

LEARN Board Game: A game for teaching Software Architecture created through Design Science Research

Tamires A. S. Sousa[†]
Universidade Federal do Ceará
Russas, Ceará, Brasil
eng.soft.tamires@gmail.com

Anna B. S. Marques
Universidade Federal do Ceará
Russas, Ceará, Brasil
beatriz.marques@ufc.br

ABSTRACT

Software Architecture specifies the set of decisions about the software components, their external properties and their relationships with other elements. These activities have a direct impact on the software quality, requiring architects and developers to know and to apply appropriately the architectural concepts and standards for their decision making involving the Software Architecture. Due to this, we identified a need for designing an approach that would dynamize the teaching and learning process in Software Architecture. Educational games are an alternative for teaching in several Computing areas. Therefore, this research proposes the creation and application of LEARN (LEarning software ARchitecture fundameNtals) board game, a game for teaching architectural concepts and standards in an interactive way. We created the game using the Design Science Research methodology. We evaluated the game through a case study with undergraduate students in Software Engineering, attending the Software Architecture discipline. We used a questionnaire based on the MEEGA+ model for data collection. The results show that LEARN board game has a good usability and provided a good learning experience for the students. The LEARN new version is available for adoption in Software Architecture learning.

CCS CONCEPTS

• Software and its engineering → Software organization and properties → Software system structures → Software architectures • Applied computing → Education → Collaborative learning

KEYWORDS

Jogos educacionais, Arquitetura de software, Design Science Research

ACM Reference format:

Tamires A. S. Sousa and Anna B. S. Marques. 2020. LEARN Board Game: A game for teaching Software Architecture created through Design Science Research. In *XXXIV Brazilian Symposium on Software Engineering (SBES 2020)*, October 21-23, 2020, Natal, Brazil. ACM, NY, USA, 10 pages. <https://doi.org/10.1145/1234567890>

1 INTRODUÇÃO

A Arquitetura de Software é uma etapa crucial do projeto de uma aplicação, pois possibilita a visão do funcionamento do sistema e visões sobre aspectos de qualidade como performance, confiabilidade, portabilidade, escalabilidade, usabilidade e interoperabilidade. Por isso, fornece uma garantia de que o sistema atingirá o objetivo que ele pretende alcançar [1], constrói a base para a sua implementação [2]. Deste modo, a Arquitetura de Software é um desafio tanto para os estudantes entenderem, como para educadores ensinarem por envolver a tomada de decisões com base em diferentes visões [3]. Estudantes geralmente possuem base em programação, no entanto, raramente conhecem os conceitos arquiteturais e as suas influências na qualidade, produtividade, custo e manutenção de um software [4].

Neste sentido, estratégias aplicadas ao ensino de Computação, como jogos educacionais, vêm se tornando cada vez mais comuns [5]. Estes jogos são projetados para ensinar as pessoas acerca de um determinado assunto, expandir conceitos, reforçar o desenvolvimento, auxiliá-las exercitando uma habilidade ou buscando uma mudança de atitude enquanto jogam [6].

Os jogos podem ser classificados como não-digitais ou digitais. Os jogos digitais são caracterizados por serem representados em forma de elementos gráficos interativos jogados por meio de um dispositivo virtual, como um celular ou computador. Por sua vez, os jogos não-digitais são caracterizados por serem representados ou envolverem objetos físicos e palpáveis, podendo ser jogados com cartas, tabuleiros, objetos esportivos, dentre outros [7]. Jogos educacionais não-digitais promovem um ambiente de interação social e imersão dos estudantes sobre a tarefa de aprendizagem [31].

Existem abordagens que utilizam elementos não-digitais para o suporte e ensino em Engenharia de Software [8, 9, 10, 11, 12, 13]. Entretanto, não foram encontrados trabalhos que utilizam cartas

*Article Title Footnote needs to be captured as Title Note

†Author Footnote to be captured as Author Note

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

WOODSTOCK'18, June, 2018, El Paso, Texas USA

© 2018 Copyright held by the owner/author(s). 978-1-4503-0000-0/18/06...\$15.00

e tabuleiro para o ensino de conceitos e padrões relacionados à Arquitetura de Software.

Diante disto, este artigo apresenta o LEARN Board Game – *LEarning software ARchitecture fundameNtals*, um jogo não-digital, composto por tabuleiro e cartas, cujo objetivo é auxiliar na aprendizagem de conceitos e padrões voltados para a Arquitetura de Software. O jogo foi desenvolvido seguindo a metodologia *Design Science Research* e avaliado por meio de um estudo de caso com estudantes de Engenharia de Software, na disciplina de Arquitetura de Software. A coleta de dados foi realizada por meio da adoção do modelo MEEGA+.

Este artigo é estruturado da seguinte forma: na Seção 2 são apresentados os conceitos que fundamentam esta pesquisa; a Seção 3 discorre sobre os trabalhos relacionados; a Seção 4 apresenta a metodologia adotada; na Seção 5 é apresentada versão inicial do LEARN Board Game, os seus elementos e a dinâmica; na Seção 6 é apresentada a avaliação do jogo por meio de um estudo de caso; os resultados da avaliação são evidenciados na Seção 7; a Seção 8 apresenta a versão atual do jogo; a Seção 9 discorre sobre as ameaças à validade; a Seção 10 contém as considerações finais e os trabalhos futuros.

2 BACKGROUND

2.1 Arquitetura de Software

A Arquitetura de Software é definida por Biel e Gruhn [22] como o conjunto de decisões significativas para a organização de um software, desde a seleção dos elementos estruturais que irão compor o sistema, até o comportamento e estilo arquitetural que orientará esta organização. Deste modo, a arquitetura é um dos artefatos mais importantes para um sistema e as decisões arquiteturais impactam diretamente nos objetivos de negócio, nos requisitos funcionais e de qualidade do software [23].

Em relação às decisões arquiteturais, Bass et al. [21] apresentam sete categorias de decisões: escolha da tecnologia, modelo de dados, mapeamento entre elementos, *binding time*, gerenciamento de recursos, alocação de responsabilidades e modelo de coordenação. Grande parte das decisões arquiteturais resultam em várias consequências [24] e, algumas das consequências mais significativas, segundo Garlan [25] são que elas afetam os atributos de qualidade do sistema.

Os atributos de qualidade representam as características não funcionais de um sistema [24]. Estes atributos podem ser definidos com base na ISO/IEC 25010 [29], um modelo que define características de qualidade de software. Bass et al. [21] especifica sete atributos de qualidade que impactam na Arquitetura de Software: disponibilidade, interoperabilidade, modificabilidade, desempenho, segurança, testabilidade e usabilidade. A solução arquitetural de um software deve permitir que os atributos de qualidade prioritários sejam favorecidos.

As variadas arquiteturas de software existentes possibilitaram a identificação de estilos e padrões arquiteturais. Os estilos e padrões arquiteturais configuram uma base de conhecimento onde se representa, comunica e constrói novas arquiteturas e se utilizam arquiteturas existentes em novos projetos [27]. Os

padrões são soluções para problemas recorrentes [24]. Bass et al. [21] permitem a caracterização de alguns padrões de acordo com o uso dos elementos arquiteturais, entre os quais são citados: padrão em camadas, padrão repositório e padrão cliente-servidor. Os padrões e estilos arquiteturais podem emergir conforme novas soluções arquiteturais são criadas, desenvolvidas e testadas [21].

