

Student-centered approaches and industry training practices for Software Process and Quality: An ah-hoc literature review

ABSTRACT: This study presents student-centered approaches to teaching and learning Software Process and Quality. The themes are relevant, as the software development market increasingly demands continuous improvement in the delivery of high quality software products. The ways in which they become possible is through adjustments in the process, which is being followed, and ascertaining the quality with which it is being delivered. The methodology adopted was a Literature Review, with due care to mitigate possible threats to the study's validity. A total of 29 studies were selected, which were evaluated and the teaching proposals were extracted and the topics covered within the contents. Therefore, through this study it is possible to identify relevant topics for research in teaching of Process and Software Quality.

Keywords: Software Process; Software quality; Student-Centered Approaches; Ad-hoc Literature Review.

Abordagens centradas no aluno e práticas de capacitação da indústria para Processo e Qualidade de Software: Uma revisão *ad-hoc* da literatura

RESUMO: Este estudo apresenta abordagens centradas no aluno sobre o ensino e aprendizagem de Processo e Qualidade de Software. Os temas são relevantes, visto que o mercado de desenvolvimento de software solicita cada vez mais o aprimoramento contínuo na entrega de produtos de software de alta qualidade. As formas pelas quais eles se tornam possíveis é através de ajustes no processo, que está sendo seguido, e averiguando a qualidade com que está sendo entregue. A metodologia adotada foi a Revisão da Literatura, com os devidos cuidados para mitigar as possíveis ameaças à validade do estudo. Foram selecionados um total de 29 estudos, os quais foram avaliados e foi realizada a extração das propostas de ensino e os tópicos abordados dentro dos conteúdos. Portanto, a partir desse estudo é possível identificar tópicos relevantes para a pesquisa em ensino de Processo e Qualidade de Software.

Palavras-chave: Processo de Software; Qualidade de Software; Abordagens Centradas no Aluno; Revisão Ad-hoc da Literatura.

Agradecimentos: Este trabalho pertence ao projeto SPIDER/UFPA (<http://www.spider.ufpa.br>).

1. INTRODUÇÃO

A preocupação com a produção de um software de qualidade não se restringe apenas a atingir os objetivos (ou requisitos) esperados pelos usuários, mas também em obter um ciclo de vida de produção de software que apresente as características desejáveis em qualquer processo de desenvolvimento de software, como boa manutenibilidade, alta reusabilidade e baixo acoplamento. Em relação à importância do software, nunca tantas pessoas dependeram tanto de sistemas de informação, pois quase tudo depende de software (MEIRA, 2015). Quanto à formação de profissionais, tipicamente, esses se formam em cursos de graduação na área de Tecnologia da Informação a fim de se preparar para atuar na indústria de software (NUNES, REIS e REIS, 2010).

O mercado de trabalho, no entanto, identifica falhas nas competências dos profissionais formados, pois os cursos de graduação não ensinam aos estudantes as competências necessárias para que eles possam começar a executar o seu trabalho com eficiência (WANGENHEIM e SILVA, 2009; MORENO *et al.*, 2012; MEIRA, 2015). Dessa forma, as empresas de software têm que complementar os conhecimentos dos recém-formados com treinamentos e prover habilidades relacionadas ao processo de desenvolvimento de software (BESSA, CUNHA e FURTADO, 2012).

O eixo de formação em Qualidade de Software (SBC, 2017) têm como competência geral “produzir software de alta qualidade que estejam em conformidade com seus requisitos e satisfaça as necessidades do usuário”. Portanto, pretende-se com o profissional formado uma pessoa que consiga gerar um software com alta qualidade. Mas, para isso o modelo também descreve que “A obtenção da qualidade de software envolve modelos e técnicas de qualidade de produto e de processo de software” (SBC, 2017).

No que tange o Processo de Software, existe também um eixo formado dentro do que prevê o documento base da Sociedade Brasileira da Computação (SBC) (SBC, 2017), o qual têm como competência geral “entender, aplicar, criar e melhorar processos envolvidos no desenvolvimento de software, tais como: requisitos, projeto, construção, teste, configuração e qualidade”. Portanto, esse perfil atua de uma forma gerencial para aprimorar a produção de software da empresa.

Revisar a literatura sempre é um esforço para ser feito com o intuito de descobrir como vem sendo trabalhado os assuntos e as tendências do momento. Sobre revisão da literatura, segundo o relatório técnico de MAFRA e TRAVASSOS (2006) sobre pesquisa em engenharia de software baseada em evidências, existem dois tipos, são elas: revisão sistemática e revisão *ad-hoc* da literatura.

No confronto dos dados sobre o mercado de trabalho e o que é relatado pela formação dos cursos é possível identificar lacunas de como o profissional está indo ao mercado de trabalho. Portanto, é necessário e interessante para se verificar e identificar como vem sendo trabalhado o processo de ensino e aprendizagem de processo e qualidade de software, em especial os que tem abordagens focadas no aluno. Uma vez que a partir dessas abordagens temos uma maior participação e engajamento do aluno no desenvolvimento das atividades.

Neste contexto, este trabalho objetiva identificar na literatura as principais abordagens adotadas para o ensino e aprendizagem das disciplinas qualidade de software e processo de software, que sejam centradas no aluno e com técnicas de capacitação da indústria, nos cursos de Tecnologia da Informação. Como metodologia de pesquisa, uma Revisão *Ad-hoc* da Literatura (MAFRA e TRAVASSOS, 2006) foi conduzida, portanto este estudo está inserido no contexto de Engenharia de Software Baseado em Evidências.

Esta Revisão identificou 29 estudos, os quais foram avaliados por técnica descrita na seção 4.

Além desta seção introdutória, o artigo está estruturado da seguinte forma: a Seção 2 apresenta a fundamentação teórica; a Seção 3 tece comentários sobre trabalhos relacionados a essa pesquisa; a Seção 4 apresenta o planejamento e a condução do método; na Seção 5 são apresentados os resultados dessa revisão sistemática, respondendo as questões de pesquisa; na Seção 6 relata-se as ameaças a essa revisão sistemática; e a seção 7 apresenta as conclusões com as limitações desse estudo e os trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

A seção de fundamentação teórico apresenta alguns conceitos para o melhor entendimento desse trabalho, tais como: ensino de qualidade de software e processo de software.

