

Processo de Aprendizagem da Metodologia Ágil Scrum com Blocos de Lego em Jogos Acadêmicos Interativos: Ponto de Vista dos Estudantes

Douglas Augusto Barcelos Bica and Carlos Alexandre Gouvea da Silva, *Member IEEE*

Title—Learning Process of Agile Scrum Methodology with Lego Blocks in Interactive Academic Games: Viewpoint of Students

Abstract—O rápido crescimento da tecnologia da informação levou ao desenvolvimento de uma grande variedade de sistemas computacionais em todo o mundo. Como resultado, o número de instituições de ensino que oferecem cursos em áreas como programação e engenharia de software aumentou. No entanto, os processos tradicionais de desenvolvimento de software não acompanharam as mudanças nas tecnologias. Nos últimos anos, o desenvolvimento de software tornou-se mais dinâmico e iterativo, exigindo que as partes interessadas trabalhassem em equipes e entregassem projetos de maior qualidade em menos tempo usando métodos como o Desenvolvimento Ágil (por exemplo, Scrum e Extreme Programming (XP)). Embora algumas instituições abordem esse conteúdo nos cursos de graduação, muitos alunos e professores são indiferentes a ele, resultando em baixo entusiasmo e prática. Este artigo apresenta um caso real de uma atividade em sala de aula para ensinar conceitos do Scrum usando blocos de Lego. Ao final, os alunos foram convidados a avaliar a eficácia da atividade. Os resultados mostraram que jogos dinâmicos e atividades práticas são mais eficazes que aulas teóricas ou em vídeo.

Index Terms—Aprendizagem na Educação, Desenvolvimento de Software, Processo Ágil, Scrum

I. INTRODUÇÃO

NOS últimos anos, o desenvolvimento de software tem sido definido como um conjunto de atividades de manufatura [1]. Tradicionalmente, o processo era pensado como uma série de atividades inter-relacionadas [2], que envolviam definição de etapas do projeto, análise de viabilidade, engenharia de requisitos, design, codificação, testes de verificação e validação.

Artigo em versão Português recebido 22 Março de 2020; aceito 31 Março de 2020.

Artigo original em inglês recebido 19 de Julho de 2019; revisado 12 Novembro de 2019 e 5 de Março de 2020; aceito 18 de Março de 2020; data da publicação xx Abril de 2020; data da versão atual xx Abril de 2020.

D. A. B. Bica do Departamento de Sistemas de Informação e Desenvolvimento Ágil, Centro Universitário de Araucária (Unifacear), Araucária 83707-067, Paraná, Brasil; e-mail:dougbrasil.bica@gmail.com.

C. A. G. Silva do Departamento de Sistemas de Informação e Desenvolvimento Ágil, Centro Universitário de Araucária (Unifacear), Araucária 83707-067, Paraná, Brasil; carlos.gouvea@ieee.org

Na literatura, processos clássicos como modelos em cascata e em forma de V (*V-shaped*) não refletiam a natureza iterativa do desenvolvimento exploratório.

Os processos tradicionais são historicamente importantes no desenvolvimento de software, no entanto, as metodologias iterativas e evolutivas de Desenvolvimento Ágil os substituíram. O desenvolvimento ágil usa desenvolvimento aditivo e iterativo, repetindo várias fases de um processo e melhorando o software, analisando o *feedback* do cliente que converge em novas soluções [2]. As principais metodologias ágeis usadas hoje são Scrum e *Extreme Programming* (XP) [3].

O desenvolvimento ágil é aceito pela maioria das empresas de tecnologia, no entanto, é uma metodologia desafiadora, pois tende a sofrer resistência de todos os níveis organizacionais [4]. Nerus et al. [5] indica que quatro questões-chave tendem a dificultar a migração para o desenvolvimento ágil: 1) Gerenciamento e cultura organizacional, estilo e sistema de recompensas; 2) Relacionamento da equipe de desenvolvimento e competência técnica; 3) A mudança de um desenvolvimento centrado no processo para um orientado à recursos e iterativo; 4) A existência de tecnologias e ferramentas adequadas. Scharff et al. [6] mostram que a maioria das questões de desenvolvimento está relacionada ao ritmo de desenvolvimento diferente e à distância entre as partes interessadas e a cultura. Ainda assim, metodologias ágeis estão sendo aceitas entre as organizações tradicionais, e a maioria delas indicando uma preferência por manter os dois estilos de desenvolvimento [7].

Atualmente, os profissionais devem estar preparados para enfrentar desafios dentro das empresas. Nos cursos de graduação, o Scrum e o XP são amplamente utilizados nas aulas e as duas abordagens se concentram em atender às necessidades de aprendizagem do aluno [8]. Ensinar metodologias ágeis em instituições educacionais tende a ser mais eficaz se for feito em um ambiente ágil: valorizando os alunos que se encaixam melhor no método, desenvolvendo suas habilidades através da prática real [8]. Scrum é uma Metodologia Ágil que se concentra no gerenciamento de projetos e permite o desenvolvimento iterativo. No entanto, o processo de aprendizado do Scrum não tende a ser eficaz, devido a ambientes de aprendizado inadequados e tutores inexperientes que levam à alunos desmotivados. Considerando isso, muitas instituições têm usado metodologias ativas para melhorar o processo de

aprendizado e ensino [9][10].

Tradicionalmente, o processo de aprendizado clássico é baseado em dois elementos: professores e alunos. Nesse caso, o professor transmite seu conhecimento para uma classe enquanto os alunos ouvem as informações, fazem anotações ou respondem perguntas. Esse processo de aprendizagem permite a aquisição de novos esquemas mentais, conhecimentos, habilidades, entre outros, nos quais poderia ser usado para resolver problemas elementares ou complexos, utilizando ferramentas, simuladores, equações matemáticas, teorias ou análises analíticas. Nos últimos anos, o processo de aprendizagem evoluiu de maneiras diferentes. Felder e Silverman [11] descreveram esse processo como uma estrutura de duas etapas nas quais os alunos recebem informações e, em seguida, processam o que deve ser memorizado. Para Prince e Felder [12], o processo de ensino e aprendizagem é baseado em um conjunto de vários métodos instrucionais, incluindo aprendizado baseado em perguntas, em problemas, em projetos, ensino baseado em casos, descoberta aprendizagem e ensino *just-in-time*. Armstrong e Fukami [13] definem a aprendizagem como um processo holístico de adaptação, no qual envolve o total de funções integradas das pessoas, como pensar, sentir, perceber e se comportar para a solução de problemas, tomada de decisão e criatividade.

Atualmente, são apresentados vários métodos para ensinar Scrum, como jogos [14][15], simuladores, softwares virtuais [16][17], videoaulas, palestras, entre outros. Para alguns artigos, por exemplo em [18][19][20], o Scrum é usado para apoiar o desenvolvimento de projetos de software reais, o que pode melhorar não apenas a produtividade da implementação do software, mas também a compreensão dos alunos sobre o Scrum. No entanto, esses trabalhos não apresentaram a perspectiva dos alunos e seus sentimentos em relação à aprendizagem.

Propomos a análise de uma atividade dinâmica do Scrum baseada em blocos de Lego, aplicada nas aulas de graduação. O principal valor inovador deste artigo é contribuir com a adaptação de um jogo interessante para os alunos entenderem e aplicarem os principais conceitos do Scrum, aprendendo a trabalhar em equipe, com responsabilidades e gerenciamento de tempo. No final do experimento, perguntamos aos alunos qual a eficácia desse método em comparação com outros que eles experimentaram. Com base no feedback dos alunos, espera-se que eles aumentem seu conhecimento sobre os tópicos do Scrum, como etapas do processo do Scrum, funções principais e artefatos. Quando comparado a outros processos tradicionais de aprendizagem, espera-se dos alunos desta atividade melhores sentimentos e boas motivações.

Na seção II, a metodologia Scrum é apresentada. A seção III apresenta brevemente algumas das técnicas e métodos usados para ensinar Scrum em ambientes educacionais. A seção IV explica o jogo utilizado para ensinar Scrum e como ele melhora o aprendizado dos alunos. Na seção V, os resultados são mostrados e algumas discussões são levantadas. Na seção VI, são apresentadas conclusões e ideias para trabalhos futuros.