2.2 Jogos educacionais não-digitais

Segundo Savi et al. [14] é necessário melhorar o ensino-aprendizagem em Engenharia de Software, visto que pela forma tradicional de ensino não há a aplicação prática dos conceitos. Assim, estratégias aplicadas ao ensino de computação, como jogos educacionais, estão cada vez mais comuns [18]. Estes jogos têm sido intensamente utilizados por profissionais da área de educação como forma de apoiar a construção do conhecimento [19].

Apesar de grande parte dos jogos educacionais existentes em Computação serem digitais, atualmente, os jogos não-digitais também são adotados para o ensino [28]. Segundo avaliações conduzidas por Petri et al. [20] a utilização de jogos não-digitais para o ensino em computação apresentou resultados positivos como a facilidade de interação social, a promoção de um ambiente agradável e divertido para a aprendizagem e o incentivo ao compartilhamento de ideias entre os estudantes.

Um jogo não-digital caracteriza por envolver o uso de tabuleiros, cartas, lápis e papel [32]. Petri et al. [20] concluiu que estes jogos não-digitais proporcionam uma boa experiência aos jogadores e são muito interativos.

3 TRABALHOS RELACIONADOS

Jogos educacionais podem ser uma estratégia instrucional eficaz e eficiente para o ensino de Computação [17]. Estes jogos visam dinamizar o processo de aprendizagem acerca de determinado assunto. Com isso, diversas áreas, como a Engenharia de Software, passaram a utilizar jogos não-digitais para o ensino teórico e prático dos conteúdos. Exemplos desses jogos incluem o 2TScrum [8], Masters of the Process [9], ProcSoft [10], Scrum Card Game [11] e Arriscando [13]. Especificamente para o ensino de Arquitetura de Software, temos os jogos Software Architecture Design Reasoning [12] e Smart Decisions [26].

O 2TScrum [8] é um jogo de tabuleiro com o objetivo de facilitar o ensino do da metodologia Scrum através da simulação do desenvolvimento de um projeto. O jogo foi avaliado por meio do modelo de avaliação de jogos educacionais descrito por Savi et al. [14] onde são consideradas as dimensões de experiência do usuário, motivação e aprendizagem. Os resultados da avaliação do jogo foram satisfatórios e mostraram que o objetivo de facilitar o ensino de Scrum por meio de um jogo foi atingido. Outro jogo que aborda o Scrum é o Scrum Card Game [11], um jogo de cartas cujo o objetivo é apresentar possíveis eventos e problemas que podem ocorrer durante um projeto de software e mostrar possíveis soluções. O trabalho proposto por Moreira e Marques [7] apresentou um estudo exploratório com a avaliação do jogo através da utilização do questionário MEEGA+ [16], uma evolução do modelo de Savi et al. [14]. Os resultados apontaram para uma

percepção positiva dos jogadores em relação ao jogo, visto que o jogo influenciou tanto no aprendizado de conhecimentos técnicos, como também no desenvolvimento de habilidades.

O Masters of the Process [9] é um jogo de tabuleiro para o ensino de competências do processo de gerenciamento e desenvolvimento de software. Durante o jogo, os jogadores desenvolvem o projeto de software de maneira divertida baseando-se nas fases do processo RUP (Rational Unified Process). O jogo foi avaliado mediante a aplicação de um questionário de usabilidade definido por [15]. Diante dos resultados foi possível observar que o jogo conseguiu unir diversão e aprendizado em Engenharia de Software. Ainda sobre o ensino de processo de software, o ProcSoft [10] é um jogo de tabuleiro que objetiva ensinar de forma descontraída os conceitos relacionados à definição, estrutura e conteúdo de um processo de software com base na norma ISO/IEC 29110. Com ele os estudantes conhecem as atividades comuns em um processo de software e as relações dessas atividades com papéis e ferramentas. Para a avaliação foi utilizado um questionário de acordo com o modelo de Savi et al. [14]. Os resultados obtidos com a avaliação destacam a capacidade de aprendizagem e a aceitação do jogo pelos participantes. Com foco no ensino de gerência de riscos, o jogo de cartas Arriscando [13] objetiva fornecer o conhecimento aos estudantes sobre o impacto dos riscos no gerenciamento de um projeto de software e as mitigações que podem ser feitas com os riscos negativos. O jogo foi avaliado por meio de uma dinâmica (*Lovers vs Haters*) e pela aplicação do questionário MEEGA+[16]. Os resultados mostraram que o jogo foi bem aceito pelos estudantes de acordo com as respostas às dimensões do modelo de avaliação.

O Software Architecture Design Reasoning [12] consiste em um jogo de cartas para auxiliar o raciocínio de arquitetos novatos na tomada de decisões arquiteturais. As cartas são divididas em atividades de raciocínio (contexto, problema e solução) e em técnicas de raciocínio (restrição, suposição, risco e *trade-off*). O jogo foi avaliado com base em observação em um estudo experimental. Os participantes foram divididos em equipes para a resolução de um problema com as cartas e sem as cartas. Como resultados, notou-se que as equipes que usaram as cartas discutiram mais o problema e apresentaram melhores resultados em relação às equipes que não usaram as cartas.

Smart Decisions [26] consiste em um jogo de cartas desenvolvido com o objetivo de ensinar e estimular discussões sobre padrões arquiteturais de aplicações Big Data. O jogo contém cartas que abordam sua mecânica e cartas relacionadas à Big Data. Foi aplicado com educadores, estudantes e profissionais com o objetivo de avaliar a proposta e captar melhorias. Dados foram coletados a partir da observação das jogadas dos jogadores e por meio de um questionário de feedback. Os resultados da aplicação das cartas foram positivos e mostraram que o jogo pode ser uma ferramenta útil para auxiliar no ensino de arquitetura de software.

Em todos os estudos foram gerados e avaliados artefatos que utilizam elementos não-digitais como tabuleiro e cartas para auxiliar no ensino de determinado assunto. Os jogos 2TScrum [8], Masters of the Process [9], ProcSoft [10], Scrum Card Game [11] e Arriscando [13] foram avaliados mediante a aplicação de

questionário. O questionário MEEGA+ [16] foi utilizado como modelo para coleta de dados nas avaliações conduzidas em [11] e [7], enquanto [8] e [10] utilizaram uma versão anterior do MEEGA+ [14]. Apesar de todos os trabalhos estarem relacionados à Computação, apenas [12] e [26] são relacionados à Arquitetura de Software. No entanto, em comparação com esta pesquisa, estes jogos não possuem tabuleiro, não abordam conceitos como atributos de qualidade e categorias de decisões arquiteturais. Além disso, não definem uma dinâmica clara para aprendizado.

