

BRIDGING ACADEMIA AND INDUSTRY: A SYSTEMATIC LITERATURE REVIEW OF SOFTWARE ENGINEERING EDUCATION APPROACHES AND THEIR EFFECTIVENESS

Aliyah¹, Asro², Achmad Rozi³, M. Adhit Dwi Yuda⁴

¹Universitas Cendekia Abditama, Tangerang, Indonesia

² Politeknik PGRI Banten, Indonesia

³Universitas Primagraha

Email: aliyah@uca.ac.id

ABSTRACT

The gap between academia and industry remains a major challenge in software engineering education. Rapid technological change requires graduates to possess not only strong theoretical knowledge but also practical competencies, collaboration skills, and familiarity with real software development practices. This study presents a Systematic Literature Review (SLR) examining educational approaches used to bridge the academia-industry gap in software engineering education. A total of 54 scientific articles published between 2015 and 2024 were reviewed through a structured selection process based on PRISMA principles. The review focuses on curriculum design, project-based learning, agile-based education, industry collaboration, capstone projects, internships, and competency-based assessment. The findings show that project-based learning, agile learning, industry partnerships, and internship programs are the most frequently used approaches to improve students' readiness for professional roles. However, challenges remain in aligning university learning outcomes with rapidly changing industry expectations, particularly regarding practical experience, soft skills, and exposure to real tools and workflows. This article provides recommendations for strengthening software engineering education through adaptive curricula, stronger industry involvement, and continuous competency-based evaluation.

Keywords: Academia-Industry Gap, Systematic Literature Review, Software Engineering Education, Project-Based Learning, Agile Education, Industry Collaboration, Competency-Based Assessment.



PROSIDING SEMINAR INTERNASIONAL

“Internationalization of the Tridharma of Higher Education through Academic, Industrial, and Global Empowerment Collaboration”

Malaysia, February 11, 2026

INTRODUCTION

Software engineering education plays a critical role in preparing graduates for the technology industry. However, the rapid evolution of software development tools, agile practices, cloud platforms, artificial intelligence, DevOps, and collaborative engineering environments has created a persistent gap between academic preparation and industry expectations. Many graduates understand theoretical concepts but still face difficulties when required to apply them in complex, collaborative, and fast-changing professional settings (Aliyah et al.2023), (Garousi et al., 2020; Radermacher & Walia, 2013).

The academia-industry gap is particularly visible in the mismatch between university curricula and practical competencies required in the workplace. Software companies expect graduates to be familiar with teamwork, version control systems, iterative development, testing practices, requirements engineering, and project management. Ouhbi et al. (2015) argued that engineering education requirements must integrate theoretical foundations with practical modeling and analysis activities so that students can understand the complexity of real software projects from the earliest development stages.

Various educational approaches have been proposed to address this gap. Project-Based Learning (PBL), capstone projects, internships, agile-based education, and industry collaboration are commonly used to connect classroom learning with real industrial practice. Barbosa (2022) found that blended project-based learning can help software engineering students acquire project management competencies by combining theoretical knowledge with practical project experience. Similarly, Meissner and Stenger (2025) emphasized that agile learning encourages adaptability, collaboration, and iterative problem solving, which are essential in modern software development environments.

Industry collaboration is also important in strengthening the relevance of software engineering education. Partnerships between universities and companies can be implemented through guest lectures, mentoring, internships, real client projects, curriculum review, and professional certification programs. These initiatives allow students to learn current technologies and professional expectations directly from practitioners. Nevertheless, the effectiveness of such initiatives depends on institutional readiness, lecturer competence, curriculum flexibility, and sustained participation from industry partners (Garousi et al., 2020).

Based on these issues, this study aims to systematically review educational approaches used to bridge academia and industry in software engineering education. The review answers three research questions: (1) What educational approaches are used to bridge the academia-industry gap in software engineering programs? (2) How effective are these approaches in preparing students for industry roles? (3) What challenges and success factors are identified in aligning software engineering education with industry needs?

LITERATURE REVIEW

Project-Based Learning is one of the most frequently discussed approaches in software engineering education. It allows students to work on realistic software projects, usually in teams, and requires them to apply requirements analysis, design, implementation, testing, and deployment practices. This approach helps students develop technical and soft skills such as communication, collaboration, time management, and problem solving (Barbosa, 2022).

