

SCRUM como Metodologia Ágil para o Desenvolvimento de Software Educativo: um relato de experiência

José Vinícius Vieira Lima, UECE, jvinicius.vieira@uece.br, 0000-0001-6161-9374

Resumo: Softwares educativos são ferramentas com grande potencial para apoiar e enriquecer o processo de ensino-aprendizagem entre as mais diversas disciplinas e níveis de ensino. Contudo, desenvolver um software educativo de qualidade que compreenda tanto os aspectos técnicos, quanto os pedagógicos, é uma tarefa desafiadora, tendo em vista que para isto é necessário seguir uma série de etapas para que o mesmo atinja inteiramente seu propósito. Assim, este trabalho apresenta um relato de experiência de alunos de graduação de Licenciatura em Computação ao desenvolver um projeto real utilizando como proposta pedagógica a Aprendizagem Baseada em Problemas e o uso da metodologia ágil SCRUM. Os resultados evidenciaram que a abordagem utilizada resultou em um conjunto de competências adquiridas pelos alunos da disciplina.

Palavras-chave: Software Educativo, Metodologia Ágil, SCRUM, Aprendizagem Baseada em Problemas.

SCRUM as an Agile Methodology for the Development of Educational Software: an experience report

Abstract: Educational software has great potential to support and enrich the teaching-learning process across the most diverse subjects and academic levels. However, developing quality educational software that encompasses both technical and pedagogical aspects is a challenging task, considering that for this, it is necessary to follow a series of steps so that it fully achieves its purpose. Thus, this work presents an experience report of undergraduate students in Computing when developing an actual project using Problem-Based Learning and the agile SCRUM methodology as a pedagogical proposal. The results showed that the approach used resulted in skills acquired by the discipline's students.

Keywords: Educational Software, Agile Methodology, SCRUM, Problem-Based Learning.

1. Introdução

Os softwares educativos são considerados ferramentas que possuem grande potencial para apoiar, enriquecer e/ou mediar o processo de ensino-aprendizagem nas mais diversas disciplinas e níveis de ensino (Almeida e Gomes, 2021). Com base nesse cenário, o uso de softwares educativos tem se tornado uma estratégia instrucional de imensurável importância. Uma vez que os mesmos são capazes de promover a aprendizagem por meio de experimentação, interpretação, visualização, indução, abstração, generalização e demonstração, além de auxiliar os docentes com um recurso contemporâneo em suas práticas educativas (Martins; Freire e Rebouças, 2022).

Assim, para que os softwares educativos possam atender os objetivos nos quais se destinam e, principalmente, sejam construídos através de uma sólida base pedagógica, se faz necessário a utilização de um processo de desenvolvimento que compreenda tanto os aspectos técnicos, quanto os pedagógicos (Lima *et al.*, 2019). Apesar disso, pesquisadores relatam que, devido às particularidades desses softwares, o processo de desenvolvimento tem enfrentado fatores importantes que influenciam diretamente a sua qualidade: tais como: (i) aumento constante de novos requisitos aos protótipos, o que exige um planejamento rigoroso; (ii) formação de uma equipe de desenvolvimento

multidisciplinar; (iii) processo de avaliação baseado em indicadores de qualidade; e (iv) a dimensão educacional (Medeiros, 2019; Bordin; Rodrigues e Casagrande, 2023).

Além dessas adversidades, a maioria dos softwares educativos têm sido desenvolvidos através de processos de desenvolvimento clássicos (modelos tradicionais). Essa prática ecoa na dificuldade desses projetos em lidar e acomodar mudanças; além da baixa flexibilidade; poucas iterações; falta de participação de atores externos (*e.g.*, profissionais na área da educação); rotatividade da equipe; escopo mal definido; e entregas a longo prazo (Gonçalves; Brancher e Bussmann, 2021). Como forma de reduzir esses obstáculos, as metodologias de desenvolvimento ágeis surgiram como sendo um conjunto de comportamentos, processos, técnicas e práticas que oferecem mais rapidez, eficiência e priorizam as melhorias contínuas. Assim, nos últimos anos, essas metodologias passaram a ser utilizadas e adaptadas às circunstâncias do contexto educacional (Medeiros, 2019).

Motivado por essas evidências, este estudo apresenta um relato de experiência prático do processo de desenvolvimento de software educativo ao utilizar a combinação do método ágil SCRUM juntamente com o apoio da metodologia ativa de ensino Aprendizagem Baseada em Problemas (PBL do inglês *Problem-Based Learning*). Assim, o objetivo desse trabalho é relatar a experiência de alunos de graduação de Licenciatura em Computação, no contexto da disciplina de Fundamentos e Análise de Software Educativo. Como principais contribuições, apresenta-se um conjunto de boas práticas para a produção de softwares educativos de qualidade, através da utilização de métodos de desenvolvimento requeridos pelo mercado de trabalho nos dias atuais; além de manifestar os benefícios que esses métodos resultam na formação pessoal e profissional dos alunos.