4 METODOLOGIA

Design Science Research (DSR) consiste em um método de pesquisa que envolve a construção de artefatos para a produção de conhecimento e resolução de problemas reais [31]. Segundo Pimentel [17] a construção de um artefato não se limita apenas ao conhecimento técnico. Para seu desenvolvimento é necessário o conhecimento do ambiente em que aquele artefato está inserido e qual problema ele objetiva solucionar, considerando as ações das pessoas em um contexto, como uma organização funciona, suas práticas e sua cultura.

Além disso se faz necessária a realização de revisões literárias para que se caracterize conhecimento técnico e científico e a condução de avaliações para avaliar o problema encontrado e o artefato produzido a partir deste problema. Como a DSR pressupõe o desenvolvimento e a avaliação de artefatos para a construção de conhecimento, a presente pesquisa baseia-se neste método. Os passos adotados na pesquisa baseiam-se nas etapas de um ciclo de DSR definida por Wieringa [33] e define as técnicas adotadas em cada uma dessas etapas para o desenvolvimento do artefato apresentado posteriormente na Seção 5.

Uma pesquisa pode envolver a execução de vários ciclos de DSR ou ainda, somente algumas etapas de um ciclo [17]. Nesta pesquisa, um ciclo DSR inclui as seguintes etapas: investigação do problema (etapa 1), design da solução (etapa 2) e validação (etapa 3). No estágio atual da pesquisa, foi realizado um ciclo de DSR e iniciado um novo ciclo com a conclusão das etapas 1 e 2 que ainda será concluído futuramente com a realização da etapa 3.

O primeiro ciclo de DSR iniciou com a investigação do problema. Por meio da revisão da literatura, foram identificadas abordagens voltadas para o ensino em Computação, em particular, jogos educacionais não-digitais. De forma mais específica, buscou-se identificar jogos educacionais para o ensino de Arquitetura de Software. Como resultado, foram encontrados os trabalhos que propõem a criar e a investigar do uso de jogos para o ensino. Entretanto, não foram encontrados jogos que utilizassem elementos não-digitais para o ensino de conceitos (atributos de qualidade) e padrões relacionados à Arquitetura de Software.

Em seguida, iniciou-se o design da solução, com a proposta de criar um jogo de tabuleiro direcionado ao problema encontrado. Idealizou-se um jogo que incentivasse a interação e a competitividade entre os jogadores visando como objetivo final a aprendizagem. Foram definidos seus elementos físicos (tabuleiro, cartas e pinos) e a fonte para o conteúdo de aprendizagem, o livro "*Software Architecture in Practice - third edition*" [21].

O protótipo das cartas iniciais (Figura 1) definiu um padrão de design para elas. As cartas possuem: título da carta (1), imagem referente ao título (2), descrição sobre o título (3), afirmativa (4), resposta referente à afirmativa (5), pontos que a carta vale (6), imagem do tipo de baralho (7), nome do tipo de baralho (8). Após a prototipação, foram criadas as cartas, o tabuleiro e os pinos com o design específico a cada um. Em seguida, foi definida a dinâmica do jogo e regras. O resultado é descrito na Seção 5.

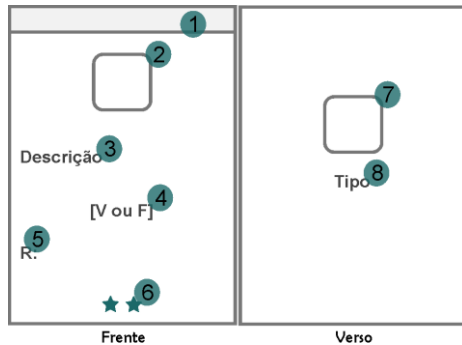


Figura 1: Protótipo das cartas iniciais.

A validação do artefato criado (o jogo) foi realizada através de um estudo de caso (etapa 3) com o objetivo de coletar dados para validar se a dinâmica do jogo promoveria uma boa experiência aos estudantes e se o conteúdo abordado auxiliaria no aprendizado. Com a validação, concluiu-se o primeiro ciclo de DSR. Contudo, foram obtidos dados que permitiram o início de um novo ciclo para realizar melhorias no artefato.

O segundo ciclo foi iniciado com a investigação do problema do ponto de vista dos usuários. Esta etapa consistiu na análise dos dados e dos feedbacks coletados durante o estudo de caso em que foram identificados os pontos fortes e os pontos em que o jogo poderia melhorar. Com isso, o artefato foi submetido novamente à etapa de design, resultando em uma nova versão (Seção 7).

5 PROPOSTA INICIAL DO JOGO

O LEARN Board Game consiste em um jogo de tabuleiro com o objetivo de apoiar o ensino e aprendizagem de conceitos de Arquitetura de Software e padrões arquiteturais. A ideia do jogo surgiu de uma necessidade identificada na disciplina de Arquitetura de Software, que prevê a aplicação prática de conceitos de arquitetura por meio de um projeto de solução arquitetural. O docente da disciplina notou que era necessária obter um entendimento claro dos conceitos envolvidos na tomada de decisões arquiteturais. Caso contrário, os estudantes criariam soluções arquiteturais sem considerar atributos de qualidade e a seleção de padrões arquiteturais adequados.

Por conter alguns conceitos abstratos como atributos de qualidade e conceitos que dependem de um certo conhecimento prévio em Computação como as decisões arquiteturais, os estudantes sentem dificuldades ao cursarem a disciplina. Com a criação do LEARN, os estudantes podem assimilar os conteúdos

de forma interativa por meio de uma dinâmica de perguntas e respostas. O jogo pode ser adotado na disciplina de Arquitetura de Software ou disciplinas que abordem conteúdos relacionados à Arquitetura de Software. A seguir é apresentada sua versão inicial.

5.1 Elementos do jogo

Plataforma: O jogo possui elementos inspirados na estrutura de jogos populares. O Banco imobiliário auxiliou na estruturação do tabuleiro e Perguntados (aplicativo móvel), reforçou o pensamento sobre aprendizagem através de perguntas e respostas. A união destes aspectos proporcionou a definição dos elementos do jogo. O LEARN é definido por elementos não-digitais e inicialmente composto por 39 cartas, 1 tabuleiro e 5 pinos.

Cartas: As cartas são subdivididas em baralhos da seguinte forma: 14 cartas de Atributo de qualidade, 11 cartas de Padrão Arquitetural, 14 cartas de Decisão Arquitetural. O conteúdo das cartas é baseado no livro “*Software Architecture in Practice*” [21].

Cartas de Atributo de Qualidade: Abordam os seguintes atributos: disponibilidade, interoperabilidade, modificabilidade, desempenho, segurança, testabilidade e usabilidade. Possuem a cor amarela (Figura 2).



Figura 2: Exemplo da carta de Atributo de Qualidade.

Cartas de Padrão Arquitetural: Refletem os seguintes padrões: Divisão em camadas, *Broker*, *Model-view-controller*, *Cliente-Servidor*, *Blackboard*, *Ponto-a-ponto (P2P)*, *Repositório*, *Piper-and-filter*, *Publish-Subscribe*, *Map-Reduce*, *Arquitetura Orientada a Serviços*. Possuem a cor vermelha (Figura 3).