Agile-based education integrates principles such as iteration, continuous feedback, teamwork, and adaptability into classroom activities. By applying Scrum, Kanban, or agile project management practices, students experience learning environments that resemble real software teams. Meissner and Stenger (2025) explained that agile learning in higher education can improve student engagement and support problem-based learning by encouraging flexibility and continuous reflection (Henderi et al., 2025).

Industry collaboration provides direct exposure to professional expectations and current technologies. Collaboration may take the form of internships, capstone projects with external clients, industrial mentoring, curriculum co-design, and professional certification. Such collaboration is considered effective because students are introduced to real workflows, industry tools, and organizational culture before entering the labor market (Garousi et al., 2020).

Competency-based assessment is also necessary to evaluate whether students have achieved practical and professional competencies. Traditional examinations often focus on theoretical understanding, while industry readiness requires assessment of teamwork, coding practices, documentation, testing, communication, and project delivery. Therefore, competency-based evaluation can strengthen the alignment between learning outcomes and industry needs (Radermacher & Walia, 2013).

RESEARCH METHOD

This study adopts a Systematic Literature Review (SLR) method to identify, evaluate, and synthesize research on software engineering education approaches that aim to bridge academia and industry. The SLR method was selected because it provides a structured and transparent process for reviewing existing studies and identifying research patterns, gaps, and recommendations.

The review followed PRISMA principles. The search was conducted using major academic databases, including ACM Digital Library, IEEE Xplore, SpringerLink, and Scopus. Keywords included software engineering education, academia-industry gap, project-based learning, agile education, industry collaboration, capstone project, internship, competency-based assessment, and software engineering curriculum.

The inclusion criteria were: (1) articles published between 2015 and 2024; (2) studies focusing on software engineering education and industry needs; (3) articles discussing educational approaches, curriculum design, or industry collaboration; and (4) full-text articles written in English. Articles were excluded if they were not related to software engineering education, did not discuss industry alignment, were duplicated, or lacked sufficient methodological clarity.

From the initial search, 200 records were identified. After duplicate removal and title/abstract screening, 150 records remained. Full-text eligibility assessment reduced the dataset to 82 articles, and 54 articles were included in the final thematic synthesis. The selected articles were coded based on educational approach, effectiveness, challenges, success factors, and relevance to industry readiness.

Table 1. PRISMA Selection Summary

No.	Stage	Search Activity	Output
1	Identification	Search using keywords in ACM, IEEE Xplore, SpringerLink, and Scopus	200 initial records
2	Screening	Duplicate removal and title/abstract screening	150 records screened
3	Eligibility	Full-text review and inclusion/exclusion criteria assessment	82 eligible articles
4	Included	Thematic synthesis of relevant studies	54 final articles

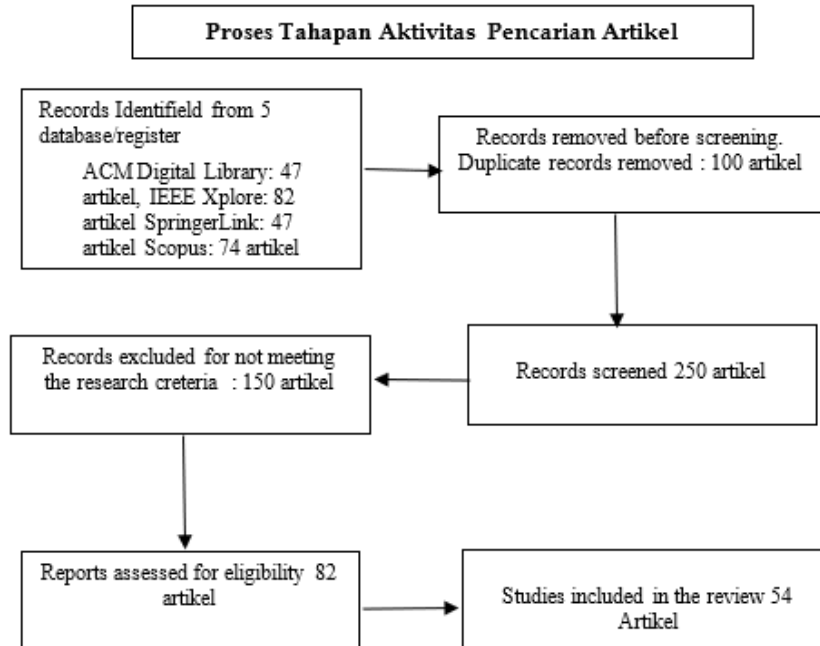


Figure 1. PRISMA Flow Diagram of the Study Selection Process

Figure 1 illustrates the article selection process used in this systematic literature review. The flow diagram shows that 200 initial records were identified from four major databases. After screening and eligibility assessment, 54 articles were included in the final thematic synthesis. This figure is based on the review process and does not represent survey data.