II. FRAMEWORK SCRUM

A origem do Scrum pode ser atribuída ao artigo "*The new new product development game*", publicado em 1986 por Takeushi e Nonaka na Harvard Business Review [21]. No artigo, os autores usam uma analogia que compara jogos de rugby e equipes de desenvolvimento de produtos. Em 1993, Jeff Sutherland e Ken Schwabe aplicaram os conceitos sugeridos por Takeushi e Nonaka no ambiente de negócios da Easel Corp. A aplicação desses conceitos passou a ser conhecida como Scrum. Inicialmente, começou como o desenvolvimento de produtos comerciais em um processo de iteração de 30 dias [22]. No entanto, o uso do Scrum para desenvolvimento de software não foi considerado.

Em 1995, Ken Schwaber publicou um artigo com o título "*Scrum development process*", no qual ele introduziu conceitos de Scrum para desenvolvimento de software [23]. Desde então, vários ajustes foram feitos na metodologia, mas os fundamentos ainda são os mesmos. O Scrum pode ser descrito como uma estrutura (*framework*) que considera o processo de desenvolvimento como um conjunto flexível de atividades, combinando ferramentas e técnicas para fazer com que as equipes de desenvolvimento alcancem os melhores resultados possíveis na construção de sistemas. Além disso, o Scrum é um ciclo de desenvolvimento incremental, composto por entregas periódicas de curto prazo [24], que utiliza iterações para obter o produto final de software. Na Figura 1 é mostrada uma visão geral do processo Scrum.

Dentro do Framework Scrum três funções são definidas: o dono do produto (PO, *Product Owner*); a equipe de desenvolvimento (DT, *Development Team*) ou equipe Scrum; e o Scrum Master (SM). As principais responsabilidades e atividades de cada um no Scrum são apresentadas na Tabela I.

Tabela I
RESPONSABILIDADES E ATIVIDADES DO PO, DT E SM [20]

Envolvido	Atividades
PO	O PO obtém o financiamento inicial e contínuo para a execução do projeto, gera os requisitos funcionais e não funcionais iniciais do sistema e os objetivos de investimento esperado. A lista de requisitos e demais atividades é chamada <i>Product Backlog</i> (PB), na qual o PO usa para definir quais recursos têm o valor mais alto e quais devem ser desenvolvidos inicialmente. O PB é atualizado continuamente pelo PO enquanto o sistema é construído nas iterações.
DT	DT é responsável por desenvolver funcionalidades do software a partir da lista de prioridades do PO com base no PB. A equipe desenvolve o PB com base em incrementos de funcionalidade nas iterações usando o conhecimento técnico de todas as partes interessadas.
SM	O SM é responsável pelo processo Scrum, por ensinar todas as regras e práticas da metodologia, auxilia na eliminação de dificuldades e impedimentos que podem prejudicar o projeto e facilita reuniões e decisões importantes.

Inicialmente, o PO mostra à equipe uma lista com todas as funcionalidades que devem ser desenvolvidas. Essas funcionalidades são conhecidas como histórias de usuário (US, *User Stories*). Uma das partes mais difíceis desse processo é definir quanto esforço será necessário para desenvolver cada US. Para resolver esse problema, algumas técnicas são usadas. A

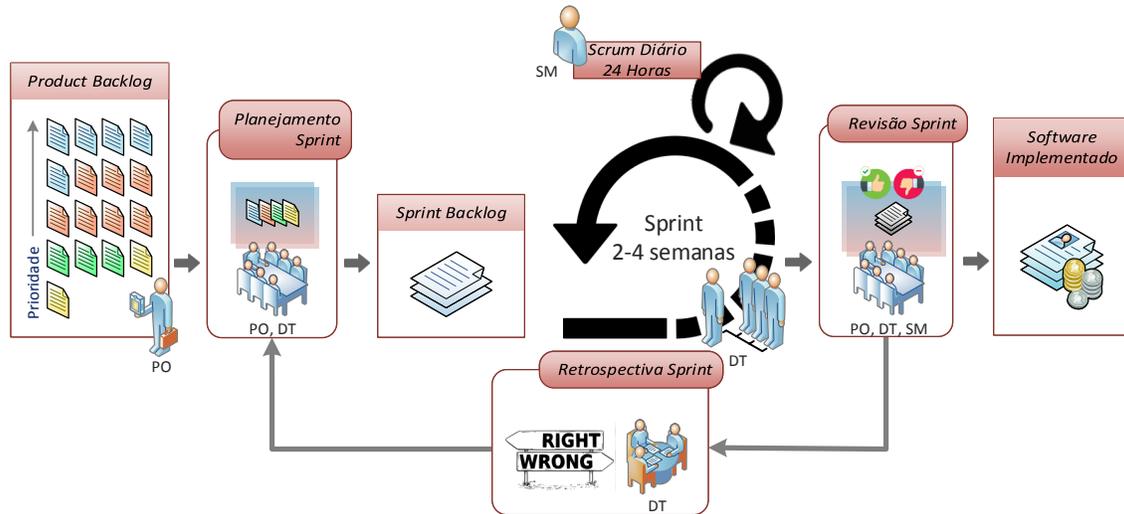


Figura 1. Sequência de etapas do framework Scrum.

maioria das metodologias ágeis recomenda o uso do *Planning Poker* para calcular uma estimativa de esforço de cada US para ser desenvolvida pela DT [25]. Na maioria dos casos, a definição de esforço necessário é definida pela DT, escolhendo um valor com base na sequência numérica de Fibonacci (1, 2, 3, 5, 8, ..., máximo) [26]. Um valor mais baixo de Fibonacci representa um esforço baixo para construir uma US e um valor máximo representa o esforço mais alto. Normalmente, um número variando de 0,5 a 20 é usado para descrever o esforço de cada US [27], mas outras sequências também podem ser usadas. Em algumas situações, especialmente em equipes pequenas, é comum definir cada esforço baseado na quantidade de horas de trabalho [28].

O Scrum divide o projeto em Sprints (iterações), mas antes de iniciar um Sprint, é necessário executar um planejamento de Sprint (SP, *Sprint Planning*) essencial para garantir o sucesso do projeto [29]. Um SP é uma reunião na qual o PO indica quais as US são mais cruciais que devem ser desenvolvidos no próximo Sprint. Nesta etapa, o DT considera apenas as US que podem ser executados no próximo Sprint, evitando as que exigem mais esforço do que o disponível. Esta reunião resultará em um Sprint Backlog (SB). O SB é uma lista das US que o DT se compromete construir e concluir no Sprint [30].

Durante um Sprint o DT desenvolve todas as atividades e ações necessárias para construir as US de acordo com os requisitos da PO. Durante esta fase, o SB não deve mudar. Devido à necessidade de comunicação entre as partes interessadas são realizadas reuniões diárias [31]. Essa reunião diária do Scrum (*Daily Scrum*) ocorre em aproximadamente em 15 minutos, na qual a equipe tem a oportunidade de compartilhar seu trabalho realizado e relatar qualquer obstáculo existente. Geralmente, os membros da equipe relatam uma das três informações:

- O que foi feito desde a última reunião;
- O que eles planejam fazer até a próxima reunião;
- Indicar quaisquer impedimentos que estejam no seu caminho.

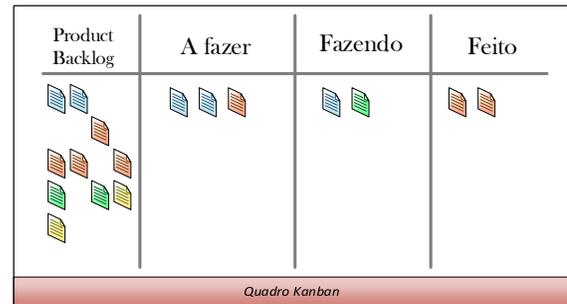


Figura 2. Representação de um quadro Kanban clássico.

Durante as reuniões diárias do Scrum, os membros da equipe atualizam duas ferramentas que permitem a fácil comunicação e visualização do processo de desenvolvimento, ou seja, o Kanban e o Burndown. O Kanban é um quadro que tem como objetivo manter atualizado as US em desenvolvimento no Sprint atual. O modelo de quadro mais comum é dividido em quatro colunas: “A fazer”, “Em andamento”, “Feito” e o PB, como mostra a Figura 2. A US que ainda não foram iniciados são colocados na coluna “Product Backlog”, os em desenvolvimento são colocados na coluna “Em andamento” e as US finalizados são colocados na coluna “Concluído”. Durante todo o processo, as US são movidos entre cada coluna, permitindo uma melhor compreensão do desempenho da equipe de acordo com a situação de cada US.