Figura 3: Exemplo da carta de Padrão Arquitetural.

Cartas de Decisão Arquitetural: Abordam as seguintes decisões: Escolha da tecnologia, Modelo de dados, Mapeamento entre elementos, *Binding time*, Gerenciamento de recursos, Alocação de responsabilidades e Modelo de coordenação. São identificadas pela cor azul conforme apresenta a Figura 4.



Figura 4: Exemplo da carta de Decisão Arquitetural.

Tabuleiro: O tabuleiro (Figura 5) representa o campo de jogo que os jogadores percorrerão para compartilhar e adquirir novos conhecimentos de acordo com a dinâmica do jogo. Em seu interior, são dispostos três locais para os conjuntos de cartas, enquanto que a borda do tabuleiro é formada por 40 casas (quadrados).



Figura 5: Tabuleiro.

As casas do tabuleiro referenciam os baralhos de cartas. Assim, há uma casa inicial/final e mais quatro tipos de casas. Há em cada casa uma estrela, representando os pontos a serem adquiridos a cada jogada realizada com sucesso. Os pontos indicam a quantidade de casas que o jogador poderá avançar no tabuleiro.

Pinos: São diferenciados por cores distintas pois referenciam em que parte do jogo o jogador está. São entregues a cada jogador no início do jogo.

5.2 Dinâmica do jogo

Preparação: O LEARN deve ser jogado por grupos de no mínimo três e até cinco jogadores. Antes de iniciar, os baralhos de cartas

devem ser separados e dispostos nos espaços pertencentes a cada um no tabuleiro. Os pinos de cada jogador devem ser colocados na casa de “início”. Em seguida, deve ser realizada a definição dos papéis.

Papéis: O jogo contém dois papéis: (i) o Maestro é aquele que fará a pergunta e iniciará o jogo e (ii) o Respondente, é o próximo jogador (em sentido horário) do grupo e consequentemente terá que responder à questão levantada pelo Maestro para que avance nas casas. Na primeira rodada, como todos os pinos estarão na casa “início”, o Respondente pode escolher sobre qual baralho de cartas responderá.

Sistema de pontuação: Após cada resposta todos devem discutir o grau de assertividade que o Respondente obteve. Cada carta contém uma quantidade de estrelas (de uma à quatro). Logo, se a resposta estiver totalmente correta o Respondente recebe todas as estrelas e se estiver totalmente errada, nenhuma estrela será recebida pelo Respondente. Estas estrelas são utilizadas para avançar no tabuleiro, cada estrela ganha se avança uma casa.

Fluxo do jogo: Após a primeira jogada, o Respondente se torna o próximo Maestro e o jogador que está ao seu lado (no sentido horário do grupo) será o novo Respondente. Nas casas posteriores, a cor da casa define o tipo de carta a ser utilizada na jogada. Ou seja, o Maestro deverá fazer uma pergunta de acordo com a cor da casa atual em que o Respondente estiver. As respostas podem ser consultadas nas cartas para que o Maestro e demais jogadores julguem a pontuação do respondente.

Simulação: Supomos que há cinco jogadores numa partida do LEARN. O jogador 1 então será o primeiro Maestro. O jogador 2 como Respondente pede que o jogador 1 pergunte sobre uma carta de Decisão Arquitetural. Os jogadores verificam que essa carta vale duas estrelas. Após a resposta, todos os jogadores validam a resposta e verificam que o Respondente acertou totalmente o que era pedido. O jogador 2 então avança duas casas e para na casa de Atributo de Qualidade. Em seguida, o jogador 2 se torna o Maestro para o jogador 3 e assim sucessivamente.

6 ESTUDO DE CASO

Um estudo de caso fora conduzido com o objetivo de validar os elementos e a dinâmica do LEARN e captar *feedbacks* para que fosse possível realizar melhorias. O estudo de caso será descrito segundo as diretrizes propostas por Runeson et al. [30] para o relato de estudos de caso em Engenharia de Software.

Questões de pesquisa: O estudo de caso foi conduzido com o intuito de responder às seguintes questões de pesquisa sobre o LEARN:

- Q1. “O jogo possui usabilidade adequada?”
- Q2. “O jogo proporciona uma experiência positiva aos estudantes?”;
- Q3. “Os estudantes consideram que o jogo contribui para a aprendizagem de conceitos e padrões em Arquitetura de Software?”
- Q4. “A dinâmica do jogo permite que o jogo seja concluído durante a aula?.”;

Contexto e participantes: O jogo foi avaliado em ambiente acadêmico, o que configura o seu contexto real de uso, uma vez que seu intuito é auxiliar na aprendizagem. Além disso, o jogo foi adotado em uma turma de Arquitetura de Software alinhada aos conteúdos abordados no jogo. Esta disciplina é ofertada no curso de graduação em Engenharia de Software da Universidade Federal do Ceará. O docente ministrava a disciplina há 4 semestres e não havia adotado jogos educacionais em turmas anteriores. Vale ressaltar que o docente atuou como orientadora desta pesquisa.

Os participantes do estudo de caso foram 16 estudantes matriculados na disciplina. A turma continha 20 estudantes matriculados, porém 4 destes estavam ausentes no dia da adoção do jogo. Todos os estudantes estavam cursando a disciplina pela primeira vez e já haviam obtido uma base introdutória sobre Arquitetura de Software nas disciplinas de Projeto Detalhado de Software. Além do docente da disciplina, o primeiro autor deste artigo atuou como monitora durante a adoção do jogo.

Procedimentos de Coleta de dados: Para a coleta de dados sobre a experiência com o jogo, utilizou-se o questionário MEEGA+ [16]. O MEEGA+ [16] é um modelo para avaliar jogos educacionais em Computação de acordo com a percepção dos jogadores. Para isso, são considerados dois fatores de qualidade, a usabilidade e a experiência de jogo. Cada fator contém dimensões que possibilitam uma avaliação mais específica sobre cada ponto. A Tabela 1 apresenta as dimensões relacionadas a cada fator.

Tabela 1: Fatores de Qualidade do MEEGA+ [16].

Usabilidade	<ol style="list-style-type: none"> 1. Estética 2. Aprendizabilidade 3. Operabilidade 4. Acessibilidade
Experiência do Jogo	<ol style="list-style-type: none"> 1. Atenção focada 2. Diversão 3. Desafio 4. Interação social 5. Confiança 6. Relevância 7. Satisfação 8. Percepção de Aprendizagem

Cada dimensão contém afirmativas para serem avaliadas pelos participantes com base em uma escala Likert que varia desde 'Concordo fortemente' até 'Discordo fortemente', além de fornecer a opção de neutralidade.

Há também três questões abertas para que os participantes listem os pontos fortes, os pontos a serem melhorados, e apresentem comentários referentes à experiência. Assim, com base nas respostas das afirmativas de cada uma das categorias apresentadas e nas questões abertas pode-se extrair os dados avaliativos do jogo.

