

Avaliando a Colaboração no Ensino de Engenharia de Software: Percepções a partir de uma Revisão Rápida

Enzo Kleinpaul
Instituto Federal Farroupilha
São Borja, Brasil
enzo.2021310072@aluno.iffar.edu.br

Rafael Parizi
Instituto Federal Farroupilha
São Borja, Brasil
rafael.parizi@iffar.edu.br

RESUMO

Atividades colaborativas de co-criação são amplamente adotadas no ensino de Engenharia de Software para simular cenários reais de desenvolvimento. Impulsionadas por práticas de Design Thinking e Métodos Ágeis, essas atividades envolvem os estudantes em diversas fases do ciclo de vida do desenvolvimento. No entanto, a ausência de métricas para avaliar a colaboração limita a capacidade de verificar sua efetividade educacional. Este estudo tem como objetivo identificar como a colaboração é avaliada em atividades de co-criação no contexto do ensino de Engenharia de Software, mapeando sistematicamente as estratégias relatadas na literatura. Para isso, foi realizada uma revisão rápida aplicando uma *string* de busca nas bases de dados IEEE Xplore e Scopus. A busca retornou 909 estudos, dos quais 32 foram selecionados com base em critérios de inclusão e exclusão. Como resultados, se observou que a maioria dos estudos avalia a colaboração por meio de questionários com perguntas em escala Likert. Alguns adotam abordagens combinadas, integrando avaliação entre pares, registros de uso de ferramentas e observações de instrutores. No entanto, muitas avaliações continuam focadas nos resultados finais, em vez do processo colaborativo em si. Desta forma, este trabalho oferece a educadores e profissionais uma visão geral das práticas atuais de avaliação da colaboração.

PALAVRAS-CHAVE

Avaliação da Colaboração, Educação em Engenharia de Software, Revisão Rápida da Literatura

1 Introdução

Atividades de cocriação colaborativa em ambientes educacionais de Engenharia de Software (ES) tem sido discutido na literatura devido ao seu potencial para dinamizar o processo de ensino-aprendizagem no desenvolvimento de software [49]. Essas atividades, muitas vezes fundamentadas em abordagens como Design Thinking (DT) e Metodologias Ágeis, promovem a interação ativa entre os participantes e a construção conjunta de soluções [39, 49].

Empresas de desenvolvimento de software têm valorizado habilidades como a colaboração, apontada como uma *soft skill* necessária entre os profissionais [5]. Essa valorização torna a competência colaborativa relevante no ensino de ES, contribuindo para a formação de profissionais bem preparados e habituados a atuar em ambientes colaborativos [12]. Além disso, atividades colaborativas se tornam relevantes no ensino dos processos de desenvolvimento de software, ao possibilitar a divisão de funções, a valorização das contribuições individuais e coletivas e por fomentar o desenvolvimento de outras *soft skills* [10, 12, 52].

No entanto, embora atividades colaborativas sejam amplamente utilizadas na indústria, ainda há uma carência de um modelo de avaliação consolidado capaz de avaliar adequadamente a colaboração como *soft skill* no desenvolvimento de software. Essa lacuna torna-se mais evidente no contexto educacional, onde faltam métricas bem definidas que realmente caracterizem e avaliem de forma precisa os elementos envolvidos na colaboração entre estudantes. Essa dificuldade se associa à escassez de estudos que investiguem como essas avaliações têm sido conduzidas na prática. Em trabalho anterior [1], foram desenvolvidas atividades colaborativas baseadas em DT no ensino de ES; entretanto, não se encontrou referências consolidadas na literatura que orientassem a definição de critérios ou métricas para avaliar a colaboração de forma sistemática, evidenciando a necessidade de investigações sobre o tema.

Posto isto, o objetivo deste trabalho é identificar como a colaboração tem sido explorada no contexto do ensino de Engenharia de Software. Nesse sentido, o artigo busca responder a seguinte questão de pesquisa: “Como a colaboração tem sido promovida, avaliada e discutida em atividades de cocriação no ensino de Engenharia de Software?”. Para isso, foi realizada uma revisão rápida da literatura, inspirados no processo apresentado por Cartaxo *et al.* (2020) [11] e Hidalgo *et al.* (2023) [20], buscando mapear diferentes abordagens de avaliação já utilizadas, especialmente aquelas que colocam a colaboração como elemento central. A revisão buscou artigos nas bases digitais IEEE e Scopus. Foram selecionados 32 estudos que exploram avaliação da colaboração, seja por meio da aplicação prática de atividades de cocriação, seja por revisões anteriores que investigam métodos utilizados em práticas colaborativas.

As contribuições deste trabalho são: (i) a consolidação das práticas utilizadas para avaliar a colaboração em atividades de cocriação no ensino de ES; (ii) a categorização das atividades educacionais e contextos em que a colaboração é promovida; (iii) o mapeamento dos métodos, instrumentos e ferramentas tecnológicas empregados na avaliação da colaboração; e (iv) o levantamento de desafios enfrentados, oferecendo subsídios para a melhoria de práticas avaliativas e futuras investigações na área.

O restante deste artigo está organizado da seguinte maneira: a Seção 2 apresenta o referencial teórico, abordando os conceitos de cocriação colaborativa, desenvolvimento de *soft skills* e estratégias de avaliação da colaboração no ensino de Engenharia de Software; a Seção 3 descreve o protocolo adotado na revisão rápida; a Seção 4 apresenta os resultados obtidos com base nas questões de pesquisa; a Seção 5 discute os achados à luz da literatura; a Seção 6 descreve as ameaças à validade do estudo; e, por fim, a Seção 7 traz as considerações finais e sugestões para pesquisas futuras.

2 Referencial Teórico

2.1 Cocriação e o Ensino de Engenharia de Software

O ensino de Engenharia de Software visa formar profissionais capazes de compreender e atuar em todas as etapas do ciclo de vida de um software, desde a concepção até a entrega final, incluindo atividades de testes, manutenção e gerenciamento [38]. Para atingir esse objetivo, o currículo das disciplinas e/ou curso de Engenharia de Software, tanto quanto ementas que abordam os conceitos e a prática relacionados a essas fases, promovem o entendimento teórico aliado à aplicação prática por meio do desenvolvimento de projetos [38, 45].

Atividades de cocriação colaborativa são frequentemente implementadas em disciplinas voltadas aos processos de desenvolvimento de software [45], por meio da introdução de fundamentos teóricos e desenvolvimento de projetos práticos como estratégia de trabalho, incentivando os alunos a aplicarem os conhecimentos adquiridos de forma integrada [48]. Essa abordagem comumente incorpora atividades de cocriação baseadas em abordagens como o DT e as Metodologias Ágeis, cujas técnicas contribuem para o refinamento das etapas do desenvolvimento e para a construção compartilhada das soluções, facilitando a compreensão prática do processo [39, 49].

Essas técnicas contemporâneas promovem métodos de ensino centrados na prática, com divisão clara de funções entre os participantes, foco em interações frequentes e incrementais, além da ênfase na participação ativa de *stakeholders* [39, 49]. Por envolverem contato direto com clientes ou usuários finais, essas metodologias estimulam os alunos a compreenderem não apenas os requisitos técnicos, mas também as necessidades e expectativas dos usuários [39, 49]. Assim, tornam-se ferramentas eficazes para desenvolver competências alinhadas à realidade do mercado, ao mesmo tempo em que favorecem o aprendizado ativo e significativo sobre como um produto de software deve ser concebido e construído [48].

Metodologias ativas aplicadas no ensino de Engenharia de Software contribuem para formar profissionais capazes de atuar de maneira eficaz em equipes e ambientes colaborativos [32]. Assim como ocorre em outras áreas, o trabalho em equipe é parte integrante da prática profissional, tornando a competência colaborativa um requisito fundamental para quem deseja se destacar e construir uma carreira sólida [50].

A colaboração sustenta o desenvolvimento de diversas outras habilidades interpessoais, como empatia, resolução de problemas, comunicação eficaz, adaptabilidade e tomada de decisão. Essas habilidades são valorizadas no mercado de trabalho [17], considerada uma competência transversal de destaque [17, 25]. Dessa forma, colaborar deixa de ser uma habilidade isolada e passa a ser a base de um conjunto de competências, exploradas por meio de experiências práticas que simulam a realidade do mercado de trabalho [17, 25].

2.2 Estratégias de Avaliação da Colaboração em Contextos Educacionais

Metodologias colaborativas são utilizadas tanto em salas de aula quanto em ambientes corporativos. No entanto, nem sempre é possível verificar se estão sendo realmente efetivas [12]. Para isso, são realizadas avaliações que consideram aspectos como a facilidade de

aplicação, o tempo de execução e a qualidade do produto construído, entre outros [33]. Apesar disso, a colaboração frequentemente não recebe um foco bem estabelecido nesses processos, ou seja, não é avaliada de forma concreta [12].