Além dessa Seção introdutória, este trabalho apresenta a Seção 2 que disserta sobre os conceitos que alicerçam a pesquisa. A Seção 3 evidencia a metodologia utilizada. Na Seção 4, o processo de desenvolvimento do software educativo é detalhado. A Seção 5 expõe a discussão dos resultados. Por fim, a Seção 6 expressa as considerações finais.

2. Fundamentação Teórica

2.1. Aprendizagem Baseada em Problemas

O PBL é um método de ensino-aprendizagem que possibilita ao aluno o desenvolvimento de habilidades técnicas, cognitivas, interpessoais e intrapessoais ao integrar teoria e prática, mediante um processo de resolução de problemas (Barros; Paiva e Hayashi, 2023). Assim, o PBL é considerado uma abordagem que promove uma formação inovadora, crítica e reflexiva, tornando o aprendizado mais significativo, onde conceitos abstratos são associados a experiências do mundo real (Lima *et al.*, 2020).

Deste modo, o PBL tem sido adotado em diversas áreas do conhecimento e em diferentes países. Para a educação na área de Computação, esta metodologia se mostra adequada por possibilitar meios de unir a formação inicial dos alunos à prática profissional. Incluindo a aquisição de competências profissionais e capacitando os mesmos a desenvolverem suas aptidões técnicas na prática para a qual estão sendo preparados. Por isso, essa abordagem foi utilizada no processo de condução da disciplina.

2.2. SCRUM

De acordo com Sutherland (2017), o SRCUM é um *framework* dentro do qual pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível. Devido sua particularidade, com ênfase no gerenciamento do desenvolvimento, o SCRUM é atualmente o método ágil mais utilizado nas empresas de desenvolvimento de software.

O SCRUM baseia-se nos princípios de transparência, inspeção e adaptação. Seu funcionamento ocorre através de equipes pequenas de, no máximo, nove pessoas; e iterações curtas. A equipe é organizada em três papéis principais: (i) Product Owner: representa os interesses de todos no projeto; (ii) Scrum Master: garante que todos sigam os valores e práticas do SCRUM, além de ser o responsável por remover os obstáculos do projeto; e (iii) Time de desenvolvimento: consiste de profissionais que realizam o trabalho de entregar uma versão usável que, potencialmente, incrementa o produto.

Para melhor gerenciamento, o desenvolvimento é dividido em intervalos de tempos que podem variar entre duas a quatro semanas, chamados de *Sprints*. Durante a execução das *Sprints*, diariamente o time faz uma reunião de 15 minutos para acompanhar o progresso do trabalho e agendar outras reuniões, caso seja necessário. Ao final da *Sprint*, realiza-se a reunião de revisão para que o Time apresente os resultados alcançados na iteração ao Product Owner. Neste momento, os requisitos são inspecionados e adaptações do projeto podem ser realizadas. Em seguida, o Scrum Master conduz a reunião de retrospectiva, a qual possui o objetivo de melhorar o processo de trabalho e/ou as práticas do time e/ou produto para a próxima execução da *Sprint*.

3. Estrutura Metodológica

A estrutura metodológica deste trabalho é baseada na utilização da metodologia ágil SCRUM e do PBL para o desenvolvimento de um software educativo. A Figura 1 ilustra todo o processo adotado e os principais atores envolvidos.

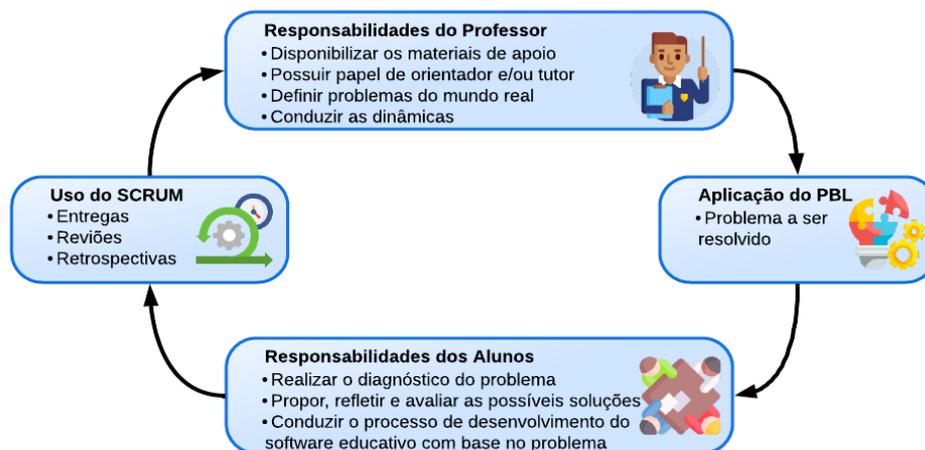


Figura 1. Processo de ensino utilizado na disciplina.

