveis na literatura para ensino de engenharia de software. Além disso, foi identificada a baixa quantidade de jogos desenvolvidos para o ensino de determinadas subáreas da ES, principalmente Gerenciamento de Projetos e nesse contexto, Gerenciamento de Riscos [Santos et al. 2020].

## **2.2. Definição e Ideação**

Os resultados obtidos na fase de Empatia, foram um guia para a definição dos aspectos que o jogo a ser desenvolvido deveria ter. Tendo identificado o público alvo, os alunos da disciplina introdutória de Engenharia de Software e a subárea que seria abordada, o Gerenciamento de Riscos (GR) - princípios de gerenciamento de projetos de desenvolvimento que relacionam riscos com orientações de sucesso, foi definido que o objetivo do jogo seria apresentar conceitos de GR e permitir que os alunos treinassem seus conhecimentos em sala de aula de forma rápida, fácil e lúdica. Como há jogos educacionais para o ensino de Gerenciamento de Projetos e tem objetivos pedagógicos avançados, optou-se por fornecer apoio aos objetivos de aprendizagem mais iniciais segundo a Taxonomia de Bloom: conhecimento e compreensão [Ferraz e Belhot 2010].

No processo de ideação foram realizadas diversas propostas de gêneros de jogos e abordagens a serem utilizadas no jogo para o ensino de GR em projetos de software. Foi proposto o desenvolvimento de um jogo no estilo de perguntas e respostas (quiz), um jogo de tabuleiro e a realização da gamificação da disciplina, porém essas propostas foram descartadas, após discussões sobre a dificuldade do aprendizado e utilização da dinâmica do jogo em uma única aula com duração de 100 a 120 minutos. Dessa forma a ideia que prevaleceu foi a de desenvolver um jogo de baralho, não digital, apresentando uma dinâmica de ataques e defesas, para que fosse possível apresentar o conteúdo aos alunos iniciantes na ES e permitir a aprendizagem e associação dos conceitos definidos.

## **2.3. Prototipação e Testes**

Para o processo de prototipação do jogo foi necessário identificar os riscos e as respectivas mitigações existentes. Dessa forma foram buscados trabalhos científicos na área de GR que apresentavam tais informações de forma objetiva e condensada. Assim, foram utilizadas abordagens formalizadas de GR apresentadas nos trabalhos de Boehm (1991), Shrivastava e Rathod (2015) e Lyytinen et al. (1998). Os trabalhos apresentam riscos sintetizados e classificados, além das respectivas estratégias de resolução usadas. Todos os trabalhos utilizam o modelo de classificação de Leavitt (1964), que classifica os elementos de uma organização em quatro componentes (tarefas, pessoas, estrutura, tecnologia). Esses componentes podem ser combinados entre si formando novos componentes compostos. Essa classificação evidencia que mudanças realizadas em um componente podem acarretar impactos nos outros.

Para a validação da dinâmica do jogo e do conteúdo utilizado diversas partidas de testes foram realizadas. Dessa forma foi possível corrigir a apresentação dos conceitos principais do jogo, riscos, mitigações e categorias. Além de aperfeiçoar a dinâmica, balanceando a competitividade e adicionando estratégias que estimulem a jogar permitindo que seja facilitado o entendimento e a jogabilidade.

### 3. Resultados

A Figura 1 apresenta o jogo Arriscando, desenvolvido para auxiliar o ensino de GR em projetos de desenvolvimento de software. É um jogo de baralho não digital baseado em uma dinâmica de ataque (o surgimento de riscos em projetos de desenvolvimento de software) e defesa (o uso de mitigações adequadas para estes riscos) para abordar os imprevistos que podem ocorrer em um projeto de desenvolvimento de software. O objetivo do jogo é apresentar aos participantes as categorias de riscos, riscos existentes dentro destas categorias e as mitigações que podem ser usadas para diminuir o impacto negativo dos riscos nos projetos. O jogo permite fazer o relacionamento existente entre categorias de riscos e riscos, assim como riscos e mitigações. Com isso, o jogo visa o entendimento do impacto que os riscos podem ocasionar nos projetos de software e a importância de realizar planos de gerenciamento de riscos. O jogo foi aplicado em diversas turmas nas disciplinas da ES em cursos de computação da UFMA, obtendo resultados positivos em termos de aceitação [Santos et al. 2019].