2.1. Ensino de Qualidade de Software

Segundo o *Software Engineering Body of Knowledge* (SWEBOK) (BOURQUE e DUPUIS, 2004) o conceito atribuído à Qualidade de Software está sobrecarregado, pois durante anos foram construindo-se vários conceitos, dentre eles: (i) Características desejáveis de produtos de software; (ii) Conformidade com os requisitos; (iii) Qualidade orientada ao mercado, onde o cliente é quem decide ao final. Porém, o mais recente encunhado pela norma ISO/IEC 25010 (ISO/IEC, 2011) é o “capacidade do produto de software de satisfazer necessidades declaradas e implícitas sob condições especificadas”.

Como é percebido pelas definições sobre Qualidade de Software e pela reafirmação dos autores no livro SWEBOK (BOURQUE e DUPUIS, 2004), o assunto é relevante tanto para a academia quanto para a indústria de desenvolvimento de software, uma vez que permite durante o ciclo de desenvolvimento de software garantir a qualidade no processo e no produto desenvolvido.

Dentro da realidade brasileira no documento para a formação dos cursos superiores em Tecnologia da Informação (SBC, 2017) o tema qualidade de software é tratado como um subtópico da disciplina engenharia de software. Logo, é visível que não se tem tempo hábil para as discussões tão aprofundadas como o assunto pede, assim como a parte prática. Portanto, é interessante a busca dentro da literatura sobre como é ensinada a qualidade de software dentro da academia.

2.2. Ensino de Processo de Software

Ainda segundo o livro SWEBOK (BOURQUE e DUPUIS, 2004), processo de software é um conjunto de atividades e tarefas relacionadas que transformam entradas em um produto de saída. Nessa descrição podem ser incluídos critérios de entrada e saída, assim como realizar a decomposição de atividades em tarefas, que são as menores unidades de trabalho sujeitas à supervisão da gerência (BOURQUE e DUPUIS, 2004). É importante e necessário que todas as condições estabelecidas sejam atendidas antes que um processo seja concluído com êxito, incluindo os critérios de aceitação para o produto ou produtos de trabalho de saída.

Os autores BOURQUE e DUPUIS (2004) identificam ainda que o processo de software pode incluir subprocessos, Além do que relatam sobre os vários motivos para se o descrever, dentre eles: facilitar a compreensão, auxiliar o gerenciamento de projetos de software, apoiar a melhoria de processos e fornecer uma base para o suporte automatizado da execução do processo. Vale ressaltar que os processos de software devem ser adaptados conforme o projeto e a cada contexto organizacional.

3. TRABALHOS RELACIONADOS

Na literatura foi possível identificar dois trabalhos relacionados à pesquisa em tela, pela justificativa de desenvolvimento, ou seja, o mapeamento sobre as práticas de ensino, são eles: um *survey* e um mapeamento da literatura.

O *survey* proposto por FERREIRA *et al.* (2018) destaca a pesquisa de opinião sobre o ensino de Engenharia de Software no Brasil. Dentro da metodologia proposta pelo trabalho tem-se o *survey* para planejamento e coleta, e como análise temos *Grounded Theory*. A contribuição dos autores circunda sobre os tópicos emergentes em Engenharia de Software no Brasil e as dificuldades no processo de Ensino e Aprendizagem.

Já o mapeamento da literatura proposto por CUNHA *et al.* (2008) relata o ensino e a aprendizagem de Engenharia de Software no Brasil. Os autores adotaram um protocolo de revisão, além do uso de metodologia auxiliar, o *Snowballing*, para a coleta de estudos da literatura. Um ponto fraco baseia-se no fato do mapeamento ser realizado apenas em conferências brasileiras, não tendo a prospecção internacional. As principais contribuições do trabalho foram: um mapa de publicações sobre ensino de Engenharia de Software no Brasil; e os principais conteúdos e as metodologias adotadas.

Como é perceptível dentro da literatura, existem várias postagens sobre a investigação sobre o ensino de Engenharia de Software, porém ainda não se tem sobre o ensino de processo de software com abrangência nacional e internacional.

4. PLANEJAMENTO DA REVISÃO

A revisão da literatura é o meio pelo qual o pesquisador pode identificar o conhecimento científico existente em uma determinada área, de forma a planejar sua pesquisa, evitando a duplicação e a repetição de erros do passado (MAFRA e TRAVASSOS, 2006).

O formato escolhido de revisão é o *Ad-hoc*, pois permite que a mesma seja realizada de forma livre, porém com critérios estabelecidos, sendo eles: (i) conteúdo, ensino e processo e qualidade de software; (ii) tempo, 5 anos de 2015 até 2019; (iii), local de busca, conferências qualificadas para ciência da computação, que sejam da área de ensino, educação e ciência da computação.

4.1. Questão de pesquisa

A questão de pesquisa é identificada como uma das formas de nortear o trabalho de revisão para identificar na literatura se já existem abordagens descritas, ou os possíveis *gaps* para se realizar a pesquisa. Portanto, para esse trabalho temos a seguinte questão de pesquisa:

- (QP01) Quais as principais abordagens de apoio ao processo de ensino e aprendizagem de processo ou qualidade de software no contexto dos cursos de graduação e pós graduação em Tecnologia da Informação, que sejam centradas no aluno e com práticas de capacitação da indústria?

Sobre abordagens entende-se: metodologia, método, técnica, ferramenta, gamificação, jogo, aplicativo, diversas formas que possam ser utilizadas no processo de ensino. Por cursos de graduação e pós-graduação em Tecnologia da Informação, entende-se, mas não se limita, aos seguintes: Ciência da Computação, Sistemas de Informação, Engenharia de Software, Engenharia da Computação, Licenciatura em Computação, além dos cursos tecnológicos descritos no curriculum (SBC, 2017).

4.2. Identificação da pesquisa (fontes de busca)

A pesquisa foi realizada de forma manual nas principais conferências sobre Engenharia de Software, Ensino e Educação, as quais sejam qualificadas para Ciência da computação. A escolha pela abrangência geral de Engenharia de Software é porque, como indica o

curriculum da ACM/IEEE (2013) e o documento base para os cursos de Tecnologia da informação da SBC (2017), é a área geral sobre assuntos referentes aos conteúdos sobre Processo e Qualidade de Software. Durante o processo de coleta dos estudos primários foi adotado o acesso via Universidade Federal do Pará, através do portal Periódico CAPES, o qual permite acesso aos artigos publicados e indexados sobre as conferências. As conferências selecionadas, bem como seu Qualis para ciência da computação, estão disponíveis no Quadro 1.