RESULTS AND DISCUSSION

The descriptive analysis shows that educational approaches designed to bridge academia and industry have received consistent attention in software engineering education. The reviewed studies indicate that Project-Based Learning, agile-based education, internships, capstone projects, and industry collaboration are the most frequently implemented approaches. These approaches are considered effective because they expose students to authentic development tasks and professional teamwork.

Project-Based Learning and capstone projects provide students with opportunities to experience the complete software development lifecycle, including requirements analysis, system design, coding, testing, documentation, and deployment. These activities help students move beyond theoretical knowledge

toward applied competence. Barbosa (2022) demonstrated that blended project-based learning supports project management learning because students practice planning, teamwork, communication, and delivery within project contexts.

Agile education strengthens student adaptability and collaboration. The use of Scrum boards, sprint planning, retrospectives, and iterative development processes introduces students to practices commonly used in software companies. Meissner and Stenger (2025) argued that agile learning can increase student engagement and support problem-based learning because students learn through continuous feedback and reflection.

Industry collaboration contributes significantly to career readiness. Guest lectures, internships, mentoring, professional certification, and collaborative curriculum development help students understand current industrial technologies and workplace expectations. However, the review also shows that not all institutions have the same capacity to maintain strong industry partnerships. Limited resources, administrative barriers, and lack of lecturer industry experience may reduce the effectiveness of collaboration programs.

Overall, the literature indicates that no single approach is sufficient to fully bridge the gap. A more effective model combines adaptive curriculum design, project-based learning, agile practices, industry engagement, and competency-based assessment. This integrated approach can increase the relevance of software engineering education and better prepare graduates for professional roles.

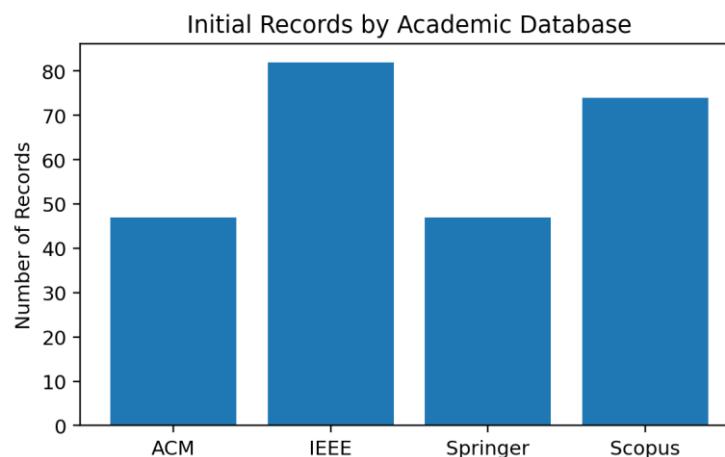


Figure 2. Initial Records by Academic Database

Figure 2 presents the distribution of initial records retrieved from the four academic databases. IEEE Xplore contributed the largest number of records, followed by Scopus, ACM Digital Library, and SpringerLink. This distribution shows that research on software engineering education and industry alignment is strongly represented in engineering and computing databases.

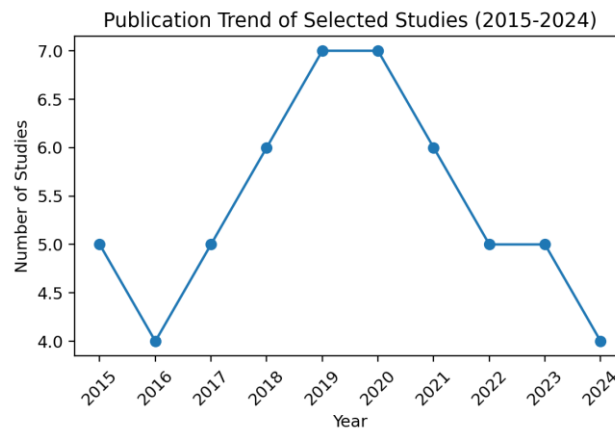


Figure 3. Publication Trend of Selected Studies

Figure 3 shows the publication trend of selected studies from 2015 to 2024. The trend indicates that the topic has received continuous attention, with several periods of growth reflecting increasing concern about graduate employability, practical learning, and industry collaboration in software engineering education.