Esse processo foi realizado em um semestre, durante a condução da disciplina de Fundamentos e Análise de Software Educativo, do curso de Licenciatura em Computação, da Universidade Estadual do Ceará (UECE). A disciplina foi composta por seis alunos. Com isso, todos formaram uma única equipe e, através de um problema do cotidiano, um software educativo foi desenvolvido, testado e avaliado perante indicadores de qualidade.

O processo adotado para esta disciplina consistiu em estabelecer papéis, por meio de estudos, reflexões, discussões e aplicações de conceitos teóricos e práticos. Ou seja, após as aulas conceituais sobre as necessidades e aspectos que compreendem o desenvolvimento de softwares educativos, os alunos aplicaram os conceitos práticos mediante a utilização da abordagem PBL, a qual teve como base um problema real. Essa ação resultou no desenvolvimento de um software educativo.

Sendo assim, o contexto dessa disciplina foi simular uma fábrica de software

levando em consideração os aspectos técnicos e pedagógicos, uma vez que se trata de um software educativo. Deste modo, a equipe ficou responsável por desenvolver um projeto, passando por cada fase do processo de desenvolvimento. Para isso, utilizou-se a metodologia ágil SCRUM, dado que o mesmo contém marcos e papéis bem definidos, além de entregas contínuas, o que permitiu a melhor gestão do desenvolvimento.

Neste contexto, o propósito com a utilização dessa estratégia foi duplo: (i) construir um software educativo de qualidade através da aplicação de uma metodologia de desenvolvimento ágil; (ii) promover a colaboração, comunicação e estimular os alunos a desenvolver habilidades exigidas pelo mercado de trabalho, simulando o ambiente real.

4. Desenvolvimento do Software Educativo utilizando o SCRUM

4.1. Definição do Problema

Essa etapa buscou identificar o problema real a ser tratado. Para isso, utilizou-se como estratégia a realização de entrevistas semiestruturadas com 20 professores do Ensino Fundamental II, os quais eram pertencentes a uma escola pública da rede estadual, que está localizada na cidade de Mombaça, onde a UECE possui um Campus. A condução da entrevista seguiu os *guidelines* recomendados por Strandberg (2019).

Dessa maneira, informações pertinentes foram obtidas, analisadas e sintetizadas a fim de designar os desafios enfrentados pelos professores no ambiente educacional. Com isso, constatou-se a necessidade de desenvolver uma plataforma educacional direcionada aos alunos do Ensino Fundamental II. Ainda, conforme as evidências analisadas, se faz necessário a inclusão de livros e PDFs voltados às disciplinas de Português, Matemática, Ciências, Geografia e História. Além disso, a plataforma deverá possuir a possibilidade de converter esses materiais em áudio book, bem como dispor de Recursos Educacionais Digitais (*e.g.*, jogos educacionais, infográficos, vídeos, aplicativos, tutoriais e simulações) relacionados às disciplinas supracitadas.

Esse problema foi utilizado como alvo a ser explorado ao aplicar a abordagem PBL com a equipe. Consequentemente, a mesma realizou o diagnóstico do problema em questão através de rodadas de discussões e reflexões levando em consideração os conceitos abordados nas aulas. Durante essa etapa, o professor agiu como mediador e os alunos tiveram a autonomia nos processos de tomadas de decisões, os quais resultaram nas escolhas das tecnologias a serem utilizadas e o tipo de software a ser desenvolvido.

4.2. Definição dos Papéis do SCRUM

Baseando-se no problema mencionado na Subseção anterior, o processo de desenvolvimento do software educativo iniciou-se com a divisão da equipe respaldando-se nos papéis do SCRUM. A escolha dos papéis se deu devido a familiaridade dos alunos diante das aptidões e responsabilidades necessárias para executá-los. Neste sentido, a Tabela 1 apresenta a composição do time SCRUM e seus respectivos papéis.

Tabela 1. Composição do time SCRUM.

Responsável	Papel
Professor da disciplina	Product Owner
Professor da disciplina	Scrum Master
Aluno 1 e Aluno 2	Desenvolvedor
Aluno 3 e Aluno 4	Web Design
Aluno 5	Administrador de Banco de Dados
Aluno 6	Testador do Software

4.3. Definição do *Product Backlog* e Condução das *Sprints*

Após estabelecer os papéis do time, foi realizada a definição do *Product Backlog*. Esse processo ocorreu em dois momentos. De início, foi criada uma lista contendo todas as possíveis funcionalidades (requisitos) do software a ser desenvolvido. Participaram desse encontro o professor da disciplina representando o Product Owner e o Scrum Master, e também cinco professores da escola pública onde realizou-se a identificação do problema, os mesmos são considerados *stakeholders* (partes interessadas).