Os métodos mais comuns para avaliar metodologias colaborativas são os *surveys* e a coleta de feedback, geralmente por meio de questionários ou entrevistas [17, 30, 33, 44]. Esses métodos apresentam limitações quando o objetivo é mensurar com precisão os aspectos da colaboração. Isso se deve, principalmente, à ausência de métricas bem definidas e amplamente aplicáveis. Alguns estudos chegam a propor métricas específicas, tais como aquelas aplicadas em *surveys*, porém, essas métricas são na maioria desenvolvidas para contextos muito específicos, o que dificulta sua reutilização e compromete a possibilidade de generalização [24, 37, 44].

Estudos como o de Nguyen *et al.* (2020) [37] avaliam a colaboração por meio de cálculos que medem o percentual de participação dos alunos, utilizando métricas específicas do projeto. Há também abordagens que adotam métodos distintos dos tradicionais, utilizando *frameworks*, software ou metodologia de avaliação desenvolvida especificamente para o contexto escolhido, com métricas igualmente formuladas para aquele projeto [46].

Métodos de avaliação utilizados podem ser classificados em três categorias principais: quantitativos, qualitativos e mistos. Os métodos quantitativos utilizam instrumentos como questionários estruturados ou análises estatísticas de interações, com o objetivo de gerar dados objetivos. Já os métodos qualitativos envolvem observações, entrevistas abertas e análises descritivas, permitindo uma compreensão mais contextual da colaboração. Por fim, os métodos mistos combinam elementos das duas abordagens, buscando uma avaliação mais completa e equilibrada [31, 33].

Dessa forma, observa-se que, embora existam métodos de avaliação da colaboração que se mostram efetivos em determinados contextos, eles ainda apresentam limitações, tais como a ausência de métricas consolidadas que permitam avaliar, de forma sistemática, a colaboração em atividades de cocriação colaborativa.

3 Protocolo da Revisão Rápida

Este artigo apresenta uma revisão rápida da literatura [11] com o objetivo de identificar como a colaboração tem sido avaliada em atividades vinculadas ao ensino de Engenharia de Software. A escolha por uma revisão rápida se justifica por ser um método que oferece uma visão consolidada, em curto prazo, e que possibilitaria identificar métricas, instrumentos e desafios relacionados à avaliação da colaboração em contextos educacionais. Esta abordagem se baseia nos princípios de revisões sistemáticas, mas com adaptações que permitem maior agilidade na coleta e análise dos estudos.

3.1 Problema de Pesquisa

O problema de pesquisa que orienta este estudo é a falta de estudos sobre métricas, desafios e abordagens utilizadas para avaliar a colaboração de estudantes em atividades de Engenharia de Software. Apesar de as práticas colaborativas serem adotadas em cursos da área, especialmente por meio de metodologias ativas e cocriação, ainda há pouco conhecimento sistematizado sobre como essa colaboração tem sido efetivamente medida e analisada.

Tabela 1: Estrutura PICOC adotada na revisão

Elemento	Descrição
Population	Estudantes e professores de Engenharia de Software; equipes de desenvolvimento ágil ("agile teams"), times distribuídos ("distributed teams")
Intervention	Avaliação da colaboração, métricas de colaboração, medição de desempenho em equipe ("team performance metrics"), análise de atividades colaborativas ("collaborative software development"), uso de práticas ágeis ("agile collaboration")
Comparison	—
Outcome	Identificar formas de avaliar a colaboração, medir sua eficácia, reconhecer padrões colaborativos em times de software ("collaboration metrics", "teamwork measurement")
Context	Ensino de Engenharia de Software, gestão de projetos ("project management"), atividades de cocriação, ambientes colaborativos distribuídos

O processo de planejamento e triagem inicial da revisão foi conduzido com o apoio da ferramenta Parsifal¹, que foi utilizada para importar os resultados das buscas, aplicar os critérios de inclusão e exclusão durante a leitura de títulos, resumos e textos completos, além de registrar decisões e comentários dos revisores. Após a seleção final dos estudos incluídos, os dados extraídos foram migrados para planilhas estruturadas com o objetivo de facilitar a análise qualitativa e a organização dos agrupamentos temáticos (por tipo de atividade, método, ferramenta e desafio). Essa combinação de ferramentas permitiu maior controle sobre as variáveis analisadas e será detalhada com mais clareza na versão revisada do artigo.

3.2 Questões de Pesquisa

Para orientar a revisão, foi estabelecida a seguinte questão principal de pesquisa (QP): “*Como a colaboração tem sido promovida, avaliada e discutida em atividades de cocriação no ensino de Engenharia de Software?*”.

A partir da QP, foram derivadas cinco questões auxiliares:

- **Q1:** Como as atividades de cocriação são estabelecidas em aulas de Engenharia de Software?
- **Q2:** Para qual finalidade (espaços de trabalho) a cocriação é utilizada?
- **Q3:** Quais métodos de avaliação são utilizados para avaliar a colaboração entre estudantes?
- **Q4:** Quais ferramentas e instrumentos são utilizados na avaliação da colaboração?
- **Q5:** Quais os principais desafios enfrentados na avaliação da colaboração em contextos educacionais?

3.3 Estratégia de Busca

Para delinear a estratégia de busca, foi utilizado o *framework* PICOC (*Population, Intervention, Comparison, Outcome, Context*)[23], conforme apresentado na Tabela 1. As *strings* de busca foram formuladas com base na *keywording strategy* conforme proposta por Petersen et al. (2015)[41], a fim de garantir abrangência e representatividade nos termos de busca.

As *strings* foram aplicadas nas bibliotecas digitais IEEE Xplore² e Scopus³. A escolha por essas duas bases se justifica por seu amplo reconhecimento e cobertura na área de ES, incluindo publicações voltadas à Educação em ES. Essa decisão está alinhada com o escopo de uma revisão rápida, que busca equilibrar abrangência e viabilidade de execução. A *string* utilizada foi:

String de busca
("collaboration metrics" OR "teamwork measurement" OR "collaborative software development" OR "team performance metrics" OR "agile collaboration") AND ("software engineering" OR "project management" OR "agile teams" OR "distributed teams")

3.4 Processo de Seleção de Artigos

A seleção dos estudos foi realizada com base em critérios de inclusão e exclusão, descritos na Tabela 2.

A Figura 1 apresenta o processo de seleção dos estudos incluídos na revisão, detalhando as etapas de exclusão e os respectivos percentuais. Inicialmente, foram identificadas 909 publicações, sendo 705 provenientes da base IEEE Xplore e 204 da base Scopus. Na primeira etapa, 3 publicações foram excluídas por se tratarem de duplicatas, não serem revisadas por pares ou não estarem disponíveis para acesso completo, resultando em uma taxa de exclusão de 0,33%. Em seguida, foram avaliadas 906 publicações com base na leitura do título, resumo e palavras-chave, o que levou à exclusão de 864 artigos (95,05%)⁴. Essa etapa contou com avaliação por dois pesquisadores e obteve um coeficiente de concordância Kappa de Cohen de 0,81, indicando concordância substancial. Após essa triagem inicial, 42 artigos foram lidos na íntegra e novamente avaliados por dois revisores, com um coeficiente Kappa de 0,89 (quase perfeito). Nessa fase, 10 estudos foram excluídos, resultando em uma taxa de exclusão de 1,1%. Ao final do processo, 32 artigos foram aceitos, correspondendo a uma taxa de inclusão de 3,52% e uma taxa total de exclusão de 96,48%.

Para garantir a consistência e confiabilidade do processo de seleção, dois pesquisadores revisaram independentemente uma amostra de 20% dos artigos identificados (n = 182). Em seguida, foi calculado o Coeficiente Kappa [26], uma medida estatística utilizada para estimar o grau de concordância entre avaliadores [51]. A Tabela 3 apresenta os valores obtidos nas etapas da seleção.

A Tabela 4 mostra a lista de artigos selecionados, incluindo ID (usado em todo o artigo), Título, Autores, Ano (de publicação), Local (Conferência ou Periódico) e Ref. (referência ao artigo).

3.5 Extração e Análise dos Dados

Após a seleção dos 32 estudos, os dados foram extraídos e organizados em planilhas estruturadas com base nas questões de pesquisa (Q1 a Q5). Para cada estudo, foram coletadas informações como: atividade realizada, número de participantes, público-alvo, metodologia empregada, objetivo da atividade, métodos de avaliação da colaboração, ferramentas e instrumentos utilizados, bem como os desafios relacionados.