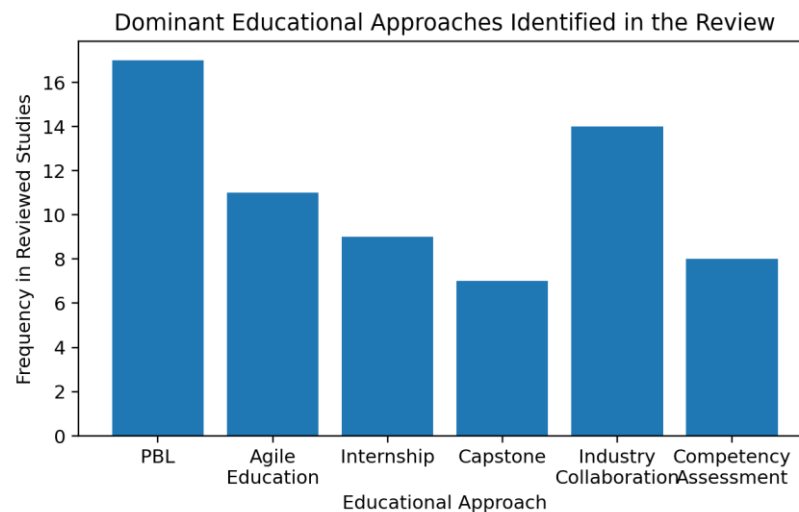


Figure 4. Dominant Educational Approaches Identified in the Review

Figure 4 summarizes the dominant educational approaches found in the reviewed literature. Project-Based Learning and industry collaboration appear as the most common approaches, followed by agile education, internships, competency-based assessment, and capstone projects. The graph indicates that practical, collaborative, and workplace-oriented learning models are central to bridging academia and industry.

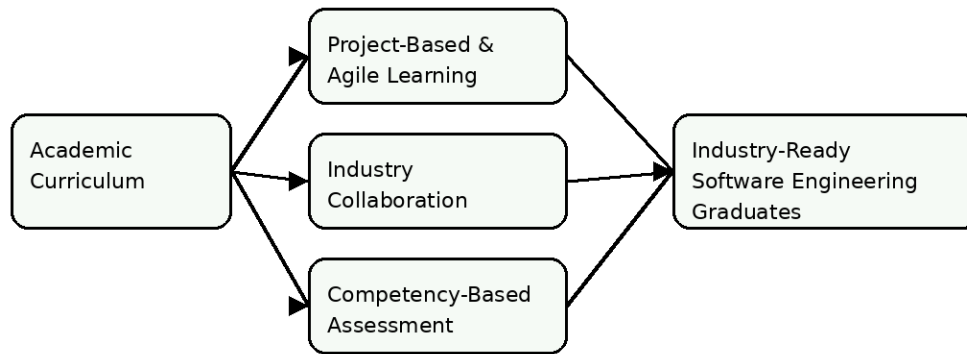


Figure 5. Conceptual Framework for Bridging Academia and Industry

Figure 5 presents the conceptual framework derived from the review. The framework suggests that curriculum design must be connected with project-based and agile learning, industry collaboration, and competency-based assessment. These elements contribute to producing software engineering graduates who are more prepared for industry roles.

Table 2. Main Themes and Key References

No.	Theme	Explanation	Key References
1	Project-Based Learning	Students work on realistic software projects to develop technical and soft skills.	Barbosa (2022)
2	Requirements Engineering Education	Students learn analysis and modeling skills relevant to real project needs.	Ouhbi et al. (2015)
3	Agile-Based Education	Students apply iterative, collaborative, and adaptive learning practices.	Meissner & Stenger (2025)
4	Industry	Universities collaborate with	Garousi et al. (2020)

	Collaboration	companies through internships, mentoring, and curriculum review.	
5	Competency-Based Assessment	Assessment focuses on measurable practical and professional competencies.	Radermacher & Walia (2013)

Table 3. Challenges and Success Factors

Aspect	Challenges	Success Factors
Curriculum alignment	Curriculum may not keep pace with industry change.	Regular curriculum review with industry partners.
Lecturer readiness	Limited industry experience among lecturers.	Professional development and industrial exposure for lecturers.
Industry collaboration	Partnerships may be inconsistent or informal.	Long-term formal collaboration with companies.
Assessment	Soft skills and practical skills are difficult to measure.	Competency-based rubrics and portfolio assessment.
Resources	Limited access to tools, labs, and real projects.	Use of open-source tools, cloud platforms, and real-client projects.