Outro procedimento adotado pelos pesquisadores foi a observação. Observou-se se a dinâmica do jogo permitia que os jogadores avançassem no jogo, se as regras estavam claras e se os participantes expressavam emoções positivas ou negativas ao jogar.

Preparação: Para que o estudante possa avançar no jogo, é necessário ter conhecimentos prévios sobre os conceitos, por meio de estudo dirigido ou aulas teóricas. Em aulas anteriores à adoção do LEARN, o docente conduziu aulas teóricas sobre os assuntos abordados pelo jogo: Atributos de Qualidade, Padrões Arquiteturais e Decisões Arquiteturais. Para explicar sobre o jogo, uma apresentação foi preparada em forma de slides. Além disso, foram confeccionados e impressos quatro conjuntos de jogo LEARN, cada um composto por 39 cartas, um tabuleiro e cinco pinos. Para a coleta dos dados foram impressos os questionários MEEGA+.

Execução: O estudo de caso foi realizado em uma sala direcionada para adoção de metodologias ativas na Universidade Federal do Ceará (UFC). A sala contém mesas redondas com cadeiras dispostas para permitir a realização de atividades em grupo. A Figura 6 ilustra o uso do jogo durante o estudo.



Figura 6: Utilização do LEARN Board Game.

Inicialmente, foi realizada uma breve apresentação sobre o jogo, com seu objetivo, elementos e sua dinâmica. Os participantes então se dividiram em grupos de três a cinco jogadores, receberam os materiais do jogo, organizaram os conjuntos de cartas e posicionaram os pinos na casa de "início" no tabuleiro.

Como a aula tinha duração de 2 horas, o tempo foi dividido da seguinte forma: 30 minutos para organização do ambiente e apresentação sobre o jogo; 1 hora para jogar o LEARN; e 30 minutos para o encerramento e preenchimento do questionário. Com o final do tempo estipulado para jogo, os estudantes de cada grupo que mais avançaram no tabuleiro do jogo foram considerados vencedores e premiados simbolicamente com adesivos e broches.

Após o tempo determinado para jogar, os participantes receberam o questionário para avaliarem o jogo. Por fim, os pesquisadores iniciaram uma discussão com os estudantes sobre as observações feitas enquanto eles jogavam. Essa discussão abordou questões como: "por qual tipo de carta começaram o jogo?", "a carta escolhida para fazer uma pergunta era relacionada à casa em que estava o Maestro ou o Respondente?" e "a dinâmica para avançar no jogo poderia considerar a utilização de outros elementos ou regras?".

Análise dos dados: Os dados obtidos por meio dos questionários foram tabulados para a geração de gráficos. As respostas às questões abertas e as observações realizadas ao longo do estudo foram transcritas para identificar possibilidades de melhorias no design da solução. A próxima seção apresenta os resultados obtidos a partir desta análise.

7 RESULTADOS E DISCUSSÃO

Os resultados obtidos serão descritos com dois focos: resultados sobre o jogo obtidos no estudo de caso. A avaliação do jogo foi realizada mediante as respostas ao questionário e as respostas às questões levantadas na discussão.

Para responder à questão de pesquisa Q2, foram analisados os dados do Meega+ referentes ao fator de qualidade **usabilidade** (Figura 7), Quanto à **estética** (1.1 e 1.2), a maioria dos participantes (15) concordam que o jogo é atraente. Mas três discordam que as fontes e as cores do texto são consistentes.

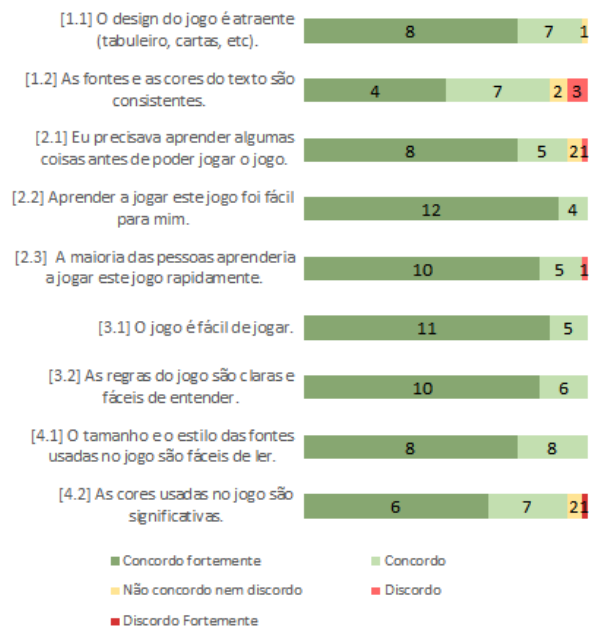


Figura 7: Dados sobre a usabilidade do jogo

Em relação à **aprendizabilidade** (2.1 à 2.3), houve total concordância de que o LEARN é um jogo fácil de aprender. Já quando afirmado que era preciso aprender algumas coisas antes de jogar, dois respondentes demonstraram neutralidade e apenas uma pessoa discordou em dizer que a maioria das pessoas aprenderiam rapidamente como jogar. Nos aspectos que avaliam a **operabilidade** (3.1 e 3.2) todos concordaram que o jogo é fácil de jogar e que suas regras são claras e fáceis de entender. Quanto à **acessibilidade** (4.1 e 4.2), treze pessoas concordaram que as cores utilizadas são significativas e todos que o tamanho e o estilos das fontes são fáceis para a leitura.

Para responder à questão de pesquisa Q2, foram analisados os dados do MEEGA+ referentes ao fator de qualidade experiência do jogador (Figura 8).

Em relação à **confiança** (1.1 e 1.2), houve total concordância de que a forma de organização do jogo ajudou os participantes a confiarem de que aprenderiam com ele e a primeira impressão fora de que seria fácil para eles. Em relação ao **desafio** (2.1 à 2.3) todos os respondentes concordaram que o LEARN é adequadamente desafiador, porém, dois participantes demonstraram neutralidade em dizer que o jogo oferece novos desafios em ritmo apropriado. Sobre a afirmação “o jogo não se torna monótono à medida em que avança” obteve-se a maior grau de discordância (sete pessoas discordaram com esta afirmativa). Em relação à **satisfação** (3.1 à 3.4), 14 indicaram que a conclusão de tarefas proporcionou sensação satisfatória de realização. Sobre a afirmativa “É devido ao meu esforço pessoal que consegui avançar no jogo”, três respondentes se mostraram neutros.

Todos sentiram-se satisfeitos com as coisas que aprenderam jogando e 15 recomendariam o jogo aos colegas. Em relação à **interação social** (4.1 à 4.3), foram obtidos resultados com concordância total à todas as afirmativas, “Consegui interagir com outras pessoas durante o jogo”, “O jogo promove a cooperação e/ou competição entre os jogadores” e “Eu me senti bem interagindo com outros jogadores durante o jogo”. Em relação à **diversão** (5.1 e 5.2), 15 participantes concordaram que se divertiram e que algo aconteceu durante o jogo que os fizeram sorrir. Em relação à **atenção focada** (6.1 à 6.3), 3 pessoas discordaram que algo no começo do jogo chamou a atenção e que perderam a noção do tempo por estar muito envolvido com o jogo. Enquanto que participantes 14 concordam que esqueceram o ambiente ao redor enquanto jogavam.