²<https://ieeexplore.ieee.org/Xplore/home.jsp>

³<https://www.elsevier.com/products/scopus>

⁴Taxa de exclusão calculada considerando o valor global de artigos, 909

¹<https://parsif.al>

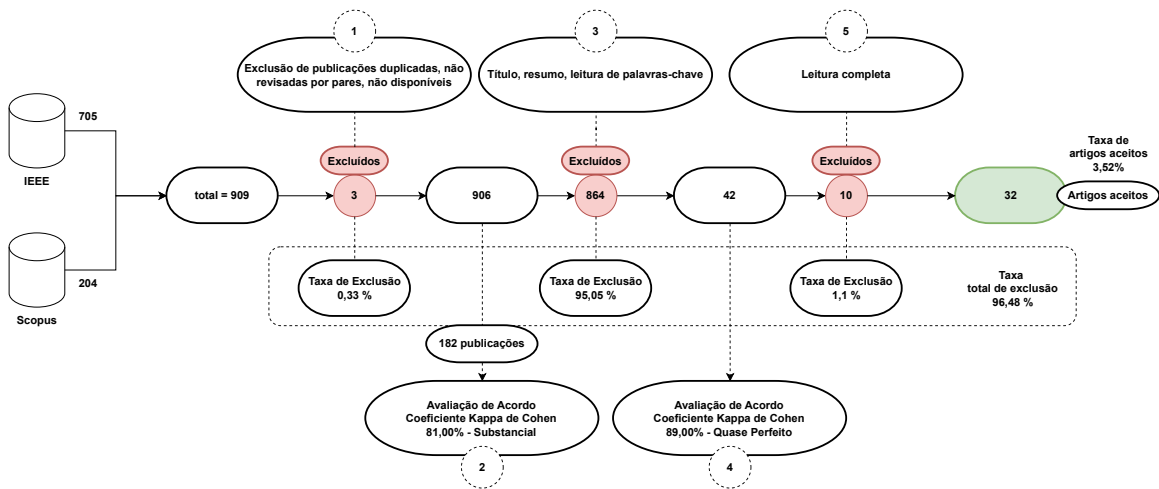


Figura 1: Diagrama do Processo de Seleção

Tabela 2: Critérios de Inclusão e Exclusão

Critérios de Inclusão	
CI1	Estudos que abordem ambientes educacionais
CI2	Aplicação de técnicas/metodologias de ensino ou colaboração
CI3	Estudos que abordem métricas de colaboração ou <i>soft skills</i>
CI4	Estudos em português ou inglês
CI5	Contexto relacionado à Engenharia de Software
Critérios de Exclusão	
CE1	Estudos com foco exclusivo em contextos profissionais/empresariais
CE2	Estudos fora da área de Engenharia de Software
CE3	Documentos duplicados ou incompletos
CE4	Estudos sem foco em colaboração ou avaliação
CE5	Estudos não revisados por pares ou sem acesso ao texto completo

Tabela 3: Valores de concordância (Coeficiente Kappa)

Etapas de Avaliação	Valor de Kappa
Seleção por título e resumo	0.81 – concordância substancial
Seleção por leitura completa	0.89 – concordância quase perfeita

A análise dos dados foi conduzida de forma qualitativa, por meio da categorização por similaridade entre os estudos. Essa abordagem permitiu identificar padrões recorrentes, agrupar práticas e estratégias utilizadas, e destacar lacunas e desafios enfrentados. O processo de agrupamento foi realizado manualmente por dois pesquisadores, de forma colaborativa e iterativa, com o objetivo de reduzir vieses individuais e garantir consistência nas classificações. Os agrupamentos resultantes são apresentados na Seção 4 e discutidos ao longo do artigo.

4 Resultados

Esta seção apresenta os achados obtidos a partir da análise dos 32 estudos selecionados na revisão. Os resultados são organizados com base nas questões de pesquisa propostas, abrangendo desde a caracterização das atividades de cocriação utilizadas no ensino

Distribuição de artigos por ano

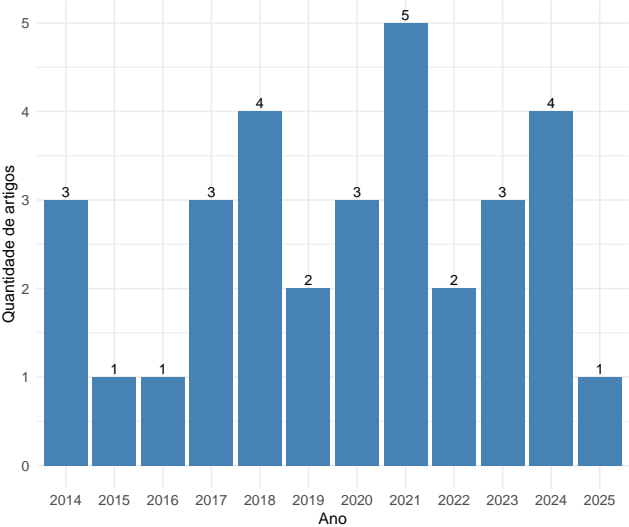


Figura 2: Distribuição de artigos por ano

de Engenharia de Software (Q1 e Q2), até os métodos, ferramentas e instrumentos empregados para avaliar a colaboração (Q3 e Q4), bem como os desafios enfrentados nesse processo (Q5). Para cada questão, são apresentados os agrupamentos temáticos por similaridade e as respectivas tabelas síntese, seguidos de uma descrição interpretativa dos dados coletados. A Figura 2 apresenta a distribuição dos 32 artigos incluídos na revisão de acordo com o ano de publicação. Apesar de não ser uma coleta exaustiva, observa-se uma concentração maior de estudos a partir de 2018, com destaque para o ano de 2021, que apresentou o maior número de publicações (5 artigos). Essa maior concentração pode indicar um crescente interesse da comunidade de ES por práticas colaborativas em ambientes de formação acadêmica.

Tabela 4: Lista de artigos selecionados na revisão

ID	Título	Autores	Ano	Local de Publicação	Ref.
1	Improving Teamwork in Software Engineering Projects in Higher Education	Geek et al.	2023	IEEE Frontiers in Education Conference (FIE)	[44]
2	SETAP: Software engineering teamwork assessment and prediction using machine learning	Petkovic et al.	2014	IEEE Frontiers in Education Conference (FIE)	[42]
3	Exploring Peer Evaluation Methods in Group Projects of Software Engineering Education	Rizquallah et al.	2024	IEEE Frontiers in Education Conference (FIE)	[9]
4	Using the random forest classifier to assess and predict student learning of Software Engineering Teamwork	Petkovic et al.	2016	IEEE Frontiers in Education Conference (FIE)	[43]
5	Toward Increasing Collaboration Awareness in Software Engineering Teams	Chowdhury et al.	2018	IEEE Frontiers in Education Conference (FIE)	[14]
6	Using GitHub Analytics to Assess the Quality of Collaboration in Software Engineering Teams	Le et al.	2024	IEEE Frontiers in Education Conference (FIE)	[27]
7	Software Metrics Classification for Agile Scrum Process: A Literature Review	Kurnia et al.	2018	International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)	[24]
8	Experiences and insights from using Github Classroom to support Project-Based Courses	Nelson e Ponciano	2021	International Workshop on Software Engineering Education for the Next Generation (SEENG)	[36]
9	Pilot experience applying an active learning methodology in a software engineering classroom	Alicia García-Holgado et al.	2018	IEEE Global Engineering Education Conference (EDU-CON)	[18]
10	Measure Students' Contribution in Web Programming Projects by Exploring Source Code Repository	Nguyen et al.	2020	International Computer Symposium (ICS)	[37]
11	Team Maturity in Software Engineering Teams	Marsicano et al.	2017	ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)	[30]
12	Gauging influence in software development teams	Marshall e Gamble	2015	IEEE Frontiers in Education Conference (FIE)	[29]
13	Assessment of a Hybrid Software Development Process for Student Projects: A Controlled Experiment	Wlodarski et al.	2021	International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)	[53]
14	Comparing Team Evaluation Software (Team+ and CATME)	Kemppainen et al	2022	IEEE Frontiers in Education Conference (FIE)	[22]
15	Metrics Driven Research Collaboration: Focusing on Common Project Goals Continuously	Schreiber et al.	2017	IEEE/ACM International Workshop on Software Engineering Research and Industrial Practice (SER&IP)	[47]
16	A Case Study in Class User Interface Design of Problem-Based Learning Modeling (UIDPBL)	Arwathananukul et al.	2021	Joint International Conference on Digital Arts, Media and Technology	[6]
17	Facilitating learning and startup formation in experience-based courses – A team-centered model	Cico et al.	2021	IEEE Global Engineering Education Conference (EDU-CON)	[15]
18	Classifying Composition of Software Development Team Using Machine Learning Techniques	Yuhana et al.	2022	International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)	[55]
19	Social Identity in Software Development	Backevik et al.	2019	International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)	[7]
20	WIP: ChatVis: Enhancing Academic Team Collaboration through WhatsApp Chat Analytics	González et al.	2024	IEEE Frontiers in Education Conference (FIE)	[40]
21	Assessing Soft Skills Development in Informatics Students Through Project-Based Learning and Agile Frameworks	Choque-Soto et al.	2024	International Symposium on Accreditation of Engineering and Computing Education (ICACIT)	[13]
22	Social Networks during Software Ecosystems' Death	Arantes et al.	2023	International Workshop on Software Engineering for Systems-of-Systems and Software Ecosystems (SESoS)	[4]
23	Continuously Evaluated Research Projects in Collaborative Decoupled Environments	Schmidts et al.	2018	International Workshop on Software Engineering Research and Industrial Practice (SER&IP)	[46]
24	Managing Soft Skills Development in Technological Innovation Project Teams: An Experience Report in the Automotive Industry	Maia et al.	2023	IEEE Frontiers in Education Conference (FIE)	[28]
25	Comparative Case Study of Plan-Driven and Agile Approaches in Student Computing Projects	Wlodarski et al.	2020	International Conference on Software, Telecommunications and Computer Networks (SoftCOM)	[54]
26	A multi-criteria item-based collaborative filtering framework	Bilge et al.	2014	International Joint Conference on Computer Science and Software Engineering (JCSSE)	[8]
27	A Computational Model for Improving the Accuracy of Multi-criteria Recommender Systems	Hassan e Hamada	2017	International Symposium on Embedded Multicore-Many-core Systems-on-Chip (MCSoc)	[19]
28	Student Perspectives on the Purpose of Peer Evaluation during Group Game Development Projects	Mitchell et al.	2021	Conference on United Kingdom & Ireland Computing Education Research	[34]
29	Examining Teamwork: Evaluating Individual Contributions in Collaborative Software Engineering Projects	Mitra e Gerber	2025	Technical Symposium on Computer Science Education	[35]
30	Introducing low-cost sensors into the classroom settings: Improving the assessment in agile practices with multimodal learning analytics	Cornide-Reyes et al.	2019	journal on the science and technology of sensors	[16]
31	Collaborative knowledge transfer via Wiki: A project based learning approach in software engineering	Abke et al.	2014	International Conference on Interactive Collaborative Learning (ICL)	[2]
32	The impact of undergraduate mentorship on student satisfaction and engagement, teamwork performance, and team dysfunction in a software engineering group project	Iacob e Faily	2020	Technical Symposium on Computer Science Education	[21]