Quadro 1 – Conferências utilizadas na Revisão *Ad-hoc* da Literatura e seus Qualis.

Sigla	Nome	Qualis
ICSE	International Conference on Software Engineering	A1
CSEE&T	Conference on Software Engineering Education and Training	B1
ITICSE	Conference on Innovation and Technology in Computer Science Education	A2
CSEDU	International Conference on Computer Supported Education	B1
SIGCSE	ACM Technical Symposium on Computer Science Education	A1
ICSOFT	International Conference on Software and Data Technologies	B1
SBIE	Simpósio Brasileiro de Informática na Educação	B1
SBQS	Simpósio Brasileiro de Qualidade de Software	B3
SBES	Brazilian Symposium on Software Engineering	B2
FIE	IEEE Frontiers in Education Conference	B1
WEI	Workshop sobre Educação em Computação	B3

Fonte: Elaboração própria (2019).

4.3. Estratégia de seleção dos estudos

No decorrer da pesquisa cada um dos anais das conferências foi consultado desde o ano de 2015 até 2019. Para a inclusão dos artigos era visualizado a aderência à questão de pesquisa, ou seja, precisa ser uma abordagem de ensino de processo ou qualidade de software. Durante a busca eram lidos os títulos, os resumos e se necessário até mesmo partes do estudo como forma de identificar a sua inclusão na pesquisa.

4.4. Extração e Síntese dos dados

Para a extração e a síntese dos dados foi adotado um modelo de avaliação para cada um dos artigos inclusos, segundo os seguintes tópicos: pontos fortes, pontos fracos, limitações, metodologia adotada, a contribuição e os trabalhos futuros.

Na análise dos pontos extraídos dos artigos era adotada a síntese e análise temática, proposta por CRUZES e DYBÅ (2011), onde foram identificadas as temáticas associadas a cada tópico onde o artigo era avaliado.

Este estudo selecionou um total de 29 artigos relevantes à questão de pesquisa, conforme ilustrado na Tabela 1. Esses estudos estão listados com identificadores, com o acrônimo Estudo (E), para melhor referência ao longo do texto, conforme pode ser visualizado no Apêndice A. Vale ressaltar que a conferência CSEE&T não ocorreu no ano de 2019, por isso na tabela se apresenta o “-” ao invés do “0”.

Tabela 1 – Inclusão de estudos por ano

Sigla	2015	2016	2017	2018	2019	Total
ICSE	0	2	0	0	0	2
CSEE&T	0	0	1	-	0	1
ITICSE	0	0	0	0	0	0
CSEDU	0	0	0	0	0	0
SIGCSE	0	0	0	0	1	1
ICSOF	0	0	0	0	0	0
SBIE	1	0	2	0	0	3
SBQS	1	0	0	4	2	7
SBES	2	0	2	1	2	7
FIE	0	2	1	0	1	4
WEI	0	1	0	0	3	4

Fonte: Elaboração própria (2020).

5. RESULTADOS E DISCUSSÕES

Nesta seção apresentam-se os resultados e as discussões como forma de resposta às questões de pesquisa.

5.1. Números da pesquisa

Como forma de mapear os artigos aceitos foi feita uma busca quantitativa dos resultados, os quais apresentaram uma distribuição dos estudos em relação ao ano de publicação, vide Tabela 1. Podemos observar uma normalização de publicações no decorrer dos anos sobre o ensino de Engenharia de software, em específico sobre processo e qualidade de software, a qual fica em 5 estudos por ano.

Um dado bastante interessante extraído da Tabela 1 e que pode auxiliar no desenvolvimento das atividades de um trabalho futuro é um crescimento de estudos sobre qualidade e processo no ano de 2019, com 9, o que mostra ser um tópico que ganha evidência das pesquisas científicas.

5.2. Avaliação dos artigos

A forma para avaliação dos artigos foi através da verificação de 7 tópicos, sendo eles: (i) sobre o que o artigo trata; (ii) pontos fortes; (iii) pontos fracos; (iv) limitações; (v) metodologia; (vi) contribuição; e (vii) os trabalhos futuros. Essa avaliação foi realizada como forma de identificar o ineditismo para uma pesquisa futura. Quanto ao assunto que era tratado em cada um dos artigos, será explanado a seguir.

O estudo E01 com o título “*Can Software Engineering Students Program Defect-free? An Educational Approach*” relata a construção de uma extensão ao modelo PSP (*Personal Software Process*) com a DFP (*Defect-Free Programming*) para o ensino de teste de software dentro do contexto de processo de software. Além disso, foi realizado um experimento para verificar como o modelo adequa-se ao ambiente educacional, uma sala de aula. A contribuição do trabalho à literatura foi um novo modelo para o ensino de processo de software com a adição para teste de software. Os autores não relatam trabalhos futuros.

O estudo E02 com o título “*Impact of CS programs on the quality of test cases generation: An empirical study*” relata sobre a qualidade no desenvolvimento de teste de

software em relação à quantidade de conteúdo estudada. Os tipos de teste adotados pelo artigo foram caixa branca e caixa preta. A metodologia empregada foi de experimento. A contribuição foi de apontar a necessidade do ensino de teste de software dentro do conteúdo dos cursos de ciência da computação. Os autores não relataram trabalhos futuros.

No estudo E03 com o título “*The use of industry training strategies in a software engineering course: an experience report*” foi feito um mapeamento dos currículos ACM/IEEE com as necessidades da indústria e criada uma proposta de workshops para o ensino dessas competências e habilidades. Além disso, foi realizado um experimento para comparar a nova proposta com o ensino tradicional. Os autores fazem toda a descrição do experimento inclusive segregando as suas etapas. Como limitações não foi apresentada ameaças à validade do estudo. O trabalho apresentou duas contribuições importantes sobre o ensino de engenharia de software em geral, sendo elas: (i) para a indústria, pois através da identificação sobre as habilidades necessárias para o melhor desempenho do profissional, a instituição de ensino forma profissionais para o mercado de trabalho; e (ii) para academia, pois consegue aproximar teoria e prática, além do que é uma abordagem considerada como centrada no aluno, o que facilita a aprendizagem.