As dimensões relevância e percepção de aprendizagem respondem à questão de pesquisa Q3. Assim, em relação à **relevância** (7.1 à 7.4), a afirmativa “O conteúdo do jogo é de meu interesse” recebeu o maior índice de neutralidade (4 respostas), enquanto que todos concordam que o conteúdo do jogo está relacionado com a disciplina, treze estudantes concordam que é o método adequado para o ensino e dez preferem aprender com o LEARN do que com outros métodos.

Por fim, a **percepção de aprendizagem** (8.1 e 8.2) mostra que para 14 respondentes o jogo contribuiu para o aprendizado na disciplina e para a maioria deles (15) o jogo permitiu um aprendizado eficiente em comparação com outras atividades práticas na disciplina.

Sobre a Q4, durante o estudo foi percebido que o tempo de 1h disponibilizado aos estudantes jogarem não foi suficiente para concluir uma volta no tabuleiro. Com a mudança na dinâmica do jogo durante o design da segunda versão, um melhor resultado pode ser obtido em um novo estudo.

Conforme sugerido por Petri et al. [34], foi realizado um cálculo para avaliar a qualidade do jogo e o valor obtido foi igual a 45,6, que indica um bom nível de qualidade para o LEARN.

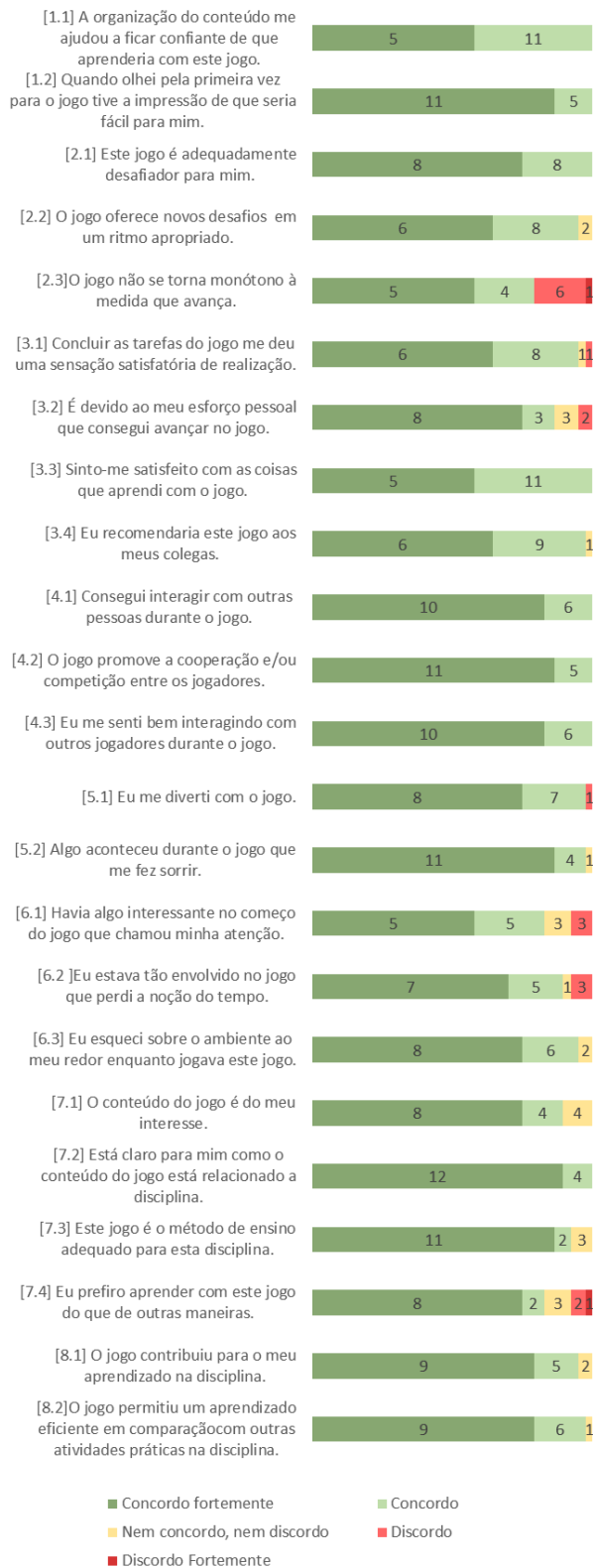


Figura 8: Dados sobre a experiência do jogador.

Este índice apresentado pelo cálculo conclui que o jogo apresenta momentos de interação e diversão, contribui com eficiência para a aprendizagem dos jogadores e é fácil de jogar, no entanto, ainda pode ser melhorado em termos de design.

Além dessas afirmativas, foram analisadas as questões abertas respondidas pelos estudantes. Em relação aos pontos fortes, os pontos mais citados foram que o jogo é atraente, proporciona uma nova forma de aprendizado e é bastante interativo. A Tabela 2 apresenta alguns dos comentários (C1, C2, C3) que referenciam esta afirmação. Quanto às sugestões para melhorar o jogo, foram citados comentários para a adição de cartas coringas e diminuição das casas de Decisão Arquitetural (C4, C5).

Tabela 2: Comentários fornecidos.

Pontos Fortes	
C1	“Coloração atraente, forma dinâmica de aprendizado e interação com outros jogadores para debates das questões.”
C2	“Muito divertido, as cores são bem chamativas, adquire muito conhecimento.”
C3	“Interação entre os jogadores, fácil de aprender e estimula a concentração para entender as questões.”
Sugestões de melhorias	
C4	“Cartas coringas com a possibilidade de conteúdos aleatórios e cartas pretas com punições”.
C5	“Diminuir as cartas de decisão arquitetural pois são muito difíceis, adicionar cartas coringas e revisar as descrições das cartas”.

Como os participantes não finalizaram o jogo no período estipulado, ao final do estudo, foi questionado entre os estudantes se seria válido considerar outros mecanismos para o avanço no jogo, como o uso de dados. No entanto, avançar com estrelas foi apontado como uma forma diferente e mais atrativa.

8 VERSÃO ATUAL DO LEARN BOARD GAME

Os resultados das avaliações apontaram aspectos a serem melhorados no jogo, por esta razão foi realizada novamente a etapa de design da solução, focando no redesign dos elementos do jogo. Nesta seção, serão apresentadas as mudanças realizadas no design do jogo. A versão atual do jogo está disponível em [35].

Cartas: foram modificados os tons das cores utilizadas nas cartas, para que as cores se tornassem mais significativas.

Sistema de pontuação: a quantidade de estrelas nas cartas foi aumentada para melhorar o ritmo do avanço do jogador no tabuleiro. Agora os pontos são de duas, quatro ou seis estrelas.

Cartas de Sorte ou Revés: foram criadas e adicionadas ao LEARN mediante as sugestões extraídas do questionário de avaliação. Estas cartas contribuem para tornar o jogo mais dinâmico. Caso o jogador retire uma carta de sorte, ele é beneficiado em alguma jogada, já quando ele retira uma carta de

revés ele é penalizado. A Figura 9 apresenta uma carta de Sorte ou Revés desse conjunto que possui a cor verde para identificação.