4.1 Caracterização do Espaço de Ensino

Para entender em que cenário as atividades colaborativas são realizadas e visando responder de forma conjunta as questões Q1 e Q2, foram coletados dados dos artigos incluindo: atividade realizada, número de participantes, público-alvo, metodologia empregada e objetivo da atividade.

A Tabela 5 sintetiza os contextos educacionais e metodológicos nos quais as atividades de colaboração foram conduzidas. Cada estudo inclui as principais características do tipo de atividade realizada, o número de participantes, o público-alvo, as metodologias aplicadas e o objetivo principal da investigação.

A Tabela 6 apresenta o agrupamento dos estudos com base nos tipos de atividade ou metodologia utilizadas nas experiências educacionais analisadas. A maioria dos trabalhos tem como foco avaliar diretamente a colaboração entre estudantes, seja por meio de medições, percepção ou eficácia das interações (1, 3, 4, 5, 6, 10, 12, 14, 15, 20, 23, 30, 32). Outros estudos se dedicam ao desenvolvimento ou validação de ferramentas de apoio à colaboração (6, 8, 20, 31), enquanto um terceiro grupo busca compreender fatores que influenciam o trabalho em equipe, como perfil dos estudantes, participação individual ou formação das equipes (2, 5, 17, 18, 19, 29). Também foram identificadas iniciativas voltadas à promoção de *soft skills* e metodologias ativas (9, 13, 16, 21, 24, 25), além de trabalhos que propõem ou analisam práticas estruturadas de ensino aplicadas à formação em Engenharia de Software (7, 11, 22, 26, 27, 28).

Além disso, fica evidente a diversidade de estratégias de avaliação da colaboração — desde métodos tradicionais baseados em *surveys* até análises automatizadas e autoavaliação entre pares.

4.2 Métodos de Avaliação da Colaboração

Para responder à questão de pesquisa Q3, que investiga os métodos de avaliação são utilizados para avaliar a colaboração entre estudantes, foi realizada uma análise das abordagens empregadas para esta finalidade nos estudos selecionados.

A Tabela 7 apresenta o agrupamento dos dados coletados com base em similaridades metodológicas, considerando os instrumentos e técnicas descritas nos artigos. Observa-se uma predominância de avaliações baseadas em *surveys* e autoavaliações (1, 2, 8, 9, 14, 16, 17, 21, 24, 25, 28, 32), que frequentemente utilizam escalas estruturadas como a de *Likert*[3]. Além disso, há estudos que adotaram avaliação por pares (3, 5, 18), muitas vezes integradas ao ciclo de desenvolvimento ágil, como parte das entregas em *sprints*.

Outros estudos fundamentaram-se em modelos estruturados de avaliação, como TAM, SAM, MEDIATION e CMMP (2, 4, 15, 23) fornecendo uma base teórica robusta para mensurar a colaboração. Também foram identificadas abordagens centradas na análise de dados oriundos de ferramentas digitais, como *logs* do GitHub, Slack, e-mails e sensores (2, 3, 4, 5, 6, 20, 22, 30, 31), bem como o uso de métricas computacionais específicas, incluindo algoritmos de filtragem colaborativa e sistemas de recomendação (1, 7, 10, 12, 13, 18, 19, 26, 27, 29). Por fim, métodos qualitativos como entrevistas (11, 17, 19) e observação direta (4, 14, 25) foram utilizados para aprofundar a compreensão das dinâmicas colaborativas. Esses agrupamentos evidenciam a diversidade de métodos aplicados e apontam combinação entre avaliação subjetiva (percepção dos estudantes) e evidências objetivas (dados gerados pelas ferramentas).

4.3 Ferramentas e Instrumentos de Avaliação

Para responder à Q4, que investiga quais ferramentas e instrumentos são utilizados na avaliação da colaboração entre estudantes, os estudos foram categorizados com base nas tecnologias mencionadas explicitamente. A Tabela 8 apresenta os grupos identificados.

O grupo mais recorrente considera o uso de formulários online, como Google Forms, Microsoft Forms e LimeSurvey, presentes em 12 dos 32 estudos (1, 2, 3, 4, 14, 16, 18, 21, 24, 28, 30, 32), aplicados em avaliações individuais e em grupo. Em seguida, destacam-se as plataformas colaborativas e ambientes de desenvolvimento, como GitHub, GitLab, Trello, Moodle e Virtual Campus, empregados tanto como espaços de trabalho quanto como fontes para coleta de dados colaborativos (5, 6, 15, 22, 23, 25, 29, 31, 9).

Outros estudos também empregaram ferramentas específicas desenvolvidas para avaliação, como CATME, Team+, MEDIATION, ChatVis, UIDPBL e o Training Program 4.0 (14, 15, 16, 17, 20, 24, 29). Adicionalmente, foram identificados estudos que desenvolveram modelos próprios ou realizaram cálculos específicos para avaliar a colaboração, refletindo abordagens mais experimentais ou exploratórias (10, 11, 12, 26, 27).

Outros estudos fizeram uso de sistemas de rastreamento e bases de dados, como WTS, SQL, Bugzilla, SmartLab e PyDriller, além de abordagens baseadas em Data Analytics (2, 4, 19, 22, 23). Também foram incluídos grupos menores que utilizaram observação direta como método principal (8, 13) e tecnologias físicas (30), como sensores de voz (ReSpeaker), aplicadas em contextos inovadores de avaliação. Esse panorama demonstra a diversidade de estratégias utilizadas e o avanço da instrumentação na avaliação da colaboração em contextos educacionais de Engenharia de Software.

4.4 Desafios para Avaliar a Colaboração

Para responder à questão de pesquisa Q5, que visa identificar quais os principais desafios enfrentados na avaliação da colaboração em ES, foram analisadas as limitações explicitadas nos estudos incluídos na revisão. Dos 32 estudos analisados, 25 mencionaram os desafios enfrentados. Estes foram agrupados por similaridade temática, conforme apresentado na Tabela 9.

A categoria mais recorrente envolve limitações metodológicas e viés na avaliação (4, 6, 7, 8, 9, 10, 12, 13, 17, 20). Os desafios nesse grupo estão relacionados à utilização de instrumentos pouco validados, ausência de triangulação de dados, foco limitado em um único aspecto da colaboração (como comunicação), ou ainda à dependência de ferramentas únicas sem verificação externa. Estudos também relataram dificuldades com a interpretação de métricas desenvolvidas pelos próprios pesquisadores, o que pode comprometer a imparcialidade da avaliação.

