Learning Policies and Strategies

Volume: 5 Issue: 2 Year: 2026
ISSN: 2957-4110

KMF Publishers

www.kmf-publishers.com/lps/

Volume: 5, Issue: 2, Year: 2026

DOI: https://doi.org/10.64907/xkmf.v5i2.lps.1

Learning Policies and Strategies

RESEARCH ARTICLE

OPEN ACCESS Freely available online

Received: 2 May 2026
Accepted: 25 June 2026

Bridging the Skills Gap in Software Engineering Education: A Qualitative Case Study of Industry-Academia Alignment

[1*]Nazia Akter; Rumana Akhter Anika[1]; Al Abdullah[2]; [3]Kazi Abdul Mannan

Corresponding author:
[1*]Nazia Akter
Department of Computer Science & Engineering (CSE),
[2]Department of Computer Science & Information Technology (CSIT)
Shanto-Mariam University of Creative Technology
Dhaka, Bangladesh
E-mail:
safayedhossainsaad@gmail.com

[3]Department of Business Administration
Faculty of Business
Shanto-Mariam University of Creative Technology
Dhaka, Bangladesh

Abstract: The growing demand for skilled software engineers has exposed a persistent gap between the competencies of graduates and the expectations of the software industry. This study investigates the underlying causes of this skills gap and explores strategies for enhancing industry-academia alignment in software engineering education. Adopting a qualitative research design based on secondary data analysis, the study synthesises insights from peer-reviewed literature, industry reports, and policy documents. The analysis is guided by an integrative theoretical framework comprising Human Capital Theory, Constructivist Learning Theory, and Experiential Learning Theory. Findings reveal that curriculum misalignment, limited experiential learning opportunities, inadequate development of soft skills, weak industry-academia collaboration, and faculty skill gaps are key contributors to the problem. The study highlights best practices such as curriculum co-design, project-based learning, internship integration, and continuous professional development for educators. The findings underscore the need for systemic reforms and collaborative ecosystems involving academic institutions, industry stakeholders, and policymakers to ensure the relevance and effectiveness of software engineering education.

Keywords: software engineering education, skills gap, industry-academia collaboration, experiential learning, curriculum development, employability, qualitative analysis.

Disclosure statement
No potential conflict of interest was reported by the author(s).

Citation information
Cite this article as Akter, N., Anika, R.A., Abdullah, A., & Mannan, K.A. (2026). Bridging the Skills Gap in Software Engineering Education: A Qualitative Case Study of Industry-Academia Alignment. *Learning Policies and Strategies*, 5(2), 1-17. DOI: https://doi.org/10.64907/xkmf.v5i2.lps.1

1. Introduction

The rapid digital transformation of contemporary societies has positioned software engineering as a cornerstone of economic development, innovation, and global competitiveness. Across sectors, including healthcare, finance, education, and governance, software systems underpin critical operations and services. Consequently, the demand for highly skilled software engineers has surged worldwide. However, despite the proliferation of software engineering programs in higher education institutions, employers consistently report a mismatch between graduates' competencies and industry expectations, commonly referred to as the "skills gap" (Garousi et al., 2020; Radermacher & Walia, 2013).

The skills gap in software engineering is not merely a matter of insufficient technical knowledge; rather, it reflects a broader misalignment between academic training and the dynamic, practice-oriented nature of the software industry. Universities have traditionally prioritised theoretical foundations such as algorithms, data structures, and programming paradigms, which are undeniably essential. However, industry environments demand additional competencies, including proficiency in modern development frameworks, familiarity with agile methodologies, collaborative problem-solving, and the ability to adapt to rapidly evolving technologies (Feldt et al., 2018). This divergence between academic focus and industry needs creates a transitional challenge for graduates entering the workforce.

One of the primary drivers of this misalignment is the pace of technological change. The software industry evolves at a rate that often exceeds the capacity of academic institutions to update curricula. Emerging technologies such as artificial intelligence, cloud computing, cybersecurity, and DevOps practices require continuous curricular adaptation. However, institutional constraints, including bureaucratic processes, limited resources, and rigid accreditation requirements, often delay such updates (Garousi et al., 2020). As a result, graduates may be equipped with knowledge that is conceptually sound but practically outdated.