No *Daily Scrum*, o DT atualiza o gráfico do quadro *Burndown*. O gráfico *Burndown* mostra a cada dia uma nova estimativa de quanto tempo (esforço ou horas) falta até que todos os US estejam concluídas. O primeiro dia de uma Sprint é apresentado por um valor que indica a soma de todos os esforços estimados durante o SP. Espera-se que esse valor seja zero no último dia da Sprint. Portanto, uma linha “ideal” de valor máximo de esforço até zero é desenhada, como mostra a Figura 3. A cada dia, o esforço restante é estimado e indicado no gráfico *Burndown*. Se o esforço restante estiver acima do

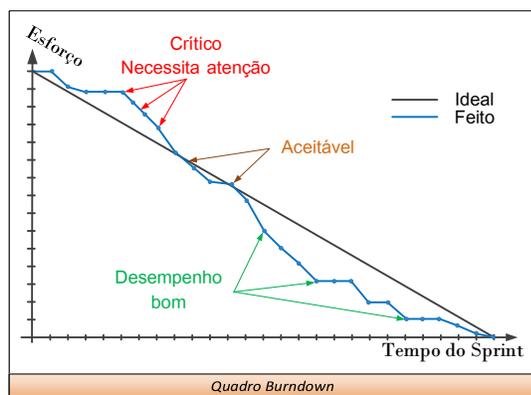


Figura 3. Gráfico de *Burndown* com visão de linha e desempenho ideais.

valor esperado, o desempenho está baixo; caso contrário, o desempenho é considerado alto.

No final de um Sprint as US são entregues. Essa etapa é chamada de *Sprint Review* (SRv) e consiste em mostrar ao PO o que foi desenvolvido. A PO analisa se o produto entregue corresponde ao que foi solicitado, fornecendo um nível de controle de qualidade. Eventualmente, as US que não foram desenvolvidas ou não foram consideradas aceitáveis pelo PO retornam em um futuro Sprint.

Após o SRv, uma Retrospectiva do Sprint (SRT) ocorre no final do Sprint. Nesta fase, o DT analisa o desempenho dos membros da equipe e indica como eles podem melhorar nos futuros Sprints. Durante essa etapa é comum a equipe questionar aspectos como:

- O que funcionou bem ou não no Sprint?
- O que causou problemas, não funcionou corretamente ou não correu bem?
- O que eles podem fazer de diferente no próximo Sprint para melhorar o desempenho da equipe e evitar problemas?

Depois que o SRv e o SRT forem concluídos, o PO poderá atualizar o PB com modificações ou novas US, se necessário. Nesta etapa, o DT pode iniciar outro ciclo de Sprint com base em uma nova reunião de SP.

III. TRABALHOS RELACIONADOS

O processo de aprendizagem pode ser afetado por vários fatores, incluindo baixa motivação, inadequados métodos de ensino e ferramentas em sala de aula inapropriadas. O uso de atividades práticas é uma alternativa interessante ao processo de aprendizado do Scrum. Nas universidades, os alunos aprendem a desenvolver habilidades e atitudes usando atividades práticas [32]. Nesta seção, mostramos uma breve visão geral dos trabalhos que usaram diferentes métodos para ensinar o Scrum no ensino superior.

Lee [16] sugere o uso do Scrum-X, um software que ajuda os alunos a melhorar sua compreensão da metodologia Scrum. O Scrum-X é uma plataforma de simulação usada para aprimorar os conceitos do Scrum por meio de exercícios experimentais. O objetivo é gerenciar um projeto de software, desenvolvendo uma página *web* de vendas de livros, abrangendo quase todos os conceitos do Scrum. O impacto do

aprendizado deste software foi demonstrado em um projeto com 15 alunos divididos em três grupos. Ao final, foi utilizado um questionário para avaliar a eficácia do experimento. Os resultados indicam um potencial positivo e os autores relatam um impacto positivo na motivação do aluno para aprender e se engajar.

Kropp et al. [33] sugeriu uma atividade Scrum usando blocos de Lego, nos quais os alunos precisam construir uma cidade. A construção foi realizada em 4 sprints de 25 minutos cada. Os autores observaram que as equipes começam com um Sprint fraco, mas através do experimento atingiram uma velocidade ideal. Os resultados apresentados relatam que os alunos gostaram da abordagem prática, apreciando a intensa comunicação e colaboração dos membros da equipe. A maioria dos estudantes tendia a aprovar documentos em reuniões muito rapidamente e se concentravam na entrega ao cliente, embora reconhecessem a importância de escrever as US, uma tarefa que consideravam difícil, mas útil. Steghofer et al. [34] também apresentou os resultados de um experimento com 450 alunos participantes dos *Workshops* Scrum que usaram blocos de Lego para construir uma cidade em vários Sprints curtos.

Com o objetivo de avaliar o impacto das atividades práticas, Castro et al. [35] usou o Scrum para simular o desenvolvimento de componentes eletrônicos simples. O quadro Kanban foi aprimorado pelo SM para acompanhar as US desenvolvidas pela equipe, com cada membro fazendo sprints de 6 minutos cada. No final de cada sprint, uma carta era retirada de um baralho com penalidades ou benefícios, adicionando dinamismo à simulação. Foi aplicado um questionário avaliativo com notas de 1 a 5 para quantificar o impacto do experimento na motivação do aluno, em comparação ao ensino teórico. Segundo o autor, foi alcançado melhor aprendizado, motivação e satisfação do aluno, apesar de os alunos enfrentarem dificuldades interpessoais.

Os jogos têm ganhado visibilidade aplicadas como ferramentas educacionais [36][37][38]. Usando blocos de Lego, Paasivaara et al. [39] usou um jogo para ensinar Scrum em uma universidade finlandesa. Durante o jogo, os alunos se familiarizaram com as funções, etapas, eventos e conceitos do Scrum, simulando vários Sprints, incluindo o planejamento e a construção de um produto a partir de blocos de Lego. Os resultados mostram que os alunos apresentaram altos níveis de aprendizado e satisfação.

A maioria dos trabalhos desta seção apresentou resultados sobre a motivação e o interesse dos alunos em atividades que usam softwares ou blocos de Lego para ensinar Scrum. Embora os alunos tenham recebido questionários para avaliar esses resultados, não foram perguntados se essa atividade é melhor para o ensino de Scrum do que outras atividades tradicionais ou aulas em sala. A principal contribuição deste trabalho é também apresentar essa perspectiva dos alunos.

IV. MATERIAIS E MÉTODOS

Para ensinar Scrum a estudantes de graduação, organizamos *Workshops* usando blocos de Lego e os convidamos a participar. Três oficinas foram organizadas em duas universidades diferentes no Brasil. As informações sobre as oficinas são mostradas na Tabela II.

Tabela II
INFORMAÇÕES SOBRE OS WORKSHOPS

Workshop	Universidade	Curso de graduação	Número de estudantes
1º	X	Sistemas de Informação	14
2º	Y	Engenharia Elétrica	20
3º	X	Sistema de Informação	27

Cada workshop apresentou o Scrum a calouros e estudantes que não conheciam o método ou como ele funcionava. Os alunos do primeiro ano também foram convidados, para permitir a compreensão do processo de aprendizagem em diferentes níveis de conhecimento.

Após uma introdução ao Scrum, apresentamos o desafio de construir uma cidade usando blocos de Lego usando a metodologia. A sequência das etapas executadas neste workshop é mostrada na Figura 4. O jogo Lego City consistia em quatro sprints, incluindo o tempo de execução de cada etapa.

Para a realização da atividade, foram definidos três tipos de envolvidos: um PO, que era o professor e elaborador do workshop, e que era responsável por manter o PB atualizado; um SM foi nomeado para colaborar com o DT e manter todos os membros da equipe envolvidos na metodologia Scrum; e, finalmente, os alunos participantes desempenharam o papel de DTs.

Os alunos foram divididos em grupos de 5 ou 6 integrantes. Cada grupo teve que construir diferentes partes da cidade, com base em uma lista de requisitos estabelecidos pelo PO. Primeiro, o PO apresentou 18 prioridades das US para os estudantes, como mostrado na tabela III. As prioridades foram listadas de acordo com a importância: 1-Baixa, 2-Média e 3-Alta. Os alunos tinham que entender que as US de alta prioridade deveriam ser construídas primeiro. Cada UE tinha requisitos específicos, exigindo que os estudantes os construíssem de maneira correta e respeitando sua prioridade. Se as US não seguissem os padrões (por exemplo, cores, peças, quantidade ou tamanho), o pedido não seria aceito pelo PO.