Posteriormente, rodadas de discussões foram realizadas pelo professor da disciplina como sendo o Product Owner justamente com o time SCRUM. O objetivo desses debates foi definir o nível de prioridade (*e.g.*, alta, média e baixa) dos requisitos que compõem o *Product Backlog*. Assim, os requisitos foram organizados de acordo com seus respectivos níveis de prioridades (do maior para o menor). É válido mencionar que, os requisitos que possuem o maior nível de prioridade foram desenvolvidos inicialmente. Além disso, o nível de prioridade de um requisito é atribuído mediante a importância do mesmo para os *stakeholders* (o que agrega maior valor diante de uma entrega mais rápida), com a possibilidade de ser re-priorizado ao longo do desenvolvimento. Conforme os requisitos foram sendo desenvolvidos a cada *Sprint*, novas funcionalidades poderiam ser descobertas, priorizadas e, conseqüentemente, incluídas no *Product Backlog*.

Assim, ao final desse processo, 15 requisitos compuseram o *Product Backlog*, os quais foram desenvolvidos durante a execução de cinco *Sprints* (com duração de 10 dias cada). Esses requisitos foram distribuídos em uma planilha eletrônica *on-line* de forma a promover o melhor gerenciamento pelo time SCRUM. A Figura 2 ilustra o modelo desta planilha em um determinado estágio do desenvolvimento do software educativo, além de apresentar todos os requisitos definidos, seus níveis de prioridade, bem como a complexidade e o esforço despendido pela equipe para desenvolvê-los. Também foi utilizado o quadro do Trello* como ferramenta visual na qual possibilitou ao time SCRUM a definição do fluxo de trabalho entre os integrantes e o monitoramento das tarefas. Essa prática buscou estimular a auto-organização do time, além de servir para que os mesmos pudessem compartilhar seus progressos e dificuldades.

PRODUCT BACKLOG					
ID	Funcionalidade	Nível de Prioridade	Story Points (Complexidade + Esforço)	Sprint a ser implementada	Status
R01	Realizar Login	Alta	3 Story Points	SPRINT 1	Concluída
R02	Acessar Material	Alta	5 Story Points	SPRINT 3	Concluída
R03	Acessar Jogo	Alta	8 Story Points	SPRINT 3	Concluída
R04	Disponibilizar Áudio	Alta	8 Story Points	SPRINT 4	Em andamento
R05	Acessar Vídeo Aula	Alta	8 Story Points	SPRINT 3	Concluída
R06	Atualizar Interface	Alta	13 Story Points	SPRINT 1	Em andamento
R07	Criar Painel Administrativo	Alta	21 Story Points	SPRINT 2	Em teste
R08	Criar Banco de Dados	Alta	13 Story Points	SPRINT 1	Concluída
R09	Disponibilizar Acesso	Alta	5 Story Points	SPRINT 5	Não iniciada
R10	Barra de Progresso	Média	5 Story Points	SPRINT 4	Em andamento
R11	Disponibilizar Responsividade	Média	5 Story Points	SPRINT 1	Concluída
R12	Mecanismo de Busca	Média	13 Story Points	SPRINT 5	Não iniciada
R13	Página De Suporte	Média	8 Story Points	SPRINT 3	Concluída
R14	Ranking Geral	Baixa	13 Story Points	SPRINT 5	Não iniciada
R15	Recurso de Dicas	Baixa	5 Story Points	SPRINT 4	Em andamento

Figura 2. Modelo de planilha utilizada para o gerenciamento do *Product Backlog*.

4.4. Reuniões de Revisão e Reuniões de Retrospectiva

Ao término de cada *Sprint*, realizou-se presencialmente a reunião de revisão, que representou a entrega das funcionalidades desenvolvidas durante a mesma. Desta forma, o professor da disciplina representando o Product Owner, efetuou a inspeção dos resultados

*<https://trello.com/>

das *Sprints*, de modo a aceitá-los ou rejeitá-los, bem como atribuir possíveis adaptações para serem incluídas ao *Product Backlog*. Ademais, também foi realizado as reuniões de retrospectivas, onde o time SCRUM refletia suas práticas, criava uma base de lições aprendidas e investigava o que foi feito de forma correta e o que poderia ser melhorado para a execução das *Sprints* futuras. Um exemplo de uma das dinâmicas empregadas durante uma das reuniões de retrospectiva pode ser observado na Figura 3. Na mesma, cada membro do time SCRUM utilizava um ou mais *post-its* e, ao realizar um processo de autoavaliação e autocrítica sobre o seu desempenho e de todo o time ao longo da condução da *Sprint*, explanava e discutia com os demais integrantes.