Outro grupo identificado inclui trabalhos listando baixo engajamento e a participação reduzida dos estudantes (14, 22, 25, 31). Estes estudos apontam desmotivação, ausência de participação ativa nas atividades colaborativas ou abandono das plataformas digitais utilizadas, afetando diretamente a coleta e a qualidade dos dados.

Também foram identificados estudos relatando cenários com contexto limitado e escopo restrito de aplicação (1, 2, 19, 28, 29), incluindo amostras pequenas, ambientes únicos, participantes que

Tabela 5: Espaço de Execução da Atividade

ID	Atividade Realizada	Parti	Público-Alvo	Metodologia	Objetivo
1	Implementação de PSD, TP e Scrum no desenvolvimento de software	50	Estudantes de graduação	Scrum	Avaliar a eficácia dessas metodologias na melhoria da colaboração em projetos de software
2	Desenvolvimento de software utilizando Machine Learning (ML)	140	Estudantes de graduação	Machine Learning	Investigar como ML pode auxiliar na identificação e processamento de informações relacionadas à efetividade do trabalho em equipe
3	Revisão baseada em survey sobre métodos de atividades em grupo	40	Estudantes de graduação	Survey	Avaliar diferentes métodos de atividades em grupo e sua eficácia na colaboração
4	Aplicação do SETAP para avaliação do trabalho em equipe	380	Estudantes de graduação	SETAP, ML	Avaliar a eficácia do SETAP na análise do trabalho em equipe em projetos de software
5	Investigação de traços de personalidade colaborativa utilizando Slack	252	Estudantes de graduação	Desenvolvimento de software em equipes	Identificar características de personalidade que influenciam a colaboração entre estudantes em ambientes de desenvolvimento de software
6	Utilização do GitHub para suporte no controle de equipes de desenvolvimento	104	Estudantes de graduação	Desenvolvimento de software em equipes, utilizando GitHub	Propor um framework para avaliar o trabalho em equipe com base nas interações no GitHub
7	Revisão sistemática sobre métricas para avaliação do processo Scrum	–	–	Revisão sistemática	Identificar e analisar métricas utilizadas na avaliação de processos Scrum
8	Apresentação do GitHub Classroom como ferramenta de suporte ao desenvolvimento de software	287	Estudantes de graduação	GitHub Classroom em projetos de desenvolvimento de software	Validar a eficácia do GitHub Classroom como ferramenta de apoio ao ensino de desenvolvimento de software
9	Experiência piloto com metodologia de ensino ativa	72	Estudantes de graduação	Experiência piloto	Validar uma metodologia de ensino baseada em aprendizado ativo e sua influência na colaboração
10	Utilização de software para medir colaboração em projetos de codificação	146	Estudantes de graduação	Projeto de codificação	Medir aspectos de codificação e colaboração utilizando a ferramenta Prog4Edu
11	Estudo exploratório com empresas sobre maturidade em engenharia de software	26	Empresas	Estudo exploratório	Compreender o conceito de maturidade em engenharia de software e sua relação com a colaboração
12	Utilização de métricas para mensurar colaboração em projetos de software	19	Estudantes de graduação	Desenvolvimento de software em 3 sprints	Validar métricas de avaliação de colaboração utilizando o framework SEREBRO
13	Experimento para validar processo de ensino híbrido	67	Estudantes de graduação	Processo de ensino híbrido	Validar a eficácia do ensino híbrido, combinando técnicas, com foco na colaboração
14	Implementação de ensino colaborativo em projetos de software	402	Estudantes de graduação	Ensino colaborativo	Avaliar a colaboração e os projetos desenvolvidos utilizando as ferramentas CATME e Team+
15	Pesquisa para expandir o entendimento sobre colaboração em desenvolvimento de software	–	Estudantes de graduação	MEDIATION	Validar o uso do MEDIATION em projetos de desenvolvimento de software
16	Aplicação de Aprendizagem Baseada em Projetos (PBL) no desenvolvimento de software	25	Estudantes de graduação	Project Based Learning	Avaliar os 4Cs (Comunicação, Colaboração, Pensamento Crítico e Criatividade) em um contexto de PBL
17	Bootcamp de inovação para identificar desafios em colaboração	44	Estudantes de graduação	Bootcamp	Identificar dificuldades e métodos de medição da colaboração por meio de bootcamp e entrevistas
18	Estudo sobre composição de equipes com base em métricas	57	Estudantes de graduação	Experimento	Melhorar a composição de equipes utilizando métricas relacionadas à formação e colaboração
19	Análise de casos de sucesso em equipes com base em perfis sociais dos participantes	15	Estudantes de graduação	Entrevistas	Investigar a influência dos perfis sociais dos engenheiros de software no sucesso de projetos
20	Estudo do ChatVis como ferramenta para melhoria da colaboração em projetos de software	33	Estudantes de graduação	ChatVis	Avaliar se o ChatVis, via WhatsApp, promove e mensura a colaboração em projetos de software
21	Aplicação de PBL, Scrum e Kanban para desenvolver soft skills	563	Estudantes de informática do ensino médio	PBL, Scrum	Avaliar se essas metodologias promovem colaboração e o desenvolvimento de outras soft skills
22	Avaliação de ambientes SECO utilizando GitHub	–	Comunidade geral de engenharia de software	Análise do GitHub	Avaliar ambientes SECO por meio do GitHub, focando em aspectos como performance e relações entre desenvolvedores
23	Aplicação de métodos para avaliação inicial da colaboração em metodologias	–	Comunidade geral de engenharia de software	MEDIATION	Avaliar metodologias utilizando o MEDIATION e, posteriormente, o SmartLab
24	Utilização do PBL para avaliação de soft skills	12	Estudantes de graduação	PBL	Avaliar o PBL por meio do Training Program 4.0, focando no desenvolvimento de soft skills
25	Aplicação e avaliação de métodos ágeis	58	Estudantes de graduação	Métodos ágeis: Scrum, Waterfall	Avaliar diferentes métodos ágeis e sua eficácia na colaboração
26	Utilização de métricas de filtragem de colaboração para sua mensuração	–	Estudantes de graduação	Experimento	Desenvolver métricas utilizando Collaborative Filtering (CF) para avaliar e filtrar a colaboração
27	Uso de CF e Sistemas de Recomendação (RS) para promover um modelo computacional de mensuração da colaboração	–	Estudantes de graduação	Análise comparativa	Comparar técnicas para desenvolver um modelo computacional que mensure a colaboração
28	Desenvolvimento de produtos e serviços online com foco na autoavaliação da colaboração	200	Estudantes e engenheiros	Pesquisa survey por e-mail	Compreender como os participantes se autoavaliavam em questões de colaboração por meio de surveys e avaliações em pares
29	Exploração de fatores que afetam a participação individual em projetos de programação em grupo	154	Estudantes de graduação	Desenvolvimento conjunto utilizando GitHub	Investigar como fatores de participação individual influenciam aspectos colaborativos em projetos de programação em grupo
30	Aplicação do Scrum para monitorar a colaboração utilizando microfones em montagem LEGO	16	Estudantes de graduação	Utilização de LEGO e Scrum	Monitorar a colaboração entre alunos utilizando a ferramenta ReSpeaker durante atividades com LEGO
31	Utilização do Moodle para criação de uma wiki sobre medição de colaboração em desenvolvimento de software	–	Estudantes de graduação	Utilização do Moodle	Criar uma wiki para medir a colaboração em projetos de desenvolvimento de software
32	Desenvolvimento de software de médio porte com mentoria em curso de desenvolvimento	–	Estudantes de graduação	Desenvolvimento de software	Mensurar a colaboração e a qualidade do software com base na atividade dos estudantes e na mentoria recebida

Tabela 6: Agrupamento dos estudos por tipo de atividade/metodologia utilizada

Grupo	IDs dos Estudos
Avaliar a colaboração diretamente (medição, percepção, eficácia)	1, 3, 4, 5, 6, 10, 12, 14, 15, 20, 23, 30, 32
Desenvolver ou validar ferramentas de apoio à colaboração	6, 8, 20, 31
Investigar fatores que influenciam a colaboração (perfil, participação, traços)	2, 5, 17, 18, 19, 29
Promover soft skills e ensino ativo	9, 13, 16, 21, 24, 25
Analisar ou propor modelos/metodologias estruturadas de ensino	7, 11, 22, 26, 27, 28

Tabela 7: Métodos de avaliação da colaboração identificados nos estudos

Grupo	IDs dos Estudos
Survey e autoavaliação	1, 2, 8, 9, 14, 16, 17, 21, 24, 25, 28, 32
Avaliação por pares	3, 5, 18
Modelos estruturados (TAM, SAM, MEDIATION, etc.)	2, 4, 15, 23
Observação direta	4, 14, 25
Análise de logs, e-mails e ferramentas	2, 3, 4, 5, 6, 20, 22, 30, 31
Métricas específicas ou computacionais	1, 7, 10, 12, 13, 18, 19, 26, 27, 29
Entrevistas com participantes	11, 17, 19