Na Figura 5, o quadro Kanban com o PB é apresentado. Como os workshops foram realizados em meses diferentes, a apresentação da US foi aprimorada, imprimindo o Kanban em um grande papel impresso. No primeiro workshop (5a), as US foram apresentadas no quadro da sala; no último workshop (5c), foram apresentadas em *post-it* coloridos.

Após a apresentação do PO, cada grupo realizou um *Planning Poker* para estimar o esforço necessário para construir cada US. Nesta etapa, a DT analisou cada US e definiu um valor baseado na sequência de Fibonacci, no qual 1 significa baixo esforço devido à baixa complexidade, seguido pelos valores 2, 3, 5 e 8. O esforço 8 é o mais alto, devido à alta complexidade. Consideramos que em cada oficina o esforço pode ser estimado em diferentes valores de acordo com a perspectiva da DT. Portanto, uma média foi estimada (ou seja, se uma US tivesse um esforço estimado de 2, 2 e 3, consideramos a média do esforço de 2). Com base na estimativa, somamos todo o esforço das US para ter o esforço total da atividade de construção da cidade e o usamos no gráfico de Burndown, como mostra a Figura 6. Embora cinco

sprints estivessem disponíveis para os alunos em cada oficina, isso não foi necessário em nenhum caso realizado.

Antes de iniciar cada Sprint, realizamos um SP, no qual o PO informou das US de alta prioridade e solicitou que um grupo aceitasse a responsabilidade de construí-lo. Cada grupo de trabalho verificou se eles tinham a capacidade de construí-lo. Nesse caso, um dos grupos aceitou cada US. Em alguns casos, um grupo pôde aceitar US que exigissem mais esforço do que outras equipes, o que não representa um problema, pois todas as equipes foram capazes de concluir as tarefas completas.

Durante os Sprints, cada grupo começou a construir suas respectivas US aceitas, com seu progresso sendo constantemente atualizado no gráfico de Burndown de "A fazer" para "Em fazendo" e de "Em fazendo" para "Feito". No Sprint final, o PO, DT e SM se reuniam e executavam o SRv para verificar cada US de cada grupo. Se uma tarefa foi executada corretamente, o PO aceitava e declarava que a US havia terminado. Caso uma US tenha sido rejeitado pelo PO, ela era remarcado para um futuro Sprint. Quando as US eram finalizadas, o valor do Esforço no gráfico Burndown era diminuído para mostrar o desempenho da Equipe de Desenvolvimento. Antes da próxima etapa, todas as equipes se reuniam em um SRT para analisar como melhorar seu desempenho. O próximo sprint foi iniciado e esse processo foi repetido até que todas as US fossem desenvolvidas.

Depois que todas as iterações foram feitas, cada aluno listou as principais vantagens do Scrum. Para validar o processo de aprendizado, coletamos feedback usando um formulário. As perguntas são apresentadas na Tabela IV.

Observe que as questões 3 e 4 medem o nível de conhecimento antes e depois do jogo. Ambas as questões usam a escala Likert para perguntar aos alunos se o reconhecimento é muito baixo (1), baixo (2), nem alto nem baixo (3), alto (5) ou muito alto (5) [40]. Além disso, nas questões 5, 6, 7 e 8, pergunte sobre quais envolvidos e artefatos os alunos conhecem antes e depois do jogo.

A questão 12 inclui opções como: livro, artigo ou manual (atividades de leitura); videoaulas; aulas em áudio; ensino teórico em sala de aula; projetos reais usando um tempo de desenvolvimento; jogos reais ou online; aplicativos para smartphones; emulação de software ou simulação educacional; discussão em grupo; jogos educativos; no trabalho ou estágio; eventos de certificação sobre Scrum; seminário ou palestras. A pergunta 13 inclui opções como: assistir a uma aula, ler um artigo técnico sobre Scrum, realizar uma atividade acadêmica ou profissional prática ou assistir a um vídeo.

V. RESULTADOS

Nesta seção, apresentamos os resultados divididos em duas partes. Primeiro, descrevemos as atividades e os resultados observados durante os workshops a fim de mostrar como os alunos interagiram dentro da atividade. Finalmente, o resultado do questionários coletados são avaliados e discutidos para apresentar como os alunos adquiriram conhecimentos sobre Scrum e como a atividade de construção da cidade de Lego melhorou o aprendizado sobre o Scrum.

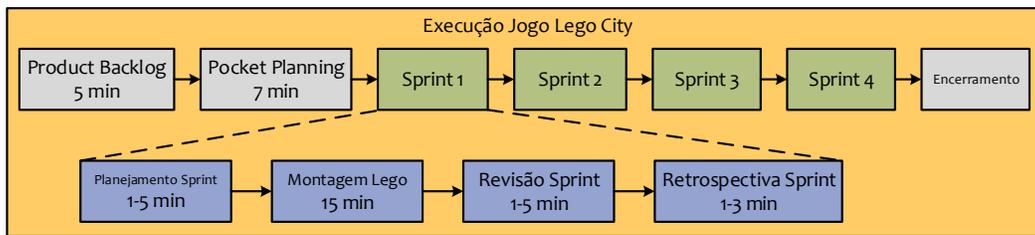


Figura 4. Sequência de etapas de execução do jogo Lego City com base na metodologia Scrum.

Tabela III
LISTA DE US DO PB GERADA PELO PO E SUAS PRIORIDADES.

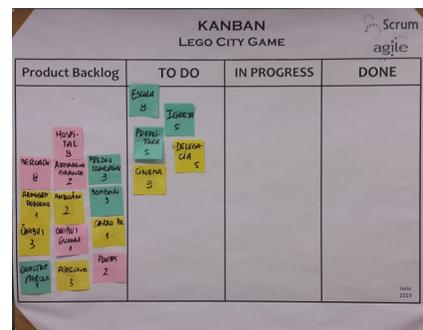
US	Prioridade	Item	Quantidade	US	Prioridade	Item	Quantidade
1	3	Escola	1	10	1	Pequeno armazém	1
2	3	Igreja	1	11	2	Ambulância	1
3	3	Prefeitura	1	12	2	Caminhão de bombeiros	1
4	3	Delegacia	1	13	2	Viaturas	1
5	3	Hospital	1	14	1	Ônibus	1
6	3	Cinema	1	15	2	Ônibus escolar	1
7	2	Supermercado	1	16	1	Esculturas no parque	1
8	2	Prédio comercial	1	17	1	Pontos de ônibus	5
9	2	Grande armazém	1	18	1	Acessórios esportivos (escola)	3



(a)



(b)



(c)

Figura 5. Quadro Kanban com US do PO: (a) primeiro Workshop, (b) segundo Workshop e (c) terceiro Workshop.

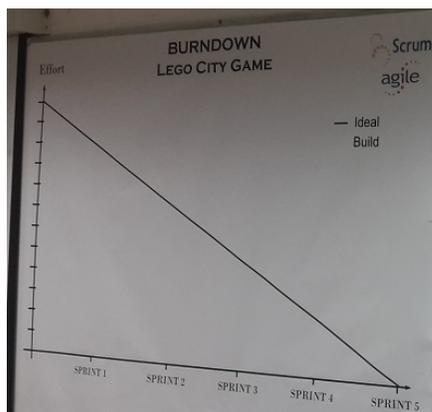


Figura 6. Quadro de burndown usados nos Workshops.

A. Atividades dos Workshops

Inicialmente, os grupos realizaram uma reunião do *Planning Poker* para definir o esforço necessário para realizar as dezoito US definidos pelo PO. A Tabela V mostra o esforço estimado

para cada DT nos três workshops. O esforço total foi aplicado em cada gráfico de Burndown. Durante o *Planning Poker*, observamos que os estudantes estavam preocupados com vários aspectos da definição dos valores do esforço. Em particular, alguns estudantes tentaram estimar o esforço com base no número de blocos que seriam necessários para construir cada US. Isso foi considerado importante para tomar decisões que afetariam o desempenho da equipe. O trabalho em equipe entre as partes interessadas também foi um dos principais avanços nesta etapa do jogo. Alguns estudantes tiveram medo de falar sobre o esforço das US, mas depois de alguns minutos começaram a interagir com outros membros. Em todos os casos, o senso de liderança de alguns membros melhorou o progresso das tarefas em todos os workshops.