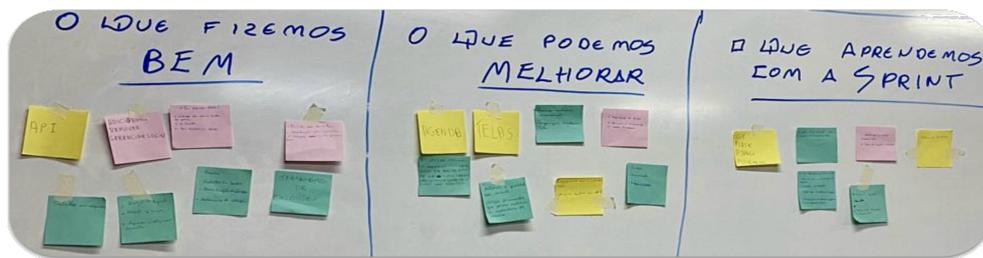


Figura 3. Exemplo de uma das estratégias utilizadas durante a reunião de retrospectiva do projeto.

4.5. Entrega e Validação

Através da condução das cinco *Sprints*, foi possível concluir o desenvolvimento da plataforma educacional diante dos anseios evidenciados pelos professores da escola pública, onde o problema real foi constatado (mais detalhes podem ser observados na Subseção 4.1). Assim, foi desenvolvido pelo time SCRUM uma plataforma educacional do tipo *WEB*. O EduAção é uma plataforma de aprendizagem *on-line* que disponibiliza materiais didáticos gratuitos, como vídeos, jogos e livros. Esses recursos educacionais estão relacionados às disciplinas do Ensino Fundamental II conforme as recomendações da Base Nacional Comum Curricular. Para maiores detalhes, a plataforma pode ser acessada através deste *link*[†]. A Figura 4 exibe a página inicial da plataforma, a qual conta com um menu na parte superior que dá acesso às demais páginas.

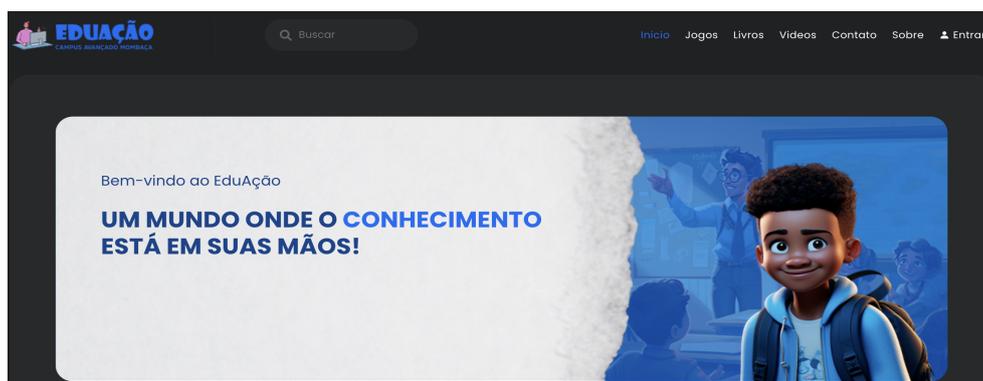


Figura 4. Página inicial do software educativo desenvolvido.

É válido mencionar que o software foi avaliado ao seguir as recomendações propostas por Almeida e Gomes (2021), além de adaptar algumas questões dos estudos realizados por Lima, Alencar e Santos (2023) e Barbosa e Petri (2023). Deste modo, dois ciclos de avaliações foram conduzidos com os *stakeholders* do projeto. Os resultados da

[†]<https://bit.ly/EduAcao>

avaliação estão sendo sintetizados e sumarizados na construção de um outro artigo. Outro ponto que merece destacar é que a plataforma encontra-se atualmente em atualização com relação a inserção de novos recursos educacionais digitais.

5. Resultados

Esta Seção apresenta a avaliação da experiência do processo de desenvolvimento do software educativo ao utilizar o SCRUM de acordo com as perspectivas dos alunos.

O processo de avaliação ocorreu através da aplicação de um questionário[‡] que avaliou um conjunto de aptidões com os alunos da disciplina. O mesmo foi composto por questões objetivas e questões discursivas. No primeiro grupo, as perguntas incluíram alternativas de múltipla escolha baseadas em *Likert Scales*, contendo uma escala ordinal de cinco graus de concordância (de 1 a 5). Nas questões discursivas, as respostas originaram-se mediante um espaço disponível para livre resposta textual. Utilizou-se a estatística descritiva para realizar a análise, interpretação e representação dos dados.

5.1. Análise, Interpretação e Discussão dos Resultados

Inicialmente, buscou-se verificar se os alunos já haviam cursado, em semestres anteriores, alguma disciplina relacionada a desenvolvimento de software. Neste sentido, observou-se que 100% dos mesmos tiveram esse tipo de experiência. Logo após, os alunos foram indagados acerca do nível de conhecimento sobre a metodologia ágil SCRUM. Assim, todos (100%) afirmaram compreendê-la apenas apenas de forma teórica. Ou seja, por mais que os alunos tivessem cursado disciplinas de desenvolvimento de software, nenhum dispôs de experiências práticas utilizando o SCRUM.