Tabela 8: Ferramentas e instrumentos utilizados na avaliação da colaboração

Grupo de Ferramentas/Instrumentos	IDs dos Estudos
Formulários online (Google Forms, Microsoft Forms, LimeSurvey)	1, 2, 3, 4, 14, 16, 18, 21, 24, 28, 30, 32
Plataformas colaborativas e ambientes de desenvolvimento (GitHub, GitLab, Trello, Moodle, Virtual Campus)	5, 6, 15, 22, 23, 25, 29, 31, 9
Ferramentas específicas para avaliação de colaboração (CATME, Team+, MEDIATION, Chat-Vis, UIDPBL, Training Program 4.0, BII Questionnaire)	14, 15, 16, 17, 20, 24, 29
Modelos e instrumentos próprios (modelagem conceitual, algoritmos, métodos estatísticos, tabelas personalizadas)	10, 11, 12, 26, 27
Observação direta	8, 13
Bases de dados e sistemas de rastreamento (WTS, SQL, Bugzilla, SmartLab, PyDriller, BDA)	2, 4, 19, 22, 23
Tecnologias de captura física (sensores)	30

já se conheciam, e ausência de variáveis contextuais como a formação das equipes. Essas limitações dificultam a generalização dos resultados para outros cenários educacionais.

Por fim, foram identificados estudos que destacam problemas operacionais e de gestão (5, 11, 24, 30, 32), como a falta de controle sobre canais de comunicação externos, dificuldades logísticas em entrevistas com muitos participantes, custos elevados de ferramentas e problemas na divisão de tarefas entre orientadores. Esses desafios refletem a complexidade da avaliação da colaboração em ambientes

Tabela 9: Agrupamento dos principais desafios enfrentados na avaliação da colaboração

Categoria de Desafio	IDs dos Estudos
Limitações metodológicas e viés na avaliação	4, 6, 7, 8, 9, 10, 12, 13, 17, 20
Baixo engajamento e participação dos estudantes	14, 22, 25, 31
Contexto limitado e escopo restrito da aplicação	1, 2, 19, 28, 29
Problemas operacionais e de gestão na condução da atividade	5, 11, 24, 30, 32

educacionais e reforçam a necessidade de abordagens metodológicas mais robustas, instrumentos validados e maior atenção aos fatores humanos e contextuais nos processos de coleta e análise.

A Seção 5 discute os resultados obtidos com um olhar para a literatura relacionada.

5 Discussões

Esta seção discute os achados da revisão à luz da literatura relacionada, estruturando a análise com base nas questões de pesquisa (Q1 a Q5). As subseções integram os agrupamentos temáticos identificados com evidências empíricas e revisões anteriores sobre avaliação da colaboração no ensino de Engenharia de Software.

5.1 Natureza e propósitos das atividades colaborativas (Q1 e Q2)

Os estudos analisados demonstram uma ampla diversidade de contextos educacionais e propósitos associados às atividades colaborativas. A maior quantidade dos trabalhos foi conduzida no ensino superior, especialmente em cursos de graduação em Engenharia de Software, adotando metodologias ativas como *Project-Based Learning* (PBL), Scrum e Kanban. Esse dado é corroborado por estudos como Quek *et al.* (2023) [44], que aplicam práticas ágeis para fomentar o trabalho em equipe, e Choque-Soto *et al.* (2024) [13], que integram métodos ágeis à promoção de *soft skills*.

As finalidades observadas incluem a avaliação direta da colaboração, o desenvolvimento de ferramentas de apoio, a promoção de competências interpessoais e a investigação de fatores influentes, como perfil dos estudantes e dinâmica de grupo. Trabalhos como os de Maia *et al.* (2023) [28] e Arwatchananukul *et al.* (2021) [6] reforçam a aplicação do PBL com foco no desenvolvimento de habilidades como comunicação, criatividade e colaboração, embora muitas vezes sem instrumentos avaliativos padronizados.

Os resultados obtidos revelam que a ausência de métricas sistematizadas para avaliar a colaboração é um dos principais achados deste estudo. A categorização apresentada neste estudo, torna evidente fragmentação das abordagens utilizadas e a falta de padronização nas práticas avaliativas. Essa constatação reforça a existência de uma lacuna em ES em termos de metodologia de avaliação.

5.2 Métodos de avaliação da colaboração (Q3)

A avaliação da colaboração é majoritariamente baseada em *surveys* e autoavaliações, com destaque para o uso de formulários estruturados, como escalas de Likert. No entanto, esses métodos tendem a capturar percepções subjetivas e, quando utilizados isoladamente,

podem comprometer a robustez dos dados. Essa limitação também é observada em estudos como o de Nelson e Ponciano (2021) [36], que utilizaram questionários para captar percepções dos estudantes e professores sobre ferramentas colaborativas, sem mensurar diretamente os aspectos da colaboração.

Observa-se uma tendência emergente de utilizar fontes mais objetivas, como *logs* de ferramentas (e.g., GitHub, Trello, Slack) e métricas extraídas automaticamente. Estudos como o de Le et al. (2024) [27] e Schmidts et al. (2018) [46] reforçam essa abordagem, conectando práticas de *Learning Analytics* ao ensino de Engenharia de Software. Apesar disso, a adoção de modelos estruturados e validados, como TAM, SAM, MEDIATION ou CMMP, ainda é restrita a poucos estudos, o que revela uma lacuna na padronização e na comparabilidade das práticas avaliativas. Além disso, a ausência de triangulação de métodos (quantitativos e qualitativos) persiste como uma limitação.

5.3 Ferramentas e instrumentos utilizados (Q4)

Formulários online são amplamente utilizados para avaliação por sua simplicidade, aparecendo em 12 dos 32 estudos. Ferramentas como Google Forms e Microsoft Forms continuam sendo a base para *surveys* e autoavaliações. No entanto, esse predomínio reflete certa dependência de instrumentos autorrelatados, com menor presença de ferramentas estruturadas.

Ambientes colaborativos como GitHub, GitLab, Moodle e Trello têm sido utilizados tanto como espaços de trabalho quanto como fontes de coleta de dados. Estudos como os de Nelson e Ponciano (2021) [36] e Schmidts et al. (2018) [46] indicam que a análise de interações em plataformas digitais oferece maior granularidade e objetividade à avaliação da colaboração.

Além disso, alguns trabalhos aplicaram ferramentas específicas como CATME, ChatVis e Training Program 4.0 para coletar dados sobre engajamento e desempenho colaborativo. Contudo, a adoção dessas ferramentas ainda é incipiente e, em muitos casos, os instrumentos utilizados são desenvolvidos pelos próprios autores, o que levanta preocupações sobre validade e reusabilidade. Ressalta-se ainda o uso inovador de sensores, como no estudo com ReSpeaker (Cornide-Reyes et al., 2019) [16], embora questões de custo e infraestrutura ainda limitem sua disseminação.

5.4 Desafios enfrentados na avaliação da colaboração (Q5)

Os principais desafios enfrentados nos estudos analisados por esta revisão rápida apontam para limitações metodológicas, como o uso de instrumentos não validados, ausência de triangulação e foco restrito em aspectos da colaboração, como a comunicação. Tais dificuldades são consistentes com o que Hidalgo et al. (2023) [20] apontam em sua revisão rápida sobre ensino com dramatizações, destacando a carência de mecanismos claros de avaliação colaborativa.

Também foram identificados desafios relacionados ao engajamento dos estudantes, escopo limitado dos estudos e problemas de logística de aplicação e divisão de responsabilidades entre orientadores. Esses achados indicam que avaliar a colaboração exige não apenas métodos adequados, mas também um ambiente educacional estruturado, clareza de objetivos e suporte institucional.

Em síntese, os estudos analisados confirmam o papel central da colaboração no ensino de Engenharia de Software e revelam que há múltiplas abordagens. A literatura aponta que apesar dos avanços na integração de metodologias ativas e ferramentas digitais, persistem lacunas na forma como a colaboração é mensurada e analisada. A ausência de modelos validados e a diversidade de estratégias dificultam a comparação entre estudos e comprometem a replicabilidade. Por isso, ainda é perceptível a dificuldade em consolidar boas práticas de avaliação da colaboração, especialmente no que diz respeito aos processos envolvidos, e não apenas aos produtos gerados. Há, portanto, uma oportunidade para o desenvolvimento de instrumentos baseados em evidências e alinhados às práticas educacionais que tem sido adotadas.

6 Ameaças à Validade

Por sua natureza, uma revisão rápida está sujeita a ameaças à validade, as quais foram consideradas ao longo do processo metodológico. No que se refere à validade da conclusão, as inferências podem ter sido influenciadas por julgamentos subjetivos durante os agrupamentos temáticos e a categorização dos estudos. Para mitigar essa limitação, os agrupamentos foram discutidos iterativamente entre os autores até o consenso.