No estudo E04 com o título “*Pragmatic Software Testing Education*” relata-se o uso da abordagem pragmática para o ensino de teste de software, através de um experimento. Como ponto forte os autores relatam sobre a estrutura de um curso superior sobre teste de software. As contribuições do trabalho são uma nova forma de ensinar teste de software e como estruturar um curso sobre o conteúdo tratado.

No estudo E05 com o título “Um mapeamento sistemático sobre ensino de teste de software” foi realizado um mapeamento sobre o ensino de teste de software. Como pontos fortes do artigo podemos citar: questões bibliométricas; identifica um planejamento através de protocolo de pesquisa e formas de busca; apresenta e discute a validade da pesquisa. Sobre os pontos fracos relatamos não apresentar a testagem da “*string*” de busca adotada na pesquisa. A metodologia é o mapeamento sistemático da literatura. A principal contribuição foi a identificação das principais abordagens de ensino de teste de software. Os autores relatam que desejam como trabalho futuro o desenvolvimento de um jogo educacional para o ensino de teste de software e programação.

O estudo E06 com o título “Um Modelo Iterativo para o Ensino de Engenharia de Software Baseado em Abordagens Focadas no Aluno” retrata um modelo iterativo baseado nas principais abordagens focadas no aluno que são aplicadas no ensino de Engenharia de Software. Os autores descrevem a pesquisa, bem como a forma como foi avaliado o modelo através de um painel de especialista na área. Como limitações foi relacionada a falta de experimento para a avaliação do modelo. Porém, nos trabalhos futuros está previsto o desenvolvimento do experimento controlado e o desenvolvimento de uma plataforma para o ensino tendo como base o modelo.

No estudo E07 com o título “Abordagem *flipped classroom* em comparação com o modelo tradicional de ensino: uma investigação empírica no âmbito de teste de software” tem-se o relato sobre o uso da abordagem de ensino *flipped classroom* comparada com a abordagem tradicional no contexto de Teste de Software. A metodologia selecionada foi o experimento e como limitações identificamos o não apontamento de ameaças à validade do estudo. Como contribuição temos uma sala de aula invertida para ensino de teste de software. Os trabalhos futuros relatados são o novo uso da abordagem definida.

No estudo E08 com o título “O Ensino Integrado de Gestão de Qualidade de Software e Engenharia de Software Aplicada como Modelo de Educação Interdisciplinar: Um Relato de Experiência no Ensino de Graduação” relata-se a experiência sobre uma abordagem interdisciplinar entre as disciplinas de Gestão de Qualidade de Software e

Engenharia de software aplicada, onde os projetos desenvolvidos foram avaliados pelo grupo de qualidade. A metodologia adotada foi o experimento. A contribuição apresentada é uma forma de ensinar qualidade de software com projetos interdisciplinares. Não há indicação de trabalhos futuros pelos autores.

O estudo E09 com o título “*A survey on software testing education in brazil*” apresenta a identificação da forma como o conteúdo teste de software é trabalhado, os mecanismos de apoio utilizados nas práticas de ensino, os desafios impostos e os instrumentos utilizados para avaliar a aprendizagem dos alunos. A metodologia é o *survey* e a limitação foi a amostragem conseguida para a pesquisa. Como contribuição relata-se: (i) nomenclatura da disciplina para teste de software; (ii) carga-horária disponível para a disciplina; (iii) conteúdos e técnicas de teste ensinadas; (iv) nível de conhecimento dos professores. Por trabalhos futuros, os autores identificaram: a realização do *survey* em inglês; e investir na experimentação de novas abordagens de ensino para o conteúdo.

O estudo E10 com o título “*Evaluating the students' experience with the Scrum Card Game: an experience report in a Software Engineering course*” descreve sobre o uso de um jogo, *Scrum Card Game*, no ensino de engenharia de software, através de um experimento. Para avaliação foi adotado para o jogo o modelo MEEGA, e para os dados coletados dos jogadores a metodologia *Grounded Theory*. A contribuição do estudo é a criação de um jogo para ensino de processo de software. Os trabalhos futuros relatados pelos autores são a execução de novos experimentos, assim como a criação do jogo em formato digital.

O estudo E11 com o título “*Gamification and Evaluation the Use of the Function Points Analysis Technique in Software Quality Subjects: The Experimental Studies*” relata sobre o uso de gamificação no ensino do conteúdo análise de pontos por função. A metodologia adotada foi através de um experimento controlado. Como contribuição é o uso da gamificação para ensino de análise de pontos por função. O trabalho futuro relatado pelos autores foi a reexecução do experimento para a melhoria das técnicas de ensino.

O estudo E12 com o título “*ProcSoft: A Board Game to Teach Software Processes Based on ISO/IEC 29110 Standard*” relata a construção de um jogo de tabuleiro definido para promover a definição de processos de software com base na norma ISO / IEC 29110, o ProcSoft. A metodologia adotada para avaliar a proposta foi o experimento, porém os autores relatam a dificuldade de captura do *feedback* dos jogadores. A contribuição do estudo para a literatura foi uma forma de ensino de processos de software com base na norma ISO / IEC 29110 através de um jogo de tabuleiro. Os trabalhos futuros relatados são a realização de mais experimentos e a melhoria do jogo.

O estudo E13 com o título “*Can we use the Flipped Classroom Model to teach Black-box Testing to Computer Students?*” apresenta um estudo experimental realizado para verificar a adequação do modelo de sala de aula invertida no ensino do conteúdo testes de software, especialmente testes de caixa preta. Como ponto fraco registra-se a não informação clara sobre como está sendo avaliada a aprendizagem dos alunos. Para trabalhos futuros os autores relatam investigar se os alunos que aprenderam o teste de software com o modelo de sala de aula invertida podem identificar com precisão os elementos exigidos pelos critérios de teste de software.

O estudo E14 com o título “*Is It Worth Using Gamification on Software Testing Education? An Experience Report*” trata-se de um estudo experimental desenvolvido para avaliar o impacto da gamificação no ensino de testes de software. Como limitações apontamos a falta de descrição e informação sobre as variáveis analisadas no experimento. Como contribuição temos um relato dos pontos positivos e negativos na adoção de gamificação no ensino de teste de software. Como trabalhos futuros são relatadas a

melhoria da abordagem, bem como a reexecução do experimento em outros contextos com finalidade de gerar dados estatísticos.