Por se tratar da primeira oportunidade prática dos alunos com o SCRUM, decidiu-se diagnosticar as potenciais habilidades e competências desenvolvidas e/ou aperfeiçoadas pelos mesmos. A Figura 5 apresenta esse panorama.

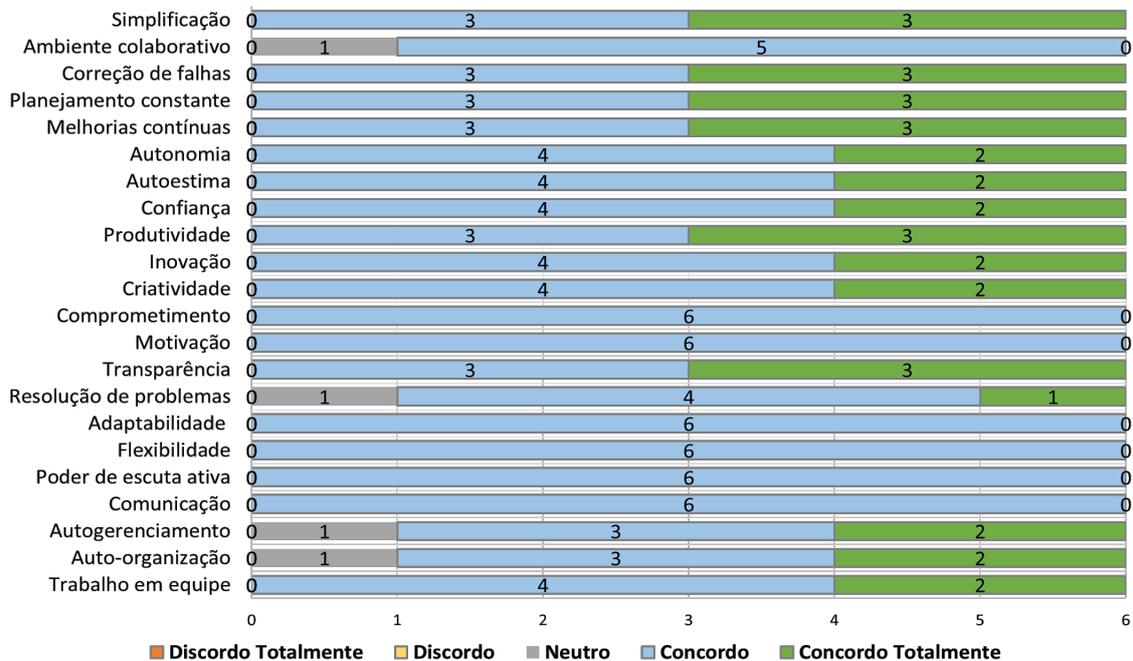


Figura 5. Respostas obtidas entre as competências e habilidades avaliadas nos alunos.

[‡]<https://bit.ly/DiagnosticoSCRUM>

Baseando-se nos dados apresentados na Figura 5, nota-se que 19 das 22 competências foram avaliadas por todos os alunos contendo o grau de satisfação “concordo” e/ou “concordo fortemente”. Neste contexto, estes dados salientam a importância do uso do SCRUM como metodologia ágil para o desenvolvimento de software educativo perante a aquisição de características que são essenciais para um projeto de sucesso (e.g., ambiente colaborativo, autogerenciamento, criatividade, adaptabilidade, flexibilidade e transparência).

Com a auto-organização do time SCRUM, os integrantes adquiriram autonomia para definir como iriam ser desenvolvidas cada atividade, além de priorizar e deliberar as entregas. Outro fator importante está relacionado a comunicação, uma vez que todos os alunos concordaram que aperfeiçoaram essa habilidade ao fazer uso do SCRUM. Acredita-se que isso se deu mediante o tamanho do time, dado que o mesmo foi formado por seis membros, o que garantiu uma comunicação assertiva e eficiente, resultando em uma maior integração entre os participantes e, principalmente, no poder de escuta ativa.

Além disso, todos os integrantes consideram-se mais motivados com a utilização do SCRUM como metodologia para o desenvolvimento do software educativo. De acordo com o Aluno 3, *“o alto nível de motivação é resultado do ambiente propício que o SCRUM possibilita, dado que o mesmo promove mais agilidade, estimula a inovação, além da satisfação em realizar tarefas com as quais existe mais afinidade”*. Considera-se que essa afirmação vai ao encontro das competências associadas à confiança e à produtividade. Isso porque, *“a confiança na execução das Sprints e do projeto como um todo ajudou a amenizar os possíveis problemas dentro da própria equipe.”*– destacou o Aluno 1. Adicionalmente, *“essas particularidades se tornaram motivos para o time se manter motivado, com senso de responsabilidade coletiva e comprometimento.”*– apontou o Aluno 4. Como consequência, *“eleva-se a produtividade do time por meio da entrega dos requisitos de alto valor para os stakeholders e, ao mesmo tempo, busca-se o melhor aproveitamento dos recursos (e.g., tempo, mão de obra e custo).”*– expôs o Aluno 6.