Quanto à validade interna, destaca-se a heterogeneidade metodológica dos estudos incluídos, o que pode ter afetado a consistência das interpretações. Como se trata de uma revisão da literatura, não há controle direto sobre variáveis, e as conclusões dependem da clareza e completude dos dados relatados nos próprios artigos.

No que diz respeito à validade de construção, a terminologia utilizada nos estudos para descrever colaboração e seus métodos de avaliação apresentou variações. Tal diversidade pode dificultar a uniformização dos conceitos e comprometer a comparabilidade entre abordagens. Para minimizar esse risco, a formulação da estratégia de busca foi guiada pelo *framework* PICOC, visando assegurar alinhamento conceitual com o objetivo do estudo.

A validade externa também representa uma limitação, uma vez que a revisão foi restrita a duas bases de dados (IEEE Xplore e Scopus) e com recorte temporal. Dessa forma, é possível que estudos relevantes em outras bases ou idiomas tenham sido excluídos, comprometendo a abrangência e generalização dos achados.

7 Considerações Finais

Este estudo investigou como a colaboração é promovida, avaliada e quais desafios são enfrentados em atividades de cocriação no ensino de Engenharia de Software. Por meio de uma revisão rápida da literatura, foram analisados 32 estudos publicados entre 2014 e 2025, com foco nas atividades educacionais, nos métodos e instrumentos de avaliação, e nos principais obstáculos relatados.

Os resultados mostram que, embora a colaboração seja reconhecida como uma habilidade importante na formação de engenheiros de software, ainda existem limitações na forma como ela é avaliada. A maioria dos estudos utiliza *surveys* e autoavaliações, muitas vezes sem combinar diferentes fontes de dados e com instrumentos criados pelos próprios autores. Mesmo com algumas iniciativas usando registros de ferramentas colaborativas ou métricas computacionais, o uso de modelos estruturados de avaliação ainda é exceção.

Além disso, os achados evidenciam que avaliar a colaboração é uma tarefa complexa e realizada por abordagens fragmentadas. Não há padronização de métricas ou modelos adotados, o que dificulta a consolidação de boas práticas. Entre os principais desafios relatados estão falhas metodológicas, baixo engajamento dos estudantes, escopo limitado das iniciativas e dificuldades operacionais. Esses fatores indicam que a avaliação da colaboração vai além da escolha dos instrumentos, ou seja, exige um planejamento estruturado, metodologias adequadas e suporte institucional.

Portanto, com base nos dados analisados, destaca-se que uma proposta de avaliação integrada deve considerar ao menos três eixos: (i) o processo colaborativo (como interações, tomada de decisão e engajamento dos participantes); (ii) o uso combinado de instrumentos (e.g., surveys, logs, observações); e (iii) a triangulação de dados quantitativos e qualitativos.

As principais contribuições do trabalho incluem: (i) de forma geral, uma revisão das práticas usadas para avaliar a colaboração; (ii) a categorização das atividades educacionais e seus contextos; (iii) o mapeamento dos métodos, instrumentos e ferramentas aplicadas na avaliação; e (iv) a identificação dos principais desafios enfrentados, contribuindo para o aprimoramento das práticas avaliativas e futuras pesquisas na área.

Por fim, como trabalhos futuros, destacamos a importância de: (i) validar instrumentos de avaliação da colaboração em diferentes contextos; (ii) identificar e organizar categorias de métricas utilizadas na avaliação da colaboração, visando a construção de repositórios compartilhados com boas práticas; (iii) realizar estudos longitudinais que investiguem o impacto dessas avaliações no desempenho acadêmico e no desenvolvimento de *soft skills*; e (iv) ampliar o escopo da revisão para outras bases de estudos e literatura cinzenta, buscando maior abrangência e diversidade de abordagens.

DISPONIBILIDADE DE ARTEFATO

Todos os artefatos produzidos durante a revisão, incluindo os dados extraídos, critérios de seleção, planilhas de categorização e o protocolo completo, estão disponíveis no repositório Zenodo em <https://doi.org/10.5281/zenodo.15427829> para consulta e reuso.

AGRADECIMENTOS

Agradecemos à FAPERGS (Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul) pelo apoio financeiro concedido a Enzo Kleinpaul, o que possibilitou sua contribuição significativa para o desenvolvimento e a execução deste estudo.

REFERÊNCIAS

- [1] Dariane Abich, Rafael Parizi, and Sabrina Marczak. 2024. Fostering Collaboration through Design Thinking: A Study among Software Engineering Students. In *Simpósio Brasileiro de Engenharia de Software* (Curitiba/PR). SBC, 532–542.
- [2] Jörg Abke, Carolin Gold, Nina Kälberer, and Martina Kuhn. 2014. Collaborative knowledge transfer via Wiki: A project based learning approach in software engineering. In *Int'l Conf. on Interactive Collaborative Learning (ICL)*. IEEE, 283–288.
- [3] Gerald Albaum. 1997. The Likert Scale Revisited. *Market Research Society Journal* 39, 2 (Feb 1997), 1–21.
- [4] Pedro Arantes, Felipe Soupinski, and Awdren Fontão. 2023. Social networks during software ecosystems' death. In *International Workshop on Software Engineering for Systems-of-Systems and Software Ecosystems (SESOS)*. IEEE, 9–12.
- [5] Sujitra Arwathananukul, Prasara Jakkaew, Wacharawan Intayoad, and Nattapol Aunsri. 2021. A Case Study in Class User Interface Design of Problem-Based Learning Modeling (UIDPBL). In *2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering*. IEEE, 364–367.
- [6] Sujitra Arwathananukul, Prasara Jakkaew, Wacharawan Intayoad, and Nattapol Aunsri. 2021. A case study in class user interface design of problem-based learning modeling (uidpbl). In *2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering*. IEEE, 364–367.
- [7] Andreas Bäcke, Erik Tholén, and Lucas Gren. 2019. Social identity in software development. In *2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*. IEEE, 107–114.
- [8] Alper Bilge and Cihan Kaleli. 2014. A multi-criteria item-based collaborative filtering framework. In *2014 11th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. IEEE, 18–22.
- [9] Mohammad Rizqullah Hafizh bin Mohamed Riduwan, Mark Peng Jung Zen, Oh Jia Wei Darien, Ng Jun Hong, Chloe Rayn Loh, Ong Shi Ya, Qi Cao, and Peter Chunyu Yau. 2024. Exploring Peer Evaluation Methods in Group Projects of Software Engineering Education. In *2024 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–6.
- [10] Luiz Fernando Capretz and Faheem Ahmed. 2018. A call to promote soft skills in software engineering. *arXiv preprint arXiv:1901.01819* (2018).
- [11] Bruno Cartaxo, Gustavo Pinto, and Sergio Soares. 2020. Rapid reviews in software engineering. *Contemporary Empirical Methods in Software Engineering* (2020), 357–384.
- [12] Victória Cavalcante, Catarina Costa, and Daricélio Soares. 2024. Beyond Code: the Development of soft skills through Training in Software Engineering. In *Anais do XXXVIII Simpósio Brasileiro de Engenharia de Software* (Curitiba/PR). SBC, Porto Alegre, RS, Brasil, 521–531. doi:10.5753/sbes.2024.3541
- [13] Vanessa Maribel Choque-Soto and Victor Dario Sosa-Jauregui. 2024. Assessing Soft Skills Development in Informatics Students Through Project-Based Learning and Agile Frameworks. In *2024 International Symposium on Accreditation of Engineering and Computing Education (ICACIT)*. IEEE, 1–8.
- [14] Shuddha Chowdhury, Charles Walter, and Rose Gamble. 2018. Toward increasing collaboration awareness in software engineering teams. In *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–9.
- [15] Orges Cico, Letizia Jaccheri, and Anh Nguyen Duc. 2021. Facilitating learning and startup formation in experience-based courses—A team-centered model. In *2021 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 599–608.
- [16] Hector Cornide-Reyes, René Noël, Fabián Riquelme, Matias Gajardo, Cristian Cechinel, Roberto Mac Lean, Carlos Becerra, Rodolfo Villarroel, and Roberto Munoz. 2019. Introducing low-cost sensors into the classroom settings: Improving the assessment in agile practices with multimodal learning analytics. *Sensors* 19, 15 (2019), 3291.
- [17] Carla M Evans. 2020. Measuring Student Success Skills: A Review of the Literature on Collaboration. 21st Century Success Skills. *National Center for the Improvement of Educational Assessment* (2020).
- [18] Alicia García-Holgado, Francisco J García-Peñalvo, and Maria José Rodríguez-Conde. 2018. Pilot experience applying an active learning methodology in a Software Engineering classroom. In *2018 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 940–947.
- [19] Mohammed Hassan and Mohamed Hamada. 2017. A computational model for improving the accuracy of multi-criteria recommender systems. In *2017 IEEE 11th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoc)*. IEEE, 114–119.
- [20] Mauricio Hidalgo, Hernán Astudillo, and Laura M Castro. 2023. Challenges to use role playing in software engineering education: A rapid review. In *International Conference on Applied Informatics*. Springer, 245–260.
- [21] Claudia Jacob and Shamal Faily. 2020. The impact of undergraduate mentorship on student satisfaction and engagement, teamwork performance, and team dysfunction in a software engineering group project. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 128–134.
- [22] Amber Kempainen, Amy Hamlin, Matthew Barron, and Mary Raber. 2022. Comparing Team Evaluation Software (Team+ and CATME). In *2022 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–6.
- [23] Barbara Kitchenham, Stuart Charters, et al. 2007. Guidelines for performing systematic literature reviews in software engineering.
- [24] Reni Kurnia, Ridi Ferdiana, and Sunu Wibirama. 2018. Software metrics classification for agile scrum process: A literature review. In *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. IEEE, 174–179.
- [25] Emily Lai, Kristen DiCerio, and Peter Foltz. 2017. Skills for Today: What We Know about Teaching and Assessing Collaboration. *Pearson* (2017).
- [26] J Richard Landis and Gary G Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics* 33 (Feb 1977), 159–174.
- [27] Quan Le, Kiet Phan, Bowen Hui, and Adara Putri. 2024. Using GitHub Analytics to Assess the Quality of Collaboration in Software Engineering Teams. In *2024 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–9.
- [28] Davi Maia, Simone C dos Santos, Guimel FG Cavalcante, and Pedro AA Falcão. 2023. Managing Soft Skills Development in Technological Innovation Project Teams: An Experience Report in the Automotive Industry. In *2023 IEEE Frontiers*