O estudo E15 com o título “A metodologia SCRUM como mobilizadora da prática pedagógica: um olhar sobre a engenharia de software” trata sobre a metodologia Scrum como alternativa para práticas pedagógicas, apresentando investigação com alunos da disciplina de engenharia de software de um curso técnico em informática. A metodologia adotada foi a do experimento e como fraqueza registra-se o uso de tempo fora de sala de aula. A contribuição é o uso da metodologia Scrum no ensino profissionalizante. Os autores não relatam trabalhos futuros.

O estudo E16 com o título “Análise da Relevância dos Tópicos e da Efetividade das Abordagens para o Ensino de Engenharia de Software” apresenta os resultados de um *survey* sobre a adoção e aprendizagem de tópicos de Engenharia de Software recomendados pelos currículos de referência da ACM/IEEE e SBC. Além disso, sobre quais abordagens de ensino são adotadas pelos professores e quais destas os alunos consideram efetivas para seu aprendizado. Como ponto forte podemos identificar o desenvolvimento de um protocolo de implementação do *survey*. Sobre pontos fracos e limitações registramos a quantidade de questionários respondidos e o não uso com profissionais da área. Como trabalho futuro os autores relataram definir um plano de ensino que enfoque nos tópicos relevantes à disciplina engenharia de software.

O estudo E17 com o título “*Problem-based learning to align theory and practice in software testing teaching*” descreve sobre o ensino de teste de software através de aprendizagem baseada em projetos alinhando teoria e prática, com o uso de experimento e análise de conteúdo. Como ponto fraco faltou identificar fatores externos que podem interferir no resultado do experimento. No quesito trabalhos futuros, os autores não os relataram, porém é válida a construção de um guia para como adotar aprendizagem baseada em problemas no ensino de teste de software.

O estudo E18 com o título “*PBL Integration into a Software Engineering Undergraduate Degree Program Curriculum: An Analysis of the Students’ Perceptions*” relata sobre a análise de percepção dos estudantes na aprendizagem baseada em problemas no curso de engenharia de software. Como ponto positivo relata as melhorias a serem implementadas para a aprendizagem coletiva. A metodologia adotada foi a de questionário, porém não houve uma descrição sobre como foram analisados os dados. Como trabalhos futuros foi indicado investigar a percepção dos autores sobre o estudo.

O estudo E19 com o título “*GreaTest: a card game to motivate the software testing learning*” relata sobre um jogo educacional de cartas que coloca os participantes no papel de analistas de teste. Eles indicam quais tipos de teste são mais apropriados para testar cada cenário apresentado nos cartões de jogo. Este jogo foi avaliado com professores, alunos de graduação e pós-graduação através de um experimento. Os resultados indicam que seu uso contribui para o aprendizado dos testes de software e a motivação dos alunos. Para trabalho futuro é indicado replicar o experimento para avaliação e melhoria.

O estudo E20 com o título “*Fábrica de Tecnologias Turing: A Strategy for Effective Teaching of Software Engineering*” apresenta uma abordagem acadêmica da fábrica de software usada como uma estratégia de ensino eficaz para a Engenharia de Software. Além disso, os autores definem o modelo através da metodologia SCRUM e Open UP. A metodologia adotada foi a de análise comparativa e o experimento. O trabalho futuro registrado é a análise dos pontos trabalhados dentro da fábrica de software para inclusão no PPC – Projeto Pedagógico do Curso.

O estudo E21 com o título “*Students Perception on the use of Project-Based Learning in Software Engineering Education*” descreve o uso de aprendizagem baseada em

projetos para o ensino de engenharia de software com experimento e grupo de controle e sem controle. A contribuição e o trabalho futuro é usar como modelo para futuras projeções de disciplinas e conteúdo da área de ciência da computação com a aprendizagem baseada em projetos.

No estudo E22 com o título “*Issues in student valuing of software engineering best practices*” os autores descrevem a necessidade de avaliar as habilidades de engenharia de software como um meio para melhorar o ensino de engenharia de software. A metodologia adotada foi a de revisão da literatura sobre os domínios do conteúdo. Como contribuição temos: uma avaliação e justificativa sobre o uso de domínio afetivo no conteúdo de teste de software.

No estudo E23 com o título “*An agile software engineering process improvement game*” é relatado pelos autores uma proposta para o desenvolvimento de um jogo com a inteligência artificial para o ensino de melhoria de processo de software. Como pontos fracos elencamos que o estudo é apenas uma proposta. O trabalho futuro é o desenvolvimento da plataforma e testagem em sala de aula.

O estudo E24 com o título “*Educational games: A contribution to software testing education*” trata sobre o ensino de teste de software através de um jogo, além disso realiza um mapeamento da literatura sobre os temas envolvidos e realiza um experimento com coleta de dados sobre motivação e aprendizado. Como pontos fracos elencamos não apresentar contenções à validação e não relatar o planejamento do experimento. A contribuição é o ensino de teste de software através de um jogo. Como trabalhos futuros é o uso da abordagem em sala de aula.

O estudo E25 com o título “*Integration of Software Testing to Programming Assignments: An Experimental Study*” retrata a inserção do teste de software nas disciplinas de programação. Como avaliação é adotada a metodologia de experimento. Como pontos fracos e limitação é o tempo de sala de aula, uma vez que teste de software tem conteúdo vasto assim como programação. A contribuição apresentada é a inserção de teste de software em disciplinas de programação.

No estudo E26 com o título “*O Uso de Recursos Lúdicos Para o Ensino de Processos em Engenharia de Software*” é tratado sobre o uso de recursos lúdicos no ensino de processo de software. Como pontos fortes os autores relatam: o histórico sobre a evolução das metodologias de ensino e definem como a engenharia de software é tratada em sala de aula. Como limitação registramos a falta de avaliação na aprendizagem. Sobre a metodologia é adotado o experimento e o questionário para avaliação em especial de motivação. Como contribuição temos uma nova forma de ensinar processo de software.

No estudo E27 com o título “*IslandTest: jogo educativo para apoiar o processo ensino-aprendizagem de testes de software*” é relatada a experiência no desenvolvimento de uma ferramenta online com base em jogos sérios para o ensino de teste de software. Como pontos fortes apresenta trabalhos correlatos na literatura. O ponto fraco elencado é a não avaliação em relação ao conteúdo. A metodologia adotada como forma de avaliação foi o questionário para verificar a motivação. A contribuição do estudo é o ensino de teste de software através de jogo. Como trabalho futuro é indicada a utilização de enredos de séries e filmes famosos como um ponto de geração do elemento de jogo narrativa épica para chamar a atenção dos estudantes no desenvolvimento das atividades.