Figura 9: Exemplo da carta de Sorte ou Revés.

Tabuleiro: foram inseridas as casas de Sorte ou Revés, assim como um espaço para alocar o conjunto no tabuleiro durante o jogo. As cores foram atualizadas de acordo com os novos tons e um logotipo com o nome do jogo foi adicionado. Além disso, diminuiu-se o número de casas de Decisão Arquitetural. A Figura 10 apresenta a versão atual do tabuleiro do LEARN após às melhorias.

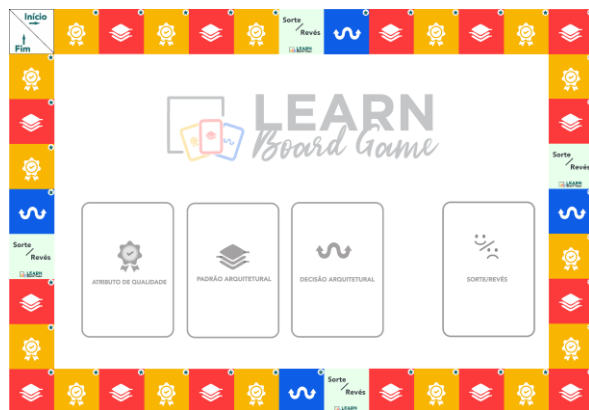


Figura 10: Versão atual do tabuleiro.

Regras: para que esse processo seja respeitado, foram criadas regras de execução, conforme descritas abaixo.

1. Quando estiverem na casa de início, os Respondentes podem optar por responderem à pergunta de qualquer um dos conjuntos de cartas.
2. A seleção da carta pelo Maestro deve ser de acordo com a casa em que se encontra o Respondente no tabuleiro. Exceto se o Respondente estiver na casa inicial. Assim, quem escolhe sobre qual assunto irá responder, é o Respondente.
3. Os papéis não se repetem em uma mesma rodada, ou seja, a cada pergunta e resposta deve-se obedecer a ordem de próximo onde o Respondente se torna o próximo Maestro.
4. Um Maestro nunca pode se auto fazer uma pergunta, já que nas cartas estão descritas as respostas.
5. A cada resposta, a equipe deve discutir sobre a pontuação atribuída ao respondente, analisando o nível corretude da resposta (resposta correta ou resposta incorreta).

6. O ganhador é aquele que consegue concluir uma volta no tabuleiro ou estiver na frente se um tempo limite para o jogo for pré-estabelecido.

Após todas as alterações descritas, foi realizada uma revisão detalhada do conteúdo das cartas para garantir a qualidade do texto e aprimorado o layout do tabuleiro. É possível que o desempenho do estudante no jogo indique seu nível de conhecimento sobre os conceitos abordados, já que o estudante avança no jogo à medida em que responde corretamente as questões sobre os conceitos.

9 AMEAÇAS À VALIDADE

No que diz respeito à representatividade dos participantes e generalização dos resultados, como o LEARN foi proposto para o ensino de conceitos de Arquitetura de Software, estudantes de graduação em processo de aprendizagem na disciplina de Arquitetura de Software são considerados representativos como participantes do estudo. No entanto, por meio de novos estudos, novos feedbacks podem ser obtidos para melhorias no jogo.

Em relação ao viés do pesquisador durante a execução do estudo, os estudantes foram informados que o jogo era uma versão inicial e que sua avaliação tinha o objetivo de evoluí-lo, para que os estudantes não tivessem receio de fornecer críticas construtivas ao jogo. Embora o uso do jogo tenha sido uma atividade da disciplina, compondo uma nota de participação, a avaliação do jogo não gerava nota para a turma, sendo opcional preencher o questionário de avaliação.

10 CONCLUSÕES E TRABALHOS FUTUROS

O presente artigo apresentou o LEARN Board Game, um jogo de tabuleiro para a aprendizagem de conceitos e padrões em Arquitetura de Software. O jogo foi criado seguindo a metodologia de *Design Science Research*. A validação da proposta inicial do jogo foi realizada por meio de um estudo de caso, onde foram coletados dados sobre a usabilidade e experiência do jogador e extraídas possíveis melhorias.

Os resultados indicaram que o jogo proporcionou uma experiência positiva aos estudantes visto que grande parte das afirmativas presentes no questionário foram bem avaliadas pelos participantes. Como por exemplo o quesito “interação social”, em que todos os participantes concordaram que o jogo promove a interação entre os jogadores e no quesito “relevância”, onde todos afirmaram que está claro como o conteúdo do jogo está relacionado com a disciplina.

A avaliação também permitiu captar possíveis melhorias como a inserção de cartas coringas para dinamizar o jogo, algo constatado pela discordância de sete participantes quando afirmado que o jogo não se torna monótono à medida em que avança. Além disso, sugeriu-se diminuir as cartas de Decisão Arquitetural pelo grau de dificuldade ser considerado maior em relação aos outros conteúdos. A observação e discussões durante o estudo permitiram definir regras mais claras no jogo.

Uma nova versão do jogo foi criada com bases nas análises dos dados e feedbacks do estudo de caso. Espera-se que o jogo seja adotado em diferentes ambientes acadêmicos que objetivem o ensino de Arquitetura de Software. Seu design e dinâmica podem também ser adaptados para o ensino de outros conteúdos de Engenharia de Software, fornecendo soluções para a educação. Devido ao isolamento social, ainda não foi possível realizar os estudos planejados para o semestre 2020.1 que visam avaliar o novo design e dinâmica do jogo. Como trabalhos futuros será realizado um experimento em uma turma com um docente externo à pesquisa será conduzido. A nova avaliação do jogo a fim de analisar as modificações feitas para esta nova versão e criar novas questões para estender a aprendizagem em Arquitetura de Software.

AGRADECIMENTOS

Os autores agradecem à participação dos respondentes da pesquisa e ao Laboratório Interdisciplinar de Computação e Engenharia de Software (LINCE).