Depois de definir o esforço necessário de todas as US, cada equipe começou a usar os blocos de Lego para construir os US. No entanto, a escolha de quais US seriam construídos em cada Sprint considerava a prioridade estabelecida pelo PO e a capacidade da equipe de desenvolver cada US. A Tabela VI mostra os resultados do esforço total necessário em cada Sprint. Podemos observar que, no primeiro e no segundo

Tabela IV
QUESTIONÁRIO DE AVALIAÇÃO.

Questão e descrição	Alternativas
Q1 - Você já conhecia ou ouviu falar sobre o Scrum antes de participar desse workshop?	Sim-Não
Q2 - Você já participou de alguma atividade acadêmica ou profissional sobre o Scrum antes desse workshop?	Sim-Não
Q3 - Em uma escala de 1 a 5, como você considera o seu conhecimento sobre o Scrum antes desse workshop?	Muito baixo (1) - Muito alto (5)
Q4 - Em uma escala de 1 a 5, como você passou a considerar o seu conhecimento sobre Scrum depois do workshop?	Muito baixo (1) - Muito alto (5)
Q5 - Selecione os envolvidos do Scrum que você conhecia antes do workshop.	SM, PO, DT ou Nenhum deles
Q6 - Selecione os envolvidos do Scrum que você passou a conhecer depois workshop.	SM, PO, DT ou Nenhum deles
Q7 - Selecione os artefatos que você conhecia antes do workshop.	PB, SB, Kanban, Burndown, SRv, SRt ou Nenhum deles
Q8 - Selecione os artefatos que você passou a conhecer depois do workshop.	PB, SB, Kanban, Burndown, SRv, SRt ou Nenhum deles
Q9 - Em uma escala de 1 a 5, como você considera a aplicação dessa atividade de ensino de metodologias ágeis utilizando um jogo para ensinar Scrum?	Pouco útil (1) - Muito útil (5)
Q10 - Em uma escala de 1 a 5, você concorda que com o jogo você pode aprender mais que em uma aula teórica em sala?	Discordo totalmente (1) - Concordo totalmente (5)
Q11 - Em uma escala de 1 a 5, você considera que esse jogo é um diferencial na sua formação acadêmica comparada com outros estudantes que apenas tiveram aulas teóricas?	Discordo totalmente (1) - Concordo totalmente (5)
Q12 - Entre as alternativas, selecione as opções que você pensa ser melhor para ensinar o Scrum comparado ao workshop que participou.	Diversas opções
Q13 - Entre as alternativas, selecione a melhor opção para um aluno que nunca ouviu falar do Scrum possa aprender a metodologia mais rapidamente.	Diversas opções

Tabela V
PLANNING POKER COM O ESFORÇO NECESSÁRIO ESTIMADO PELOS ESTUDANTES.

US	DT 1	DT 2	DT 3
1	3	5	8
2	5	3	5
3	5	5	5
4	3	3	5
5	8	8	8
6	3	2	3
7	3	3	8
8	5	5	3
9	3	2	2
10	2	1	1
11	1	1	2
12	1	1	3
13	1	1	1
14	1	3	3
15	1	1	1
16	2	1	1
17	2	1	2
18	1	3	3
Esforço total	50	49	64

workshop, a DT aceitou menos US no Sprint 1 do que no Sprint 2, isso porque as equipes não tinham conhecimento

real de sua capacidade de construção. Nesse sentido, o Sprint 2 foi mais crítico porque mostrou o resultado das premissas de desempenho feitas no Sprint 1. O primeiro Sprint do terceiro workshop apresentou maior valor de esforços das US do que em outras. Podemos relacionar esse alto valor do esforço devido ao maior número de estudantes ser 27 (em comparação com 14 e 20 dos outros workshops). Devido ao alto número de estudantes, as US foram construídas mais rapidamente, precisando de apenas três Sprints para concluir o projeto.

Tabela VI
DESEMPENHO DOS ESTUDANTES EM CADA WORKSHOP.

Sprint	Esforço total DT 1	Esforço total DT 2	Esforço total DT 3
1	13	16	26
2	22	18	18
3	9	6	20
4	6	9	-
Esforço total	50	49	64

A Figura 7 mostra os gráficos Burndown de cada DT. Em todos os casos, os estudantes construíram todas as US e foram motivados a usar os conceitos do Scrum para criar a cidade dos blocos de Lego. Um dos desafios do ensino foi fazer com que os estudantes entendessem a importância de atualizar o Kanban. Era necessário avisar os alunos para não esquecerem de atualizar o status (A fazer, Em andamento e Feito) de cada US. Os estudantes ficaram mais confiantes quando as US foram atualizadas em tempo real, permitindo a visualização de cada atividade.

A Figura 8 mostra a cidade final de Lego. Uma das grandes vantagens desses workshops não é exatamente o fato de os estudantes estarem aprendendo novos conteúdos ou uma nova metodologia, mas a maneira como as atividades são realizadas. Um ambiente criativo e divertido faz com que os alunos se sintam fora do estado mental acadêmico, melhorando seu processo de aprendizado devido a técnicas incomuns. Ambientes criativos são aceitos positivamente, especialmente entre pessoas de 18 a 25 anos, porque são usados para mudar rapidamente conceitos como tecnologia e jogos eletrônicos. Um jogo baseado em Scrum busca criar um ambiente interativo e colaborativo entre os alunos. Entre outras características desenvolvidas, podemos citar: responsabilidade, trabalho em equipe, bom design, aderência a prazos, organização do espaço de trabalho, respeito às etapas da metodologia e comunicação constante.

B. Avaliação dos estudantes

Inicialmente, qualquer aluno poderia participar das oficinas, sem que nenhum conhecimento prévio de Scrum fosse obrigatório. Nesse contexto, verificou-se que 93,6% de todos os estudantes possuíam algum conhecimento sobre scrum. Quando perguntados se eles já haviam participado anteriormente de uma atividade do Scrum, verificamos que 66% dos alunos tinham alguma experiência em workshops ou tarefas do Scrum.

Com base na primeira e segunda perguntas, pedimos aos estudantes que considerassem seu nível de conhecimento em

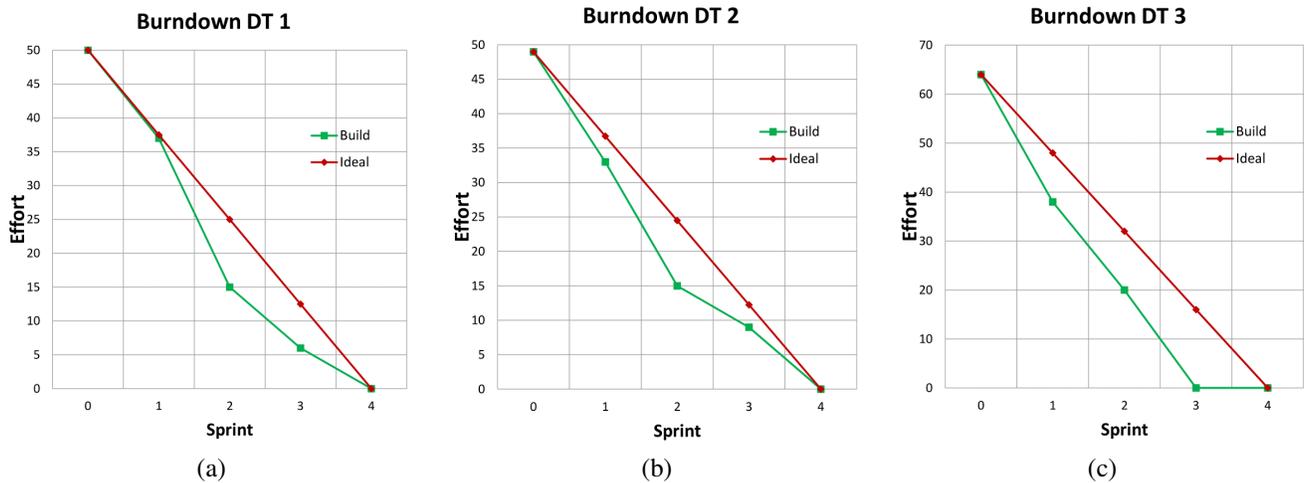


Figura 7. Quadro de desempenho Burndown (a) primeiro Workshop, (b), segundo Workshop e (c) terceiro Workshop.