Através das reuniões de revisões que o SCRUM estabelece, a habilidade de melhorias contínuas também foi desenvolvida pelos alunos. Para o Aluno 5, *“quando o professor, no papel de Product Owner, realizava a inspeção dos resultados das Sprints, os feedbacks atribuídos originaram um ciclo virtuoso de melhorias contínuas. Com essa prática, o time se sentia mais encorajado e estimulado a aperfeiçoar as entregas a cada nova Sprint, buscando atender, constantemente, a qualidade esperada.”*. Intrinsecamente, as reuniões de retrospectivas possibilitaram ao time SCRUM competências de resolução de problemas e planejamento constante para correção de falhas. Isso ocorreu por intermédio da cultura de autoavaliação que foi implantada, o que provocou a identificação de possíveis falhas e a criação de lições aprendidas tendo como objetivo planejar e realizar ajustes à face da realidade e das necessidades do projeto.

De modo geral, constata-se que o SCRUM favoreceu a capacidade dos integrantes do time em trabalhar em equipe. Para o Aluno 3, *“a atuação coletiva foi fundamental para o sucesso do projeto dentro do prazo estipulado”*. Em paralelo, *“esse ambiente colaborativo refletiu no aumento do desempenho do time, o que permitiu a simplificação no desenvolvimento de um projeto real durante a disciplina.”*– ressaltou o Aluno 2.

Em resumo, as habilidades e competências avaliadas obtiveram média geral de 4,25 graus. Esses números são considerados significativos, em razão do alcance máximo ser do tipo grau 5 (concordo fortemente), o que equivale a 85% de acurácia. Assim, a análise desses resultados permite, dentro dos limites estabelecidos, concluir que os alunos da disciplina, ao utilizar o SCRUM como metodologia ágil para o desenvolvimento do software educativo, adquiriram aptidões que são recomendadas pelo mercado de trabalho.

As questões discursivas proporcionaram a análise dos aspectos relativos às dificuldades enfrentadas na adoção do SCRUM. A análise qualitativa resultou em duas categorias: (i) barreira cultural; e (ii) mudanças constantes nos requisitos. A primeira adversidade está relacionada ao processo de adaptação dos membros do time SCRUM, uma vez que todos nunca trabalharam com a metodologia de forma prática. Acredita-se que isso ocorreu mediante a familiaridade dos integrantes com as metodologias tradicionais de desenvolvimento de software. Neste cenário, essa dificuldade também harmoniza-se com a segunda categoria. Ou seja, por se tratar da primeira experiência prática dos alunos com o SCRUM, os mesmos desconheciam que essa metodologia costuma ser altamente configurada para gerenciar as mudanças no escopo de forma eficaz e eficiente, dado que mudanças são inevitáveis no desenvolvimento de software educativo.

5.2. Limitações e Ameaças à Validade

Embora a pesquisa tenha sido realizado cuidadosamente, existem algumas limitações e ameaças que devem ser consideradas, sendo elas: (i) o quantitativo baixo de alunos matriculados na disciplina pode ter interferido nos resultados (entretanto, é indicado, pelo Guia Scrum[§] que o time seja composto de 5 a 9 integrantes); (ii) o processo de avaliação com os alunos foi efetuado com apenas um método proposto; (iii) a definição do questionário pode ter limitado a pesquisa ou não ter abordado todos os conteúdos relevantes sobre o tema; e (iv) a pesquisa foi realizada em apenas um semestre letivo (de forma a mitigar essa ameaça, todas as atividades planejadas levando em consideração as características da metodologia SCRUM foram devidamente executadas no prazo).

6. Considerações Finais

O uso de softwares educativos no auxílio do processo de ensino-aprendizagem é considerada uma estratégia fundamental nos dias atuais. No entanto, a maioria dos softwares educativos não são desenvolvidos levando em consideração as suas reais necessidades, bem como os aspectos técnicos, de modo que possam garantir a qualidade pedagógica de forma efetiva. Ainda neste cenário, os softwares educativos têm sido, predominantemente desenvolvidos, através da adoção de modelos de desenvolvimento tradicionais, o que reflete em uma série de dificuldades enfrentadas.

Como forma de apoiar o processo de desenvolvimento de softwares educativos, esta pesquisa apresenta um relato de experiência de alunos de graduação de Licenciatura em Computação, na disciplina de Fundamentos e Análise de Software Educativo. Estes foram incumbidos da realização de um projeto de desenvolvimento de software educativo. Para isso, utilizou-se a combinação da metodologia ágil SCRUM e da Aprendizagem Baseada em Problemas, a qual teve como base um problema real da comunidade.