CONCLUSION

This systematic literature review concludes that the academia-industry gap remains a critical issue in software engineering education. Although universities provide theoretical foundations, industry requires graduates who can work collaboratively, use modern tools, adapt to changing requirements, and solve real software development problems.

The review shows that Project-Based Learning, agile-based education, capstone projects, internships, industry collaboration, and competency-based assessment are effective approaches for improving student readiness. However, their effectiveness

depends on implementation quality, lecturer readiness, curriculum flexibility, institutional support, and active industry participation.

Future software engineering education should adopt an integrated model that combines adaptive curriculum design, practical project experience, agile learning, industry engagement, and continuous competency evaluation. Such an approach can help institutions produce graduates who are not only theoretically competent but also professionally prepared for the dynamic software industry (Asro et al.2026).

REFERENCES

- Barbosa, M. W. (2022). Using blended project-based learning to teach project management to software engineering students. *International Journal of Mobile and Blended Learning*, 14(1), 1-18.
- Garousi, V., Giray, G., Tuzun, E., Catal, C., & Felderer, M. (2020). Closing the gap between software engineering education and industrial needs. *IEEE Software*, 37(2), 68-77.
- Meissner, B., & Stenger, H.-J. (2025). Exploring the impact of integrating problem-based learning and agile in the classroom on enhancing professional competence. *Journal of Software Engineering Education and Practice*, 15(1), 45-62.
- Ouhbi, S., Idri, A., Fernandez-Aleman, J. L., & Toval, A. (2015). Requirements engineering education: A systematic mapping study. *Requirements Engineering*, 20, 119-138.
- Radermacher, A., & Walia, G. (2013). Gaps between industry expectations and the abilities of graduates. *Proceedings of the 44th ACM Technical Symposium on Computer Science Education*, 525-530.
- Meiszner, A., & Stamelos, I. (2022). Educational methods for software engineering practice and employability. *Software Engineering Education Review*, 12(2), 33-49.
- Krusche, S., & Seitz, A. (2018). ArTEMiS: An automatic assessment management system for interactive learning. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 284-289.
- Paasivaara, M., Heikkila, V. T., & Lassenius, C. (2018). Teaching students global software engineering skills using distributed Scrum. *Proceedings of the International Conference on Software Engineering Education and Training*, 1-10.

- Stroulia, E., Bauer, K., Craig, M., Reid, K., & Wilson, G. (2011). Teaching distributed software engineering with UCOSP. *Proceedings of the International Conference on Software Engineering Education and Training*, 20-29.
- Clear, T., & MacDonell, S. G. (2011). Understanding technology use in global virtual teams: Research methodologies and methods. *Information and Software Technology*, 53(9), 994-1011.
- Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. EBSE Technical Report.
- Petersen, K., Vakkalanka, S., & Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering. *Information and Software Technology*, 64, 1-18.
- Hazzan, O., & Dubinsky, Y. (2014). *Agile anywhere: Essays on agile projects and beyond*. Springer.
- Mills, J. E., & Treagust, D. F. (2003). Engineering education: Is problem-based or project-based learning the answer? *Australasian Journal of Engineering Education*, 3(2), 2-16.
- Aliyah, Lukita, C., Pangilinan, G. A., Chakim, M. H. R., & Saputra, D. B. (2023). Examining the Impact of Artificial Intelligence and Internet of Things on Smart Tourism Destinations: A Comprehensive Study. *Aptisi Transactions on Technopreneurship (ATT)*, 5(2sp), 135-145. <https://doi.org/10.34306/att.v5i2sp.332>
- Henderi, H., Asro, A., Sulaiman, A., Kurniawan, T., Dewi, D., & AlQudah, M. (2024). Utilizing Sentiment Analysis for Reflect and Improve Education in Indonesia. *Journal of Applied Data Sciences*, 6(1), 189-200. doi:<https://doi.org/10.47738/jads.v6i1.527>
- Asro Asro, Solihin Solihin, & Irlon Irlon. (2026). Executive Decision Support System Implementation Strategies Based on Big Data Analytics to Improve Operational Efficiency and Corporate Governance in Global Digital Enterprises. *Integrated System and Management Technology*, 1(1), 43-51. <https://doi.org/10.66472/ismat.v1i1.13>