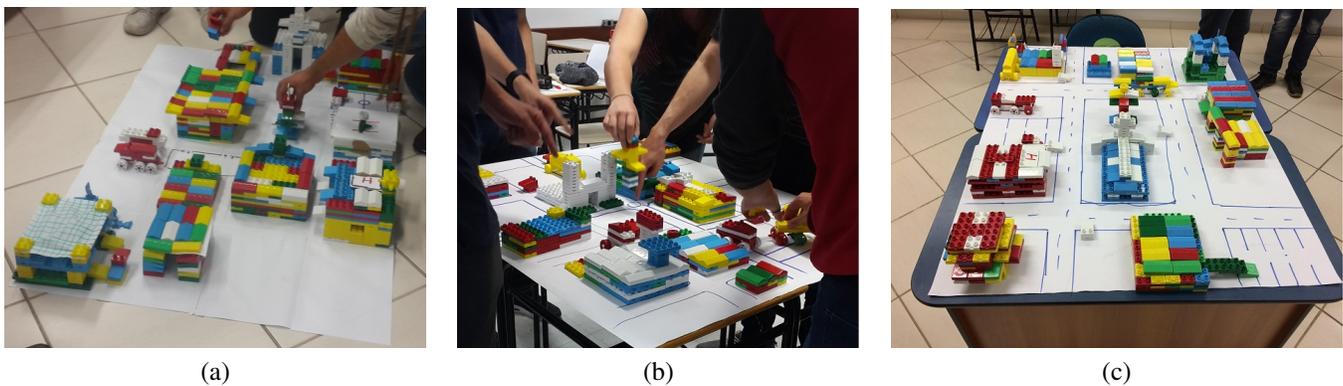


Figura 8. Resultados finais: (a) primeiro Workshop, (b) segundo Workshop, e (c) terceiro Workshop.

Scrum antes e depois de participar do workshop. A Tabela VII apresenta os resultados obtidos com o feedback do estudante. É possível verificar que antes do workshop, os alunos que consideravam o nível de conhecimento 1, 2 ou 3 eram de 74,5% e após o workshop o percentual foi reduzido para 8,5%. Esses feedbacks mostram quanto o workshop aumentou seus conhecimentos. Os alunos que consideraram seu conhecimento alto (4) ou muito alto (5) passaram de 25,5% para 91,5% após o workshop. Não houve casos de estudantes que reduziram seu nível de conhecimento, com a maioria subindo 1 ou 2 níveis.

Tabela VII
NÍVEL DE CONHECIMENTO DO SCRUM ANTES E DEPOIS DO WORKSHOP.

Nível de conhecimento	Antes	Depois
Muito baixo (1)	12,8%	0,0%
Baixo (2)	25,5%	2,1%
Normal (3)	36,2%	6,4%
Alto (4)	21,3%	40,4%
Muito alto (5)	4,2%	51,1%

A tabela VIII mostra os resultados das questões 5 e 6. Os alunos foram questionados sobre qual envolvido do Scrum eles conheciam antes e quais eles conheceram após o workshop. Entre os alunos que não tinham conhecimento do Scrum, os resultados foram positivos. Infelizmente, muitos estudantes indicaram que conheciam “Equipe de desenvolvimento” por-

que tinham alguma experiência em projetos acadêmicos que usavam esse termo e o relacionavam ao conceito específico de equipe de desenvolvimento em Scrum. Devido à possibilidade de selecionar mais de uma opção, a soma dos valores pode ser maior que 100%.

Tabela VIII
PERCENTUAL DE CONHECIMENTO SOBRE OS ENVOLVIDOS ANTES E DEPOIS DO WORKSHOP.

Envolvido do Scrum	Antes	Depois
Scrum Master	85,1%	93,6%
Product Owner	72,3%	93,6%
Development Team	76,6%	97,9%
Conhecia nenhum deles	4,3%	2,1%

A variedade de elementos no Scrum pode representar um desafio para os estudantes. Nas questões 7 e 8, avaliamos quais artefatos do Scrum os estudantes conheciam antes de participar. Os alunos puderam escolher mais de uma resposta, resultando em uma soma maior que 100%. Na tabela IX, os resultados dessas questões podem ser observados. Em todos os casos, os alunos indicaram um aumento do conhecimento sobre os artefatos do Scrum. Alguns estudantes que não conheciam o Scrum indicaram conhecer o Kanban devido ao uso dessa técnica em outros cenários, como empresas, fábricas e escritórios. Os elementos que mais conheciam foram

Burndown (de 34% para 85,1%), Sprint Retrospective (de 40,4% para 80,9%) e Sprint Review (de 46,8% para 87,2%). Isso se deve ao fato de esses elementos raramente serem apresentados aos alunos durante atividades teóricas e práticas.

Tabela IX
PERCENTUAL DE ARTEFATO DO SCRUM QUE O ESTUDANTE CONHECE ANTES E DEPOIS DOS WORKSHOPS.

Artefato do Scrum	Antes	Depois
Product Backlog	72,3%	87,2%
Sprint Backlog	72,3%	87,2%
Kanban	78,7%	80,9%
Burndown	34,0%	85,1%
Sprint Retrospective	40,4%	80,9%
Sprint Review	46,8%	87,2%
Nenhum deles	8,5%	0,0%

Com base nos resultados das Tabelas VII, VIII e IX, é possível verificar que o *feedback* dos estudantes indica uma melhor compreensão dos tópicos do Scrum após o jogo.

A Figura 9 mostra os resultados obtidos nas questões Q9, Q10 e Q11. As escalas são apresentadas nos valores de 1 a 5, em que 1 representa inútil e 5 representa muito útil para a questão 9. Nas questões 10 e 11, os valores representam 1 (discordo totalmente) até 5 (concordo totalmente). A pergunta 9 mostra que todos os alunos consideram o uso de atividades como jogo com Lego para ensinar ao Scrum é um método muito eficaz. Confirmamos esse feedback no Q10, que mostra mais de 85% dos alunos acham que aprender Scrum a cantar Lego City é mais eficaz do que aulas, slides e aulas teóricas. Quando perguntado na pergunta 11, se os alunos consideram a cidade de Lego uma vantagem competitiva em comparação com os alunos que tiveram apenas aulas teóricas, os resultados variaram. Os alunos que concordaram plenamente foram maioria com cerca de 72,3%, mas também indicaram as respostas número 2, 3 e 4, com 4,3%, 10,6% e 12,8%, respectivamente.

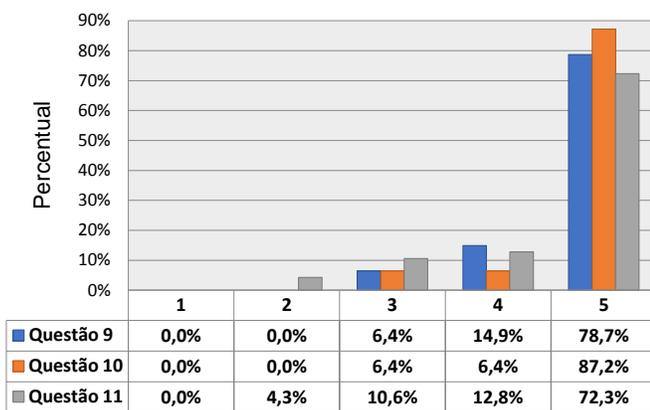


Figura 9. Feedback das questões Q9, Q10 e Q11.

As perspectivas de atividades que podem auxiliar o aprendizado dos estudantes são apresentadas na Tabela X. A questão Q12 pergunta aos alunos sobre as diferentes maneiras pelas quais o processo de aprendizado do Scrum pode ser aprimorado, apresentando diferentes alternativas. Os resultados mostraram que cerca de 70% dos estudantes escolheram atividades reais de projeto com uma equipe de desenvolvimento

real e 66% escolheram no trabalho ou durante um estágio. As atividades consideradas menos efetivas foram: leitura de livros e artigos e áudio-aulas. Além disso, observamos que apenas 12,8% das opiniões estão relacionadas às aulas teóricas, mostrando que o ensino tradicional pode gerar baixa motivação e eficácia. Estes resultados são baseados principalmente no sentimento dos estudantes. Vale ressaltar que não foi proposto neste trabalho aplicar outras ferramentas para ensinar Scrum, como assistir a vídeos, desenvolver projetos reais no processo Scrum, ler livros, ter aulas tutoriais, entre outros.