No estudo E28 com o título “*Playing the Project: Incorporating Gamification into Project-based Approaches for Software Engineering Education*” os autores relatam sobre o desenvolvimento de uma gamificação com aprendizagem baseada em projetos para ensino de engenharia de software. A metodologia adotada para avaliar a proposta foi a de experimento. Como contribuição foi elencada a forma de ensino de engenharia de software

através de aprendizagem baseada em projetos e gamificação, além disso a definição de metas para o ensino dessa disciplina. Por trabalhos futuros os autores relataram a proposição de um *framework* para apoio ao uso de gamificação e metodologias ativas no ensino de engenharia de software.

No estudo E29 com o tema “EduGamification: uma metodologia de gamificação para apoiar o processo ensino-aprendizagem” os autores trabalham sobre a construção de gamificação para apoiar o ensino de disciplinas teóricas. Como testagem e avaliação usaram a disciplina de engenharia de software. Como pontos fracos relatamos a falta de avaliação de aprendizagem. Para a avaliação o estudo adotou como modelo questionário e entrevista com os alunos sobre a proposta, com ênfase na motivação para o aprendizado. A contribuição do artigo é o ensino de engenharia de software através de gamificação e metodologias ativas. Como trabalhos futuros foi apontado o desenvolvimento de uma plataforma com o princípio de suportar o processo.

5.3. Proposta de Ensino em Engenharia de Software

As propostas de ensino em Engenharia de software foram coletada dentro da literatura e sintetizadas na Figura 1. Como é possível perceber no decorrer dos estudos, a maioria optou pelo uso de jogo para ensino, com 9, gamificação ficou em segundo, com 4, e Modelo, com 3. Portanto, para a concepção de uma nova abordagem uma boa prática seria adotar as propostas que a literatura apresenta.

Vale ressaltar que tivemos alguns artigos contabilizados como revisão da literatura, portanto não foram selecionados para serem discutidos dentro desse tópico do artigo.

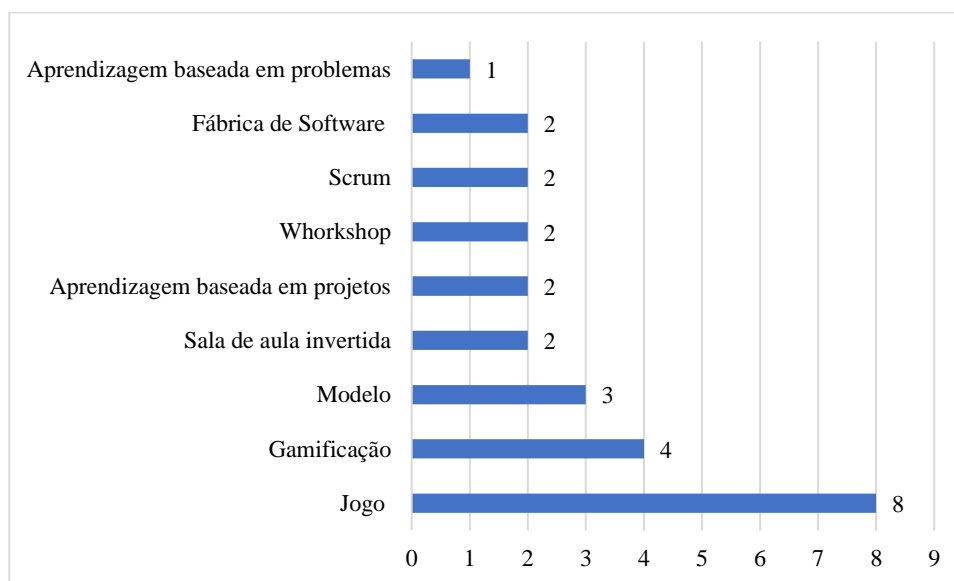


Figura 1. Propostas de Ensino em Engenharia de Software

Fonte: Elaboração própria (2020).

5.4. Tópicos abordados em Engenharia de Software

No que tange aos tópicos abordados sobre Engenharia de Software, o relato dos estudos foi sintetizado para gerar a Figura 2. Dentro da percepção é visível que o principal tópico é o Teste de software, com 13 indicações, Engenharia de Software não especificando o tópico do conteúdo, foram 11, e Processo de Software, com 3 indicações.

Vale ressaltar que por teste de software já possuir uma grande quantidade de estudos na literatura não seria uma proposta validada pra novos estudos, assim como Engenharia de software no geral. Porém, adotar como conteúdo base Processos de

Software apresenta-se como uma boa proposta uma vez que têm ascendência na quantidade de estudos.

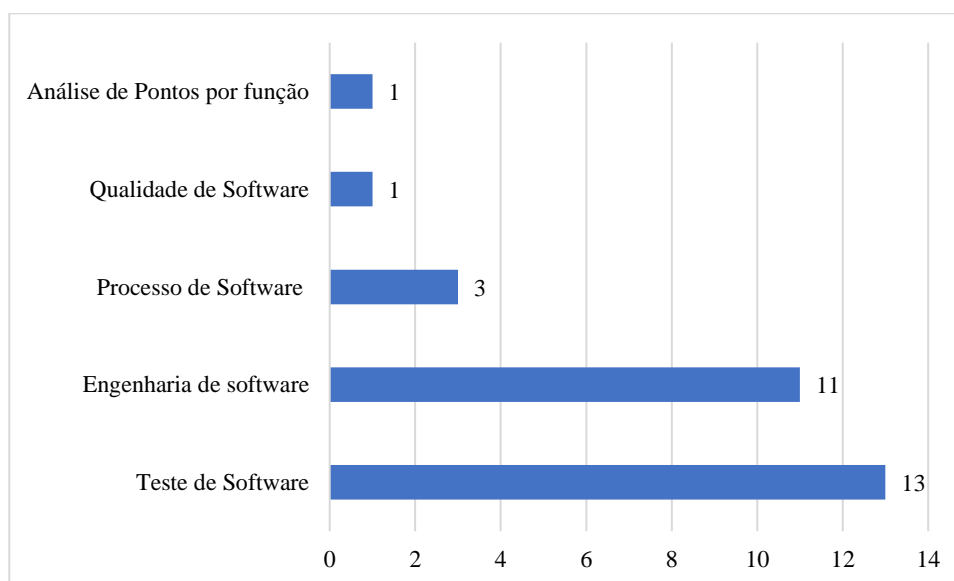


Figura 2. Tópicos abordados em Engenharia de Software

Fonte: Elaboração própria (2020).