In addition to technical competencies, employers increasingly emphasise the importance of soft skills, including communication, teamwork, critical thinking, and problem-solving. Software development is inherently collaborative, involving cross-functional teams, stakeholder engagement, and iterative feedback processes. Yet, many educational programs provide limited opportunities for students to develop these interpersonal and professional skills (Begel & Simon, 2008). This deficiency further exacerbates the employability gap, as graduates struggle to integrate into team-based development environments.

The issue of the skills gap is particularly pronounced in developing countries, where educational systems may face additional challenges such as inadequate infrastructure, limited industry engagement, and resource constraints. In these contexts, the gap not only affects individual employability but also has broader implications for national economic development and participation in the global digital economy. However, it is important to note that the problem is not confined to developing regions; even in technologically advanced economies, employers report difficulties in finding suitably skilled software engineers (Radermacher & Walia, 2013).

Industry-academia alignment has emerged as a critical strategy for addressing this challenge. Effective collaboration between educational institutions and industry stakeholders can facilitate the integration of practical skills into curricula, enhance the relevance of academic programs, and provide students with real-world experience (Petersen et al., 2014). Such collaboration may take various forms, including internships, cooperative education programs, joint research initiatives, guest lectures, and curriculum co-design. When implemented effectively, these initiatives can bridge the gap between theory and practice, enabling students to develop both technical and professional competencies.

Despite the recognised importance of industry-academia collaboration, its implementation remains inconsistent and often limited in scope. Challenges such as differing institutional priorities, lack of sustained partnerships, and communication barriers can hinder effective collaboration. Furthermore, faculty members may lack exposure to current industry practices, limiting their ability to integrate practical insights into teaching (Feldt et al., 2018). These challenges underscore the need for a systematic and holistic approach to alignment.

This study seeks to explore the factors contributing to the skills gap in software engineering education and to examine strategies for enhancing industry-academia alignment. By adopting a qualitative research design based on secondary data analysis, the study synthesises insights from existing literature, industry reports, and policy documents. The research is guided by three theoretical perspectives: Human Capital Theory, Constructivist Learning Theory, and Experiential Learning Theory. These frameworks provide a comprehensive lens for understanding the processes of skill acquisition, application, and alignment with labour market demands.

The significance of this study lies in its potential to inform educational policy, curriculum development, and institutional practices. By identifying the underlying causes of the skills gap and highlighting effective strategies for alignment, the research contributes to ongoing efforts to enhance the quality and relevance of software engineering education. Moreover, the findings have implications for industry stakeholders, who play a crucial role in shaping the future workforce.

In summary, bridging the skills gap in software engineering education is a complex and multifaceted challenge that requires coordinated efforts from academia, industry, and policymakers. As the demand for skilled software engineers continues to grow, addressing this gap is essential for ensuring both individual employability and broader economic development.

2. Literature Review

The concept of the skills gap has been widely discussed in both academic and industry literature, often defined as the discrepancy between the skills possessed by graduates and those required by employers (Radermacher & Walia, 2013). In the context of software engineering, this gap encompasses both technical and non-technical competencies, reflecting the multifaceted nature of the profession.

Research indicates that the skills gap is not a new phenomenon but has become more pronounced with the increasing complexity of software systems and the rapid evolution of technologies (Garousi et al., 2020). Employers frequently report that graduates lack proficiency in practical programming, software design, and system integration, as well as in soft skills such as communication and teamwork (Begel & Simon, 2008).

The literature also highlights the distinction between "academic preparedness" and "industry readiness." While graduates may possess strong theoretical knowledge, they often struggle to apply this knowledge in real-world contexts. This disconnect underscores the need for educational approaches that integrate theory with practice.