Tabela X
PERSPECTIVAS DOS ALUNOS SOBRE MELHORES MANEIRAS DE APRENDER METODOLOGIAS SCRUM.

Método de ensino de Scrum	Percentual
Projetos e equipe de desenvolvimento reais	70,2%
No trabalho ou em estágios	66,0%
Jogos ou games online	40,4%
Jogo educacionais	27,7%
Aplicativos para celulares	23,4%
Vídeo aulas	19,1%
Seminários ou palestras	17,0%
Realizando exames de certificação Scrum	12,8%
Softwares ou simuladores educacionais	12,8%
Grupos de discussão	12,8%
Aulas teóricas em sala de aula	12,8%
Livros, artigos e manuais (apenas leitura)	6,4%
Áudio-aulas	6,4%

Finalmente, na questão 13, os estudantes foram questionados sobre a melhor alternativa para aprender Scrum rapidamente. Os alunos puderam escolher apenas uma alternativa. Os resultados mostram que 91,5% dos estudantes consideram atividades práticas em ambientes acadêmicos ou profissionais são mais eficazes do que palestras (4,3%), ler livros (0%) ou assistir a vídeos (4,3%).

C. Trabalhos comparados

Nesse sentido, o objetivo da avaliação é analisar o jogo com blocos de Lego no que diz respeito à motivação para a aprendizagem no contexto de cursos de tecnologia. Com base na perspectiva de motivação dos estudantes observada nos questionários, realizamos um estudo comparativo com os trabalhos anteriores indicados na seção III. Uma comparação dos resultados com base na motivação dos estudantes é mostrada na Tabela XI, na qual focamos nas perguntas 9, 10 e 11, onde os estudantes podem avaliar como o jogo melhorou o aprendizado e o ensino do Scrum pela atividade.

Nos trabalhos [14], [17] e [35], focamos apenas em onde os estudantes verificaram um alto nível de concordância (5) com as questões Q9, Q10 e Q11 considerando o nível mais baixo (1) até (5). Com base nas respostas dos estudantes, é possível ver que nosso trabalho proposto apresenta resultados melhores ou iguais do que [14] e [17]. No trabalho [35], a avaliação dos estudantes apresenta uma motivação melhor para os Q9 e Q11, mas o número de estudantes é menor que o nosso trabalho proposto.

O trabalho [15] não apresentou resultados numéricos, mas mostra opiniões sobre a motivação e como os estudantes se sentem sobre a aplicação da atividade. Em [15] e no nosso

Tabela XI
COMPARAÇÃO COM TRABALHOS RELACIONADOS.

Trabalho relacionado	Número de estudantes	Resultado do trabalho relacionado	Resultado do nosso trabalho proposto
[14]	73	Resposta para concordo totalmente (5) por questão Q9 - 32% / Q10 - 40% / Q11 - 35%	Resposta para concordo totalmente (5) por questão Q9 - 78,7% / Q10 - 87,2% / Q11 - 73,3%
[15]	-	"Gostei de como algo tão divertido e desafiador terminou com uma lição que nunca esquecerei ... as estimativas iniciais estão sempre erradas ... faça o seu melhor, mas não quebre as regras!"	"Achei a dinâmica divertida, bem aplicada e organizada. Para mim, foi muito mais fácil entender os conceitos do Scrum" Opinião de estudante.
[16]	15	Níveis de 1 (discordo totalmente) para 5 (concordo totalmente) Q9 - 4,4 / Q10 - 4,5 / Q11 - 4,3	Níveis de 1 (discordo totalmente) para 5 (concordo totalmente) Q9 - 4,7 / Q10 - 4,8 / Q11 - 4,5
[17]	45	Respostas para concordo totalmente (5) por questão Q9 - 78% / Q10 - 28,9%	Respostas para concordo totalmente (5) por questão Q9 - 78% / Q10 - 87,2%
[35]	26	Respostas para concordo totalmente (5) por questão Q9 - 82% / Q10 - 80% / Q11 - 80%	Respostas para concordo totalmente (5) por questão Q9 - 78,7% / Q10 - 87,2% / Q11 - 73,3%

trabalho proposto, os estudantes referiram a atividade como divertida, desafiadora, bem aplicada, muito bem organizada e fácil de entender os conceitos do Scrum. No artigo original [15], os *feedbacks* de outros estudantes estão disponíveis.

Os estudantes participantes do trabalho de [16] indicaram sua motivação usando a escala Likert [40] com notas de pontuação (1 - pior até 5 melhor). Em nosso questionário, adaptamos essas escalas para discordar totalmente (1) até concordar totalmente (5). Nossos resultados apresentam melhores valores quando comparados com [16], conforme mostrado na Tabela XI.

VI. CONCLUSÕES

Neste artigo, apresentamos os resultados da aplicação de uma atividade acadêmica baseada na criação de uma cidade de blocos de Lego. A construção da cidade utiliza etapas, partes interessadas e elementos da metodologia ágil Scrum. O experimento foi realizado em três workshops diferentes, com a participação de estudantes. Inicialmente, os estudantes foram orientados nas diferentes etapas do Scrum e depois divididos em grupos para a construção de dezoito atividades do PB elaboradas pela PO. No final, os estudantes responderam a um questionário para entender qual era a percepção deles na realização do workshop.

Os resultados mostram que os estudantes consideraram a atividade muito eficaz para aprender Scrum. Ao analisar outras técnicas, os estudantes também indicaram que a praticidade de atividades como a jogo de blocos de Lego é uma boa alternativa.

Como trabalhos futuros, incentivamos a análise de atividades de aprendizagem, incluindo blocos de Lego com metodologias como XP, FDD, Lean ou Kanban Software. A maioria dos trabalhos relacionados apresentou sentimentos de motivação e encorajamento dos estudantes nas atividades propostas, no entanto, não fornece nenhum comparativo com outros processos tradicionais de aprendizagem, sendo necessário estudar com pesquisas complexas de muitas atividades em comparação com o Lego City Game. Ainda, sugere-se correlacionar a interação

do trabalho entre professores e estudantes durante o processo de aprendizagem, incluindo a motivação dos professores para usar essa atividade mais do que outras atividades tradicionais de aprendizagem. O papel de líder dos estudantes pode ser analisado nesta atividade, ou seja, aluno da equipe como mestre ou líder do Scrum, a fim de gerenciar a equipe, cumprir prazos, corrigir problemas de construção, mitigar a menor motivação da equipe ou propor soluções para desafios. Para avaliar tópicos específicos do Scrum, sugere-se aplicar testes a estudantes que usam os exames de Certificação Scrum e não apenas avaliar as percepções e intuição dos estudantes.

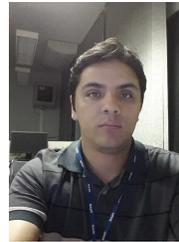
REFERÊNCIAS

- [1] L. Osterweil, "Software processes are software too," in *Engin. of Soft.* Springer, 2011, pp. 323–344.
- [2] Y. B. Leau, W. K. Loo, W. Y. Tham, and S. F. Tan, "Software development life cycle AGILE vs traditional approaches," in *Inter. Conf. on Inf. and Netwo. Techn.*, vol. 37, no. 1, 2012, pp. 162–167.
- [3] R. Fojtik, "Extreme Programming in development of specific software," *Proc. Comp. Scie.*, vol. 3, pp. 1464–1468, 2011.
- [4] B. Boehm and R. Turner, "Management challenges to implementing agile processes in traditional development organizations," *IEEE Soft.*, vol. 22, no. 5, pp. 30–39, 2005.
- [5] S. Nerur, R. Mahapatra, and G. Mangalaraj, "Challenges of migrating to agile methodologies," *Commun. of the ACM*, vol. 48, no. 5, pp. 72–78, 2005.
- [6] C. Scharff, S. Heng, and V. Kulkarni, "On the difficulties for students to adhere to Scrum on Global Software Development projects: Preliminary results," in *2nd Intern. Works. on Collab. Teach. of Glob. Dist. Softw. Devel.*, June 2012, pp. 25–29.
- [7] V. Vinekar, C. W. Slinkman, and S. Nerur, "Can agile and traditional systems development approaches coexist? An ambidextrous view," *Infors. Syst. Manag.*, vol. 23, no. 3, pp. 31–42, 2006.
- [8] V. Devedzic and S. R. Milenkovic, "Teaching Agile Software Development: A Case Study," *IEEE Trans. on Educ.*, vol. 54, no. 2, pp. 273–278, May 2011.
- [9] V. M. F. Fonseca and J. Gómez, "Applying Active Methodologies for Teaching Software Engineering in Computer Engineering," *IEEE Rev. Ibero. de Tecn. del Apend.*, vol. 12, no. 3, pp. 147–155, August 2017.
- [10] J. L. Duffany, "Application of Active Learning Techniques to the Teaching of Introductory Programming," *IEEE Rev. Ibero. de Tecn. del Apend.*, vol. 12, no. 1, pp. 62–69, February 2017.
- [11] R. M. Felder and L. K. Silverman, "Learning and teaching styles in engineering education," *Engin. Educ.*, vol. 78, no. 7, pp. 674–681, 1988.
- [12] M. J. Prince and R. M. Felder, "Inductive teaching and learning methods: Definitions, comparisons, and research bases," *Jour. of Engin. Educ.*, vol. 95, no. 2, pp. 123–138, 2006.