Ao final do processo, um software educativo foi desenvolvido, testado e avaliado de acordo com indicadores de qualidade. Através da avaliação realizada com os alunos, observou-se que a abordagem se mostrou bastante favorável, uma vez que os conceitos teóricos foram aliados à prática. Sendo assim, este estudo contribui para a comunidade acadêmica ao apresentar boas práticas para a produção de softwares educativos de qualidade através da utilização do SCRUM. O mesmo também colabora para a formação pessoal e profissional dos alunos, dado que os resultados evidenciaram que abordagem utilizada originou-se na aquisição de um conjunto de competências e habilidades.

Como propostas futuras, pretende-se melhorar a abordagem utilizada com base nas dificuldades apontadas pelos alunos durante o processo de avaliação, de modo a aplicar em novas turmas e disciplinas relacionadas ao desenvolvimento de software.

[§]<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-PortugueseBR-3.0.pdf>

Também é esperado realizar novas avaliações empregando outros tipos de métodos. Com relação a esta perspectiva, busca-se efetuar uma análise minuciosa sobre as percepções de aprendizagem. Para isso, é desejado utilizar grupos de controle como forma de determinar o nível de aprendizado e desenvolvimento de competências nos alunos. Além disso, também é planejado realizar oficinas e treinamentos relacionados a aplicação do SCRUM. Com isso, espera-se possibilitar a disseminação de boas práticas para o desenvolvimento de softwares de qualidade, além de promover a capacitação sobre o método ágil mais requisitado pelo mercado de trabalho atualmente.

Referências

- Almeida, A.; Gomes, L. d. Q. L. Avaliação de softwares educacionais através de indicadores de qualidade. In: SBC. **Anais do XXXII Simpósio Brasileiro de Informática na Educação**. [S.l.], 2021. p. 249–258.
- Barbosa, E. R.; Petri, G. Desenvolvimento e avaliação de um jogo educacional híbrido para apoiar o ensino de gerência de projetos de software. **Revista Novas Tecnologias na Educação**, v. 21, n. 1, p. 200–209, jul. 2023. Disponível em: <https://seer.ufrgs.br/index.php/renote/article/view/134347>.
- Barros, V. A. M. D.; Paiva, H. M.; Hayashi, V. T. Using pbl and agile to teach artificial intelligence to undergraduate computing students. **IEEE Access**, IEEE, 2023.
- Bordin, A.; Rodrigues, L.; Casagrande, T. Ensino, pesquisa e extensão no ensino de engenharia de software: Um relato de experiência. In: **Anais do XXXI Workshop sobre Educação em Computação**. Porto Alegre, RS, Brasil: SBC, 2023. p. 30–40. ISSN 2595-6175. Disponível em: <https://sol.sbc.org.br/index.php/wei/article/view/24888>.
- Gonçalves, V. M.; Brancher, J. D.; Bussmann, A. J. Um mapeamento sistemático sobre recursos educacionais digitais na educação básica. In: SBC. **Anais do XXXII Simpósio Brasileiro de Informática na Educação**. [S.l.], 2021. p. 524–537.
- Lima, J. *et al.* As metodologias ativas e o ensino em engenharia de software: uma revisão sistemática da literatura. In: **Anais do XXV Workshop de Informática na Escola**. Porto Alegre, RS, Brasil: SBC, 2019. p. 1014–1023. Disponível em: <https://sol.sbc.org.br/index.php/wie/article/view/13251>.
- Lima, J. V.; Silva, C. D.; Alencar, F. R. de; Santos, W. Metodologias ativas como forma de reduzir os desafios do ensino em engenharia de software: diagnóstico de um survey. In: **Anais do XXXI Simpósio Brasileiro de Informática na Educação**. Porto Alegre, RS, Brasil: SBC, 2020. p. 172–181. ISSN 0000-0000.
- Lima, J. V. V.; Alencar, F.; Santos, W. B. Selection of active methodologies for software engineering teaching: a supporting guide. In: IEEE. **2023 18th Iberian Conference on Information Systems and Technologies (CISTI)**. [S.l.], 2023.
- Martins, A.; Freire, E.; Rebouças, A. D. Softwares educativos para apoiar a alfabetização: Um mapeamento sistemático. **Revista Novas Tecnologias na Educação**, v. 20, n. 1, p. 368–378, 2022.
- Medeiros, A. Ypeduc: Uma adaptação de metodologia ágil para o desenvolvimento de software educativo. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. [S.l.: s.n.], 2019. v. 30, n. 1, p. 379.
- Strandberg, P. E. Ethical interviews in software engineering. In: IEEE. **2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)**. [S.l.], 2019. p. 1–11.
- Sutherland, J. **Scrum: a arte de fazer o dobro do trabalho na metade do tempo**. [S.l.]: Leya, 2017.