- in *Education Conference (FIE)*. IEEE, 1–9.
- [29] Allen Marshall and Rose Gamble. 2015. Gauging influence in software development teams. In *2015 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–8.
- [30] George Marsicano, Diana Valença Pereira, Fabio QB da Silva, and César França. 2017. Team maturity in software engineering teams. In *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, 235–240.
- [31] João Mattar and Daniela Karine Ramos. 2021. *Metodologia da pesquisa em educação: abordagens qualitativas, quantitativas e mistas*. Almedina Brasil.
- [32] Gerardo Maturro, Florencia Raschetti, and Carina Fontán. 2019. A systematic mapping study on soft skills in software engineering. *J. Univers. Comput. Sci.* 25, 1 (2019), 16–41.
- [33] José P Miguel, David Mauricio, and Glen Rodríguez. 2014. A review of software quality models for the evaluation of software products. *arXiv preprint arXiv:1412.2977* (2014).
- [34] Alexander Mitchell, Terry Greer, Warwick New, Joseph Walton-Rivers, Matt Watkins, Douglas Brown, and Michael James Scott. 2021. Student Perspectives on the Purpose of Peer Evaluation During Group Game Development Projects. In *Proceedings of the 2021 Conference on United Kingdom & Ireland Computing Education Research*. 1–7.
- [35] Joydeep Mitra and Eric Gerber. 2025. Examining Teamwork: Evaluating Individual Contributions in Collaborative Software Engineering Projects. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*. 777–783.
- [36] Maria Augusta Nelson and Lesandro Ponciano. 2021. Experiences and insights from using Github Classroom to support Project-Based Courses. In *2021 Third International Workshop on Software Engineering Education for the Next Generation (SEENG)*. IEEE, 31–35.
- [37] Bao-An Nguyen, Kuan-Yu Ho, and Hsi-Min Chen. 2020. Measure students' contribution in web programming projects by exploring source code repository. In *International Computer Symposium (ICS)*. IEEE, 473–478.
- [38] Sofia Ouhbi and Nuno Pombo. 2020. Software Engineering Education: Challenges and Perspectives. In *2020 IEEE Global Engineering Education Conference (EDUCON)*. 202–209. doi:10.1109/EDUCON45650.2020.9125353
- [39] Rafael Parizi, Matheus Prestes, Sabrina Marczak, and Tayana Conte. 2022. How has design thinking being used and integrated into software development activities? A systematic mapping. *Journal of Systems and Software* 187 (2022), 111217.
- [40] Héctor G Pérez-González, Raymundo A González-Grimaldo, Joshua Chibuike Nwokeji, Mei-Huei Tang, Reyes Juárez-Ramírez, Davide Piovesan, Andrew Meneely, and César Guerra-García. 2024. WIP: ChatVis: Enhancing Academic Team Collaboration through WhatsApp Chat Analytics. In *2024 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–5.
- [41] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. 2015. Guidelines for Conducting Systematic Mapping Studies in Software Engineering: An Update. *Information and Software Technology* 64 (2015), 1–18.
- [42] Dragutin Petkovic, Marc Sosnick-Pérez, Shihong Huang, Rainer Todtenhoefer, Kazunori Okada, Swati Arora, Ramasubramanian Sreenivasen, Lorenzo Flores, and Sonai Dubey. 2014. Setap: Software engineering teamwork assessment and prediction using machine learning. In *2014 IEEE frontiers in education conference (FIE) proceedings*. IEEE, 1–8.
- [43] Dragutin Petkovic, Marc Sosnick-Pérez, Kazunori Okada, Rainer Todtenhoefer, Shihong Huang, Nidhi Miglani, and Arthur Vigil. 2016. Using the random forest classifier to assess and predict student learning of software engineering teamwork. In *2016 IEEE frontiers in education conference (FIE)*. IEEE, 1–7.
- [44] Ser Wee Darren Quek, Jared Kai Fu Teo, Nasruddine Louahemmsabah, Yap Boon Cyrus Tong, Aaron Zheng Rong Poh, Qi Cao, Chee Kiat Seow, Peter Chunyu Yau, and Alex Qiang Chen. 2023. Improving teamwork in software engineering projects in higher education. In *2023 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–8.
- [45] Eric D. Ragan, Stephen Frezza, and Jeremy Cannell. 2009. Product-based learning in software engineering education. In *2009 39th IEEE Frontiers in Education Conference*. 1–6. doi:10.1109/FIE.2009.5350648
- [46] Oliver Schmidt, Bodo Kraft, Marc Schreiber, and Albert Zündorf. 2018. Continuously evaluated research projects in collaborative decoupled environments. In *Proceedings of the 5th International Workshop on Software Engineering Research and Industrial Practice*. 2–9.
- [47] Marc Schreiber, Bodo Kraft, and Albert Zündorf. 2017. Metrics driven research collaboration: Focusing on common project goals continuously. In *2017 IEEE/ACM 4th International Workshop on Software Engineering Research and Industrial Practice (SER&IP)*. IEEE, 41–47.
- [48] Maurício Souza, Renata Moreira, and Eduardo Figueiredo. 2019. Students perception on the use of project-based learning in software engineering education. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering*. 537–546.
- [49] Carlos Eduardo Stefani and Marcelo Duduchi. 2023. Elementos de colaboração nos métodos ágeis de desenvolvimento de software. In *Anais do XVIII Simpósio Brasileiro de Sistemas Colaborativos* (Rio de Janeiro/RJ). SBC, Porto Alegre, RS, Brasil, 86–100. doi:10.5753/sbsc.2023.229086
- [50] Laura Suzanne Tribble. 2009. *The Importance of Soft Skills in the Workplace as Perceived by Community College Instructors and Industries*. Ph.D. Dissertation. Mississippi State University, Mississippi, USA. <https://scholarjunction.msstate.edu/td/4529/> Accessed: 2025-05-13.
- [51] Ranjith Unnikrishnan and Martial Hebert. 2005. Measures of similarity. In *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05)-Volume 1*, Vol. 1. IEEE, 394–394.
- [52] Vladimir Vujović, Mirjana Maksimović, and Branko Perišić. 2014. Collaboration in software engineering classroom. In *2014 IEEE 12th IEEE International Conference on Emerging eLearning Technologies and Applications (ICETA)*. 505–510. doi:10.1109/ICETA.2014.7107636
- [53] Rafał Włodarski, Jean-Remy Falleri, and Corinne Parvéry. 2021. Assessment of a hybrid software development process for student projects: a controlled experiment. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*. IEEE, 289–299.
- [54] Rafał Włodarski, Aneta Poniszewska-Marañda, and Jean-Remy Falleri. 2020. Comparative case study of plan-driven and agile approaches in student computing projects. In *2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, 1–6.
- [55] Umi Laili Yuhana, Umi Sa'Adah, Chandra Kirana Jatu Indraswari, Siti Rochimah, and Maulidan Bagus Afridian Rasyid. 2022. Classifying Composition of Software Development Team Using Machine Learning Techniques. In *2022 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*. IEEE, 122–127.