- [13] S. J. Armstrong and C. V. Fukami, *The SAGE handbook of management learning, education and development*. Sage, 2009.
- [14] C. G. Von Wangenheim, R. Savi, and A. F. Borgatto, "SCRUMIA - An educational game for teaching SCRUM in computing courses," *Jour. of Syst. and Soft.*, vol. 86, no. 10, pp. 2675–2687, 2013.
- [15] J. May, J. York, and D. Lending, "Teaching tip: Play ball: Bringing scrum into the classroom," *Jour. of Inform. Syst. Educ.*, vol. 27, no. 2, p. 87, 2016.
- [16] W. L. Lee, "SCRUM-X: An interactive and experiential learning platform for teaching scrum," in *Proc. of the 7th Intern. Confer. on Educ., Train. and Inform.* Elsevier, 2016.
- [17] G. Rodriguez, Á. Soria, and M. Campo, "Virtual Scrum: A teaching aid to introduce undergraduate software engineering students to Scrum," *Comp. Applic. in Engin. Educ.*, vol. 23, no. 1, pp. 147–156, 2015.
- [18] A. Godoy and E. F. Barbosa, "Game-Scrum: An approach to agile game development," in *Proc. of SBGames*, 2010, pp. 292–295.
- [19] C. Scharff and R. Verma, "Scrum to support mobile application development projects in a just-in-time learning context," in *Proc. of the Works. on Cooper. and Human Aspec. of Soft. Engin.* ACM, 2010, pp. 25–31.
- [20] C. A. G. da Silva, E. L. dos Santos, L. M. Angelo, R. V. de Moraes et al., "A utilização do SCRUM como recurso educacional no processo de aprendizagem em Engenharia de Software," *Intern. Jour. on Alive Engin. Educ.*, vol. 3, no. 2, pp. 87–102, 2016.
- [21] H. Takeuchi and I. Nonaka, "The new new product development game," *Harvard business review*, vol. 64, no. 1, pp. 137–146, 1986.
- [22] C. Larman and V. R. Basili, "Iterative and incremental developments. a brief history," *Computer*, vol. 36, no. 6, pp. 47–56, 2003.
- [23] K. Schwaber, "Scrum development process," in *Business object design and implementation*. Springer, 1997, pp. 117–134.
- [24] J. Highsmith and A. Cockburn, "Agile software development: The business of innovation," *Computer*, vol. 34, no. 9, pp. 120–127, 2001.
- [25] V. Mahnič and T. Hovelja, "On using planning poker for estimating user stories," *Jour. of Syst. and Soft.*, vol. 85, no. 9, pp. 2086–2095, 2012.
- [26] R. Tamrakar and M. Jørgensen, "Does the use of Fibonacci numbers in planning poker affect effort estimates?" in *16th Intern. Conf. on Eval. Asses. in Soft. Engin.*, May 2012, pp. 228–232.
- [27] V. Mahnic, "A capstone course on agile software development using Scrum," *IEEE Trans. on Educ.*, vol. 55, no. 1, pp. 99–106, 2011.
- [28] M. Paasivaara, J. Vanhanen, V. T. Heikkilä, C. Lassenius, J. Itkonen, and E. Laukkanen, "Do high and low performing student teams use scrum differently in capstone projects?" in *39th Intern. Conf. on Soft. Engin.: Soft. Engin. Educ. and Train. Track*. IEEE, 2017, pp. 146–149.
- [29] M. Golfarelli, S. Rizzi, and E. Turricchia, "Sprint planning optimization in agile data warehouse design," in *Inter. Conf. on Data Wareh. and Knowl. Discov.* Springer, 2012, pp. 30–41.
- [30] Z. Azham, I. Ghani, and N. Ithnin, "Security backlog in Scrum security practices," in *Malays. Conf. in Soft. Engin.* IEEE, 2011, pp. 414–417.
- [31] L. Rising and N. S. Janoff, "The Scrum software development process for small teams," *IEEE Software*, vol. 17, no. 4, pp. 26–32, 2000.
- [32] C. A. G. da Silva, E. L. dos Santos, and D. A. F. Pelacini, "Evaluation of Academic Experience in Learning Education over Simulators Softwares," *Inter. Jour. on Alive Engin. Educ.*, vol. 5, no. 2, pp. 23–40, 2018.
- [33] M. Kropp, A. Meier, M. Mateescu, and C. Zahn, "Teaching and learning agile collaboration," in *IEEE 27th Confer. on Soft. Engin. Educ. and Train.* IEEE, 2014, pp. 139–148.
- [34] J.-P. Steghöfer, H. Burden, H. Alahyari, and D. Haneberg, "No silver brick: Opportunities and limitations of teaching scrum with lego workshops," *Jour. of Syst. and Soft.*, vol. 131, pp. 230–247, 2017.
- [35] R. Castro et al., "AGILITY SCRUM—Um Jogo para Ensino da Metodologia SCRUM," in *25ª WEI-Workshop sobre Educação em Computação*, 2017.
- [36] J. Marty and T. Carron, "Observation of Collaborative Activities in a Game-Based Learning Platform," *IEEE Trans. on Learn. Techn.*, vol. 4, no. 1, pp. 98–110, Jan 2011.
- [37] K. Getchell, A. Miller, J. R. Nicoll, R. Sweetman, and C. Allison, "Games Methodologies and Immersive Environments for Virtual Fieldwork," *IEEE Trans. on Learn. Techn.*, vol. 3, no. 4, pp. 281–293, October 2010.
- [38] . Serrano-Laguna, J. Torrente, B. M. Iglesias, and B. Fernández-Manjón, "Building a Scalable Game Engine to Teach Computer Science Languages," *IEEE Rev. Ibero. de Tecn. del Aprend.*, vol. 10, no. 4, pp. 253–261, November 2015.
- [39] M. Paasivaara, V. Heikkilä, C. Lassenius, and T. Toivola, "Teaching students scrum using LEGO blocks," in *Proc. of the 36th Intern. Confer. on Soft. Engin.* ACM, 2014, pp. 382–391.
- [40] R. Likert, "A technique for the measurement of attitudes," *Arch. of Psych.*, 1932.



Douglas Augusto Barcelos Bica recebeu seu diploma de bacharel em química pela Universidade Federal do Paraná (UFPR) em 2016. Atualmente, ele é um estudante de graduação do curso tecnológico de Análise e Desenvolvimento de Sistemas da Universidade de Araucária (Unifacear) no Brasil. Suas principais áreas de interesse são desenvolvimento de software, metodologias ágeis, Scrum Framework e Educação em Computação.



Carlos Alexandre Gouvea da Silva (SM'16-M'18) natural de Antonina, PR é Engenheiro de Computação pela Pontifícia Universidade Católica do Paraná (PUCPR) em 2012, Mestre em Engenharia Elétrica em 2015 e atualmente doutorando também em Engenharia Elétrica pela Universidade Federal do Paraná (UFPR). Possui experiências em engenharia, teste e qualidade de softwares, onde também pôde lecionar em cursos superiores de engenharia. Suas principais áreas de interesse são telecomunicações, redes de computadores, transmissão e codificação de vídeo,

redes wireless e modelos de perdas, educação e qualidade de ensino em engenharias.