Figura 1. Elementos do jogo Arriscando

### 4. Conclusão

Esse trabalho apresentou os passos realizados no processo de desenvolvimento de um jogo educacional para o ensino de gerenciamento de riscos. Utilizando a metodologia colaborativa e iterativa *Design Thinking*, o jogo contou com o envolvimento de um profissional e docente na área da computação e estudantes de computação e design na realização das diversas atividades do processo de desenvolvimento. Como resultado obteve-se o jogo Arriscando, para ensinar os conceitos iniciais de gerenciamento de riscos em projetos de desenvolvimento de software de forma interativa e lúdica. O jogo visa alcançar os objetivos de aprendizagem de compreensão, análise e associação dos diferentes tipos de categorias de riscos, riscos e mitigações.

A partir dos resultados da avaliação do jogo, foi possível concluir que o jogo Arriscando foi bem aceito pelos alunos, em que todos os itens do modelo de avaliação utilizado tiveram mediana 1 ou 2 (em uma escala de 5 pontos), indicando respostas positivas. As dimensões de Diversão e Interação Social percebidas pelos participantes não obtiveram nenhuma resposta discordante com as afirmações. Como pontos negativos foi destacado a pouca variedade de cartas no jogo, gerando uma repetitividade

do conteúdo, a falta de material de apoio e um design pouco apelativo. Espera-se que a metodologia adaptada sirva de exemplo no desenvolvimento de futuras propostas de jogos educacionais que atendam às necessidades das distintas disciplinas dentro da computação, considerando jogos existentes e as ementas das disciplinas similares em cursos de computação no Brasil.

## Referências

- Boehm, B. W. (1991). Software risk management: principles and practices. *IEEE software*, 8(1):32–41.
- Doorley, S., Holcomb, S., Klebahn, P., Segovia, K., and Utley, J. (2018). Design thinking bootleg. Retrieved September, 12:2018.
- Ferraz, A. P. D. C. M., & Belhot, R. V. (2010). Taxonomia de Bloom: revisão teórica e apresentação das adequações do instrumento para definição de objetivos instrucionais. *Gestão & Produção*, 17(2), 421-431.
- Lyytinen, K., Mathiassen, L., and Ropponen, J. (1998). Attention shaping and software risk—a categorical analysis of four classical risk management approaches. *Information Systems Research*, 9(3):233–255.
- Mendes, J., Costa, Y., Frazão, K., Santos, R., Santos, D., and Rivero, L. (2019). Identificação das expectativas e dificuldades de alunos de graduação no ensino de engenharia de software. In *Anais do XXVII Workshop sobre Educação em Computação*, pages 334–347, Porto Alegre, RS, Brasil. SBC.
- de Oliveira, H. C., da Silva Hounsell, M., & Kemczinski, A. (2014). Mapeamento sistemático de metodologias de desenvolvimento centrado no usuário para jogos sérios. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (Vol. 25, No. 1, p. 727).
- Plattner, H., Meinel, C., and Leifer, L. (2010). *Design thinking: understand–improve–apply*. Springer Science & Business Media.
- Prikladnicki, R., Albuquerque, A. B., von Wangenheim, C. G., and Cabral, R. (2009). Ensino de engenharia de software: desafios, estratégias de ensino e lições aprendidas. *FEES-Fórum de Educação em Engenharia de Software*, pages 1–8.
- Santos, S. H. N., Costa, Y. D. J. S., dos Santos, D. V., Barradas Filho, A. O., Junior, J. B. B., & Rivero, L. (2020). Identifying Serious Games for Software Engineering Teaching in Brazil Through a Systematic Mapping Study. *Research, Society and Development*, 9(7), 329973702.
- Santos, S., Carvalho, F., Costa, Y., Viana, D., & Rivero, L. (2019, October). Risking: A Game for Teaching Risk Management in Software Projects. In *Proceedings of the XVIII Brazilian Symposium on Software Quality* (pp. 188-197).
- Savi, R., Wangenheim, C., and Borgatto, A. (2011). Um modelo de avaliação de jogos educacionais na engenharia de software. *Anais do XXV Simpósio Brasileiro de Engenharia de Software (SBES 2011)*, São Paulo.
- Shrivastava, S. V. and Rathod, U. (2015). Categorization of risk factors for distributed agile projects. *Information and Software Technology*, 58:373–387.