REFERÊNCIAS

- [1] Steferson L. C. Ferreira and Rosario Girardi. 2002. Arquiteturas de Software baseadas em Agentes: do Nível Global ao Detalhado. *Revista Eletrônica de Iniciação Científica da SBC*.
- [2] Sandra Schröder, Mohamed Soliman and Matthias Riebisch. 2018. Architecture enforcement concerns and activities-an expert study. *Journal of Systems and Software* 145: 79-97.
- [3] Eng L. Ouh and Yunghans Irawan. 2018. Teaching Adult Learners on Software Architecture Design Skills. *IEEE Frontiers in Education Conference (FIE)*.
- [4] Daniela Rosca, William Tepfenhart and James McDonald. 2003. Software engineering education: following a moving target. *Proceedings 16th Conference on Software Engineering Education and Training (CSEET'03)*.
- [5] Craig Caulfield, Jianhong C. Xia, David Veal and S Maj. 2011. A systematic survey of games used for software engineering education. *Modern Applied Science*, 5(6), 28-43.
- [6] John V. Dempsey, Barbara Lucassen, and Karen Rasmussen. , 1996. The instructional gaming literature: Implications and 99 sources. South Carolina: University of South Carolina, College of Education.
- [7] Gabriel G. Moreira and Anna B. S. Marques. 2018. Evaluating the students' experience with the Scrum Card Game: an experience report in a Software Engineering course. In *Proceedings of the 17th Brazilian Symposium on Software Quality (SBQS)*. ACM, New York, NY, USA, 344-353. DOI:<https://doi.org/10.1145/3275245.3275288>
- [8] A. Brito and J. Vieira. 2017. "2TScrum": A Board Game to Teach Scrum. In *Proceedings of the 31st Brazilian Symposium on Software Engineering (SBES'17)*. Association for Computing Machinery, New York, NY, USA, 279-288. DOI:<https://doi.org/10.1145/3131151.3131177>
- [9] Victor Travassos Sarinho. 2019. Masters of the Process: A Board Game Proposal for Teaching Software Management and Software Development Process. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering (SBES 2019)*. Association for Computing Machinery, New York, NY, USA, 532-536. DOI:<https://doi.org/10.1145/3350768.3352459>
- [10] Victor Moura and Gleison Santos. 2018. ProcSoft: A Board Game to Teach Software Processes Based on ISO/IEC 29110 Standard. In *Proceedings of the 17th Brazilian Symposium on Software Quality (SBQS)*. Association for Computing Machinery, New York, NY, USA, 363-372. DOI:<https://doi.org/10.1145/3275245.3276319>
- [11] T. Yevgrashyn. Scrum Card Game. Disponível em: <<https://scrumcardgame.com/>> Acesso em: 07 de Abril de 2020.
- [12] Courtney Schriek, Jan M. E. M. va der Werf, Antony Tang and Floris Bex. 2016. Software architecture design reasoning: A card game to help novice designers. In *European Conference on Software Architecture* (pp. 22-38). Springer, Cham.
- [13] Sebastião Santos, Flávia Carvalho, Yandson Costa, Davi Viana, and Luis Rivero. 2019. Risking: A Game for Teaching Risk Management in Software Projects. In *Proceedings of the XVIII Brazilian Symposium on Software Quality (SBQS'19)*. Association for Computing Machinery, New York, NY, USA, 188-197. DOI:<https://doi.org/10.1145/3364641.3364662>
- [14] Rafael Savi. 2011. Avaliação de jogos voltados para a disseminação do conhecimento. Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC.
- [15] Arnold M. Lund. 2001. Measuring usability with the use questionnaire12. *Usability interface*, 8(2), 3-6.
- [16] Giani Petri, Christiane G. von Wangenheim, Adriano F. Borgatto. 2017. Evolução de um Modelo de Avaliação de Jogos para o Ensino de Computação. *XXV Workshop sobre Educação em Computação*, pp.2327-2336 .
- [17] Mariano Pimentel, Denise Filippo. and Flávia M. Santoro. 2019. Design Science Research: fazendo pesquisas científicas rigorosas atreladas ao desenvolvimento de artefatos computacionais projetados para a educação. *Metodologia de Pesquisa em Informática na Educação: Concepção da Pesquisa*. Porto Alegre: SBC.
- [18] Paulo E. Battistella, Christiane G. von Wangenheim and João M. Fernandes. 2014. Como jogos educacionais são desenvolvidos? Uma revisão sistemática da literatura. *Workshop sobre Educação em Computação*, Brasília, Brasil.
- [19] Márcia Souza and César França. 2020. O que explica o sucesso de jogos no ensino de engenharia de software? Uma teoria de motivação. *Anais do XXIV Workshop sobre Educação em Computação*. SBC.
- [20] Giani Petri, Alejandro Calderón, Christiane G. von Wangenheim, Adriano F. Borgatto and Mercedes Ruiz. 2018. Benefícios dos jogos não-digitais no ensino de computação. In *Anais do XXVI Workshop sobre Educação em Computação*. SBC.
- [21] Len Bass, Paul Clements and Rick Kazman. 2003. *Software architecture in practice*, 2nd ed. Addison-Wesley Professional.
- [22] Bettina Biel and Volker Gruhn. 2009. Towards a Method for Analyzing Architectural Support Levels of Usability. In *2009 Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture* (pp. 273-276). IEEE.
- [23] Jens Knodel and Daniel Popescu. 2007. A comparison of static architecture compliance checking approaches. In *2007 Working IEEE/IFIP conference on software architecture (WICSA'07)* (pp. 12-12). IEEE.
- [24] Harrison, Neil B., and Paris Avgeriou. "Leveraging architecture patterns to satisfy quality attributes." *European conference on software architecture*. Springer, Berlin, Heidelberg, 2007.
- [25] David Garlan, 2000. Software architecture: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering* (pp. 91-101).
- [26] Humberto Cervantes, Serge Haziyeu, Olha Hrytsay, and Rick Kazman. 2016. Smart decisions: an architectural design game. In *Proceedings of the 38th International Conference on Software Engineering Companion (ICSE '16)*. Association for Computing Machinery, New York, NY, USA, 327-335. DOI:<https://doi.org/10.1145/2889160.2889184>.
- [27] José R. Xavier, Cláudia M. L. Werner, and Guilherme Horta Travassos. 2002. Uma abordagem para a seleção de padrões arquiteturais baseada em características de qualidade. *XVI Simpósio Brasileiro de Engenharia de Software*, Gramado, RS, Brasil.
- [28] Paulo Battistella and Christiane G. von Wangenheim. "Games for teaching computing in higher education—a systematic review." *IEEE Technology and Engineering Education* 9.1 (2016): 8-30.
- [29] ISO/IEC 25010:2011. *System and Software Engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – System and Software Quality Models*. 2011.
- [30] Per Runeson, Martin Host, Austen Rainer, and Bjorn Regnell. 2012. *Case study research in software engineering: Guidelines and examples*. John Wiley & Sons.
- [31] Fernando H. Zaidan., Marcello P. Bax, and Fernando S. Parreiras. 2016. Design Science Research: Application in a Research and Development Project. In *3th International Conference on Information Systems & Technology Management-CONTECSI*.
- [32] Thomas M. Connolly, Mark Stansfield, and Thomas Hainey. "An application of games-based learning within software engineering." *British Journal of Educational Technology* 38.3 (2007): 416-428.
- [33] Wieringa, Roel. "Design science as nested problem solving." *Proceedings of the 4th international conference on design science research in information systems and technology*. 2009.
- [34] Giani Petri, Christiane G. von Wangenheim, and Adriano F. Borgatto. 2018. MEEGA+: A Method for the Evaluation of Educational Games for Computing Education. *INCoD–Brazilian Institute for Digital Convergence*: 1-47.
- [35] Tamires Ariane and Anna Beatriz Marques. 2020. LEARN Board Game: A game for teaching Software Architecture created through Design Science Research. <https://doi.org/10.6084/m9.figshare.12774773.v2>