6. AMEAÇAS À VALIDADE

Na revisão *Ad-hoc* da literatura o viés sobre a seleção dos estudos é uma das principais ameaças a sua validade enquanto pesquisa científica. Para tentar minimizar essa situação, adotou-se como base uma questão de pesquisa e critérios estabelecidos para a seleção das fontes, bem como dos estudos. Vale ressaltar que a pesquisa foi conduzida por um pesquisador e orientada por um professor com larga experiência em revisão da literatura, bem como no assunto sobre Engenharia de Software.

Sobre a avaliação dos artigos adotou-se os mesmos critérios para todos, conforme especificado na Seção 4 desse artigo, assim como ao final da revisão foi feita uma análise pelo professor pesquisador. Dentro de cada tópico extraído dos artigos foi realizada uma categorização das abordagens, através da técnica de Síntese e Análise Temática proposta por (CRUZES e DYBÅ, 2011). Por meio dessa técnica foi possível identificar os principais assuntos de realização do trabalho, sendo eles: as propostas de ensino e os tópicos trabalhos em Engenharia de Software.

7. CONCLUSÕES E TRABALHOS FUTUROS

Como forma de atender ao objetivo de identificar abordagens centradas no aluno sobre o ensino de processo e qualidade de software, além dos tópicos estudados pela literatura foi conduzida uma revisão *ad-hoc* da literatura. A revisão contou com um planejamento e uma questão de pesquisa norteadora.

Um total de 29 estudos sobre abordagens e revisões da literatura no contexto do processo de ensino e aprendizagem de temas em engenharia de software, em especial processo e qualidade de software, foram selecionados e extraídos. A principal abordagem trabalhada hoje na literatura é a de jogos e gamificação para o ensino, sendo que a forma adotada para avaliação e análise de como o estudo foi conduzido foi através do experimento. Outro ponto importante mapeado foi sobre a ascensão de publicações sobre

ensino em engenharia de software e tópicos relevantes para iniciar e desenvolver uma pesquisa.

Logo, através dos resultados desta revisão *ad-hoc* temos uma noção de como conduzir um estudo sobre ensino de processo de software com abordagens centradas no aluno em cursos de Tecnologia da Informação. Portanto, pretende-se como trabalho futuro alinhar a identificação da pesquisa como uma justificativa e coleta de informações para modelar uma abordagem para ensino de processo de software. Vale ressaltar um estudo futuro que é entender como o mercado e a área acadêmica veem e compreendem processo de software, a partir do uso da técnica de *survey*.

REFERÊNCIAS BIBLIOGRÁFICAS

BESSA, B., CUNHA, M., FURTADO, F. (2012) Engsoft: Ferramenta para simulação de ambientes reais para auxiliar o aprendizado baseado em problemas (pbl) no ensino de engenharia de software. In Anais do XX Workshop sobre Educação em Informática. Curitiba-PR.

BOURQUE, P., DUPUIS, R. (2004) Software engineering body of knowledge (SWEBOK). *IEEE Computer Society, EUA*.

CUNHA, J. A. O., MARQUES, G. A., LEMOS, W. L., CÂMARA Jr, U. D., VASCONCELLOS, F. J. (2018) Software engineering education in Brazil: a mapping study. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering* (pp. 348-356).

FERREIRA, T., VIANA, D., FERNANDES, J., SANTOS, R. (2018). Identifying emerging topics and difficulties in software engineering education in Brazil. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering* (pp. 230-239).

ISO/IEC. (2011) ISO/IEC 25010: 2011 Systems and software engineering--Systems and software Quality Requirements and Evaluation (SQuaRE)--System and software quality models.

MAFRA, S. N., TRAVASSOS, G. H. (2006) Estudos Primários e Secundários apoiando a busca por Evidência em Engenharia de Software. Universidade Federal do Rio de Janeiro. Rio de Janeiro. Brasil.

MEIRA, S. (2015) Sistemas de Informação e Engenharia de Software – Cadê as Escolas? Revista da SBC Engenharia de Software - Qual é o impacto da ES no mercado de Computação e na sociedade como um todo? Porto Alegre, Brasil.

MOERNO, A. M., SANCHEZ-SEGURA, M. I., MEDINA-DOMINGUEZ, F., CARVAJAL, L. (2012) Balancing software engineering education and industrial needs. *Journal of systems and software*, 85(7), 1607-1620.

NUNES, D. J., REIS, C., REIS, R. (2010) Educação em Engenharia de Software: A carreira de pesquisador em engenharia de software: princípios, conceitos e direções. Salvador.

SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. (2017) Referenciais de formação para os cursos de graduação em Computação 2017. Brasil.

WANGENHEIM, C. G., SILVA, D. A. (2009) Qual conhecimento de engenharia de software é importante para um profissional de software? Proceedings of the Fórum de Educação em Engenharia de Software, 2, 1-8.

APÊNDICE A – ARTIGOS ACEITOS POR ORDEM DAS CONFERÊNCIAS – MODELO APA DE CITAÇÃO

E01 - Rong, G., Zhang, H., Shan, Q., & Shao, D. (2016, May). Can Software Engineering Students Program Defect-free? An Educational Approach. In 2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C) (pp. 364-373). IEEE.

E02 - Gómez, O. S., Vegas, S., & Juristo, N. (2016, May). Impact of CS programs on the quality of test cases generation: An empirical study. In Proceedings of the 38th International Conference on Software Engineering Companion (pp. 374-383).

E03 - Portela, C., Vasconcelos, A., Oliveira, S., & Souza, M. (2017, November). The use of industry training strategies in a software engineering course: an experience report. In 2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T) (pp. 29-36). IEEE.

E04 - Aniche, M., Hermans, F., & van Deursen, A. (2019, February). Pragmatic Software Testing Education. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education (pp. 414-420).

E05 - Valle, P., Barbosa, E. F., & Maldonado, J. (2015, October). Um mapeamento sistemático sobre ensino de teste de software. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (Vol. 26, No. 1, p. 71).

E06 - Portela, C., Vasconcelos, A., & Oliveira, S. R. B. (2017, October). Um Modelo Iterativo para o Ensino de Engenharia de Software Baseado em Abordagens Focadas no Aluno. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (Vol. 28, No. 1, p. 304).

E07 - Paschoal, L. N., Silva, L., & Souza, S. (2017, October). Abordagem flipped classroom em comparação com o modelo tradicional de ensino: uma investigação empírica no âmbito de teste de software. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (Vol. 28, No. 1, p. 476).

E08 - Aguiar, Y. P., Saraiva, J., Dias, J., & Malheiros, Y. (2015). O Ensino Integrado de Gestão de Qualidade de Software e Engenharia de Software Aplicada como Modelo de Educação Interdisciplinar: Um Relato de Experiência no Ensino de Graduação. *Simpósio Brasileiro de Qualidade de Software*, 252-259.

E09 - Paschoal, L. N., & de Souza, S. D. R. S. (2018, October). A survey on software testing education in brazil. In *Proceedings of the 17th Brazilian Symposium on Software Quality* (pp. 334-343).

E10 - Moreira, G. G., & dos Santos Marques, A. B. (2018, October). 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* (pp. 344-353).

E11 - Santos, E. D., & Oliveira, S. R. (2018, October). Gamification and Evaluation the Use of the Function Points Analysis Technique in Software Quality Subjects: The Experimental Studies. In *Proceedings of the 17th Brazilian Symposium on Software Quality* (pp. 354-362).

E12 - Moura, V., & Santos, G. (2018, October). ProcSoft: A Board Game to Teach Software Processes Based on ISO/IEC 29110 Standard. In *Proceedings of the 17th Brazilian Symposium on Software Quality* (pp. 363-372).

E13 - Paschoal, L. N., Oliveira, B. R., Nakagawa, E. Y., & Souza, S. R. (2019, October). Can we use the Flipped Classroom Model to teach Black-box Testing to Computer Students?. In *Proceedings of the XVIII Brazilian Symposium on Software Quality* (pp. 158-167).

E14 - Jesus, G. M., Paschoal, L. N., Ferrari, F. C., & Souza, S. R. (2019, October). Is It Worth Using Gamification on Software Testing Education? An Experience Report. In *Proceedings of the XVIII Brazilian Symposium on Software Quality* (pp. 178-187).

E15 - Rocha, F. G., Sabino, R. F., & Acipreste, R. H. L. (2015). A metodologia SCRUM como mobilizadora da prática pedagógica: um olhar sobre a engenharia de software. *Fórum de educação em engenharia de software*, 13-23.

E16 - Portela, C. S., Vasconcelos, A. M., & Oliveira, S. R. (2015). Análise da Relevância dos Tópicos e da Efetividade das Abordagens para o Ensino de Engenharia de Software. In *Fórum de Educação em Engenharia de Software (FEES)*. In VI Congresso Brasileiro de Software: Teoria e Prática (CBSOFT).

E17 - Cheiran, J. F. P., de M. Rodrigues, E., de S. Carvalho, E. L., & da Silva, J. P. S. (2017, September). Problem-based learning to align theory and practice in software testing teaching. In *Proceedings of the 31st Brazilian Symposium on Software Engineering* (pp. 328-337).

E18 - Guedes, G. T. A., Bordin, A. S., Mello, A. V., & Melo, A. M. (2017, September). PBL Integration into a Software Engineering Undergraduate Degree Program Curriculum: An Analysis of the Students' Perceptions. In *Proceedings of the 31st Brazilian Symposium on Software Engineering* (pp. 308-317).

E19 - Beppe, T. A., de Araújo, Í. L., Aragão, B. S., de Sousa Santos, I., Ximenes, D., & Andrade, R. M. C. (2018, September). GreatTest: a card game to motivate the software testing learning. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering* (pp. 298-307).

E20 - Pocivi, V. C. B., Marins, W. F., & Diogo, K. S. (2019, September). Fábrica de Tecnologias Turing: A Strategy for Effective Teaching of Software Engineering. In Proceedings of the XXXIII Brazilian Symposium on Software Engineering (pp. 523-531).

E21 - Souza, M., Moreira, R., & Figueiredo, E. (2019, September). Students Perception on the use of Project-Based Learning in Software Engineering Education. In Proceedings of the XXXIII Brazilian Symposium on Software Engineering (pp. 537-546).

E22 - Frezza, S. T. (2016, October). Issues in student valuing of software engineering best practices. In 2016 IEEE Frontiers in Education Conference (FIE) (pp. 1-4). IEEE.

E23 - Maxim, B. R., Kaur, R., Apzynski, C., Edwards, D., & Evans, E. (2016, October). An agile software engineering process improvement game. In 2016 IEEE Frontiers in education Conference (FIE) (pp. 1-4). IEEE.

E24 - Valle, P. H. D., Toda, A. M., Barbosa, E. F., & Maldonado, J. C. (2017, October). Educational games: A contribution to software testing education. In 2017 IEEE Frontiers in education Conference (FIE) (pp. 1-8). IEEE.

E25 - Avellar, G. M., da Silva, R. F., Scatalon, L. P., Andrade, S. A., Delamaro, M. E., & Barbosa, E. F. (2019, October). Integration of Software Testing to Programming Assignments: An Experimental Study. In 2019 IEEE Frontiers in Education Conference (FIE) (pp. 1-9). IEEE.

E26 - Castro, R., & Souza, G. (2020, January). O Uso de Recursos Lúdicos Para o Ensino de Processos em Engenharia de Software. In Anais do XXIV Workshop sobre Educação em Computação (pp. 270-279). SBC.

E27 - Queiroz, R., Pinto, F., & Silva, P. (2019, July). IslandTest: jogo educativo para apoiar o processo ensino-aprendizagem de testes de software. In Anais do XXVII Workshop sobre Educação em Computação (pp. 533-542). SBC.

E28 - Souza, M., Moreira, R., & Figueiredo, E. (2019, July). Playing the Project: Incorporating Gamification into Project-based Approaches for Software Engineering Education. In Anais do XXVII Workshop sobre Educação em Computação (pp. 71-80). SBC.

E29 - Pinto, F., & Silva, P. (2019, July). EduGamification: uma metodologia de gamificação para apoiar o processo ensino-aprendizagem. In Anais do XXVII Workshop sobre Educação em Computação (pp. 414-428). SBC.