2.1 Curriculum Design and Limitations

Curriculum design plays a central role in shaping students' competencies. Traditional software engineering curricula are typically structured around foundational topics such as algorithms, data structures, operating systems, and programming languages. While these subjects are essential, they may not adequately reflect the demands of modern software development (Garousi et al., 2020).

One of the key limitations identified in the literature is the slow pace of curriculum updates. Academic programs often require extensive review and approval processes, which can delay the incorporation of emerging technologies and practices. As a result, curricula may lag behind industry developments, leaving graduates underprepared for contemporary roles.

Another limitation is the lack of emphasis on interdisciplinary learning. Modern software engineering increasingly intersects with fields such as artificial intelligence, data science, cybersecurity, and human-computer interaction. However, many programs do not provide sufficient opportunities for students to engage with these areas.

Furthermore, the assessment methods used in many programs prioritise theoretical knowledge over practical skills. Traditional examinations and assignments may not effectively evaluate students' ability to design, develop, and deploy software systems in real-world environments.

2.2 Industry Expectations and Employability Skills

Industry expectations for software engineers have evolved significantly in recent years. In addition to technical expertise, employers seek candidates who possess a range of professional and interpersonal skills. These include communication, teamwork, adaptability, problem-solving, and the ability to learn continuously (Feldt et al., 2018).

The increasing adoption of agile methodologies has further emphasised the importance of collaboration and flexibility. Agile development practices require frequent communication, iterative development, and responsiveness to change. Graduates who are unfamiliar with these practices may find it challenging to adapt to industry environments.

Employers also value experience with tools and technologies commonly used in software development, such as version control systems, continuous integration pipelines, and cloud platforms. However, many graduates lack exposure to these tools, limiting their readiness for professional roles (Radermacher & Walia, 2013).

The literature suggests that employability is not solely determined by technical skills but also by the ability to apply knowledge in practical contexts. This highlights the importance of experiential learning and industry engagement in education.

2.3 Experiential Learning and Pedagogical Approaches

Experiential learning has been widely recognised as an effective approach for bridging the gap between theory and practice. Kolb's Experiential Learning Theory emphasises the role of experience, reflection, and application in the learning process (Kolb, 1984). In the context of software engineering education, experiential learning can take various forms, including project-based learning, internships, and cooperative education programs.

Project-based learning allows students to work on real-world problems, often in teams, thereby developing both technical and soft skills. Studies have shown that such approaches enhance students' problem-solving abilities, creativity, and engagement (Feldt et al., 2018).

Internships and cooperative education programs provide students with direct exposure to industry environments. These experiences enable students to apply their knowledge, gain practical skills, and develop professional networks. However, access to such opportunities may be limited, particularly in regions with fewer industry partnerships.

Despite the benefits of experiential learning, its implementation in higher education remains uneven. Challenges include resource constraints, large class sizes, and the need for faculty training.

2.4 Industry-Academia Collaboration

Industry-academia collaboration is widely regarded as a key mechanism for addressing the skills gap. Such collaboration can take various forms, including curriculum co-design, joint research projects, internships, and guest lectures (Petersen et al., 2014).

Collaborative initiatives enable educational institutions to stay updated with industry trends and ensure the relevance of their programs. At the same time, they provide the industry with access to emerging talent and opportunities for innovation.

However, the literature also highlights several challenges associated with collaboration. These include differences in objectives, with academia focusing on knowledge generation and industry prioritising practical outcomes. Additionally, resource constraints and lack of institutional support can hinder the development of sustainable partnerships.

Effective collaboration requires clear communication, mutual trust, and shared goals. Institutional frameworks and policies play a crucial role in facilitating such partnerships.

2.5 Faculty Development and Institutional Challenges

Faculty members play a critical role in bridging the skills gap, as they are responsible for designing and delivering curricula. However, many faculty members may lack exposure to current industry practices, particularly if they have primarily academic backgrounds (Feldt et al., 2018).

Continuous professional development is essential for ensuring that faculty remain updated with technological advancements. This may include industry training, collaborative research, and participation in professional networks.

Institutional challenges, such as limited funding, rigid structures, and administrative constraints, can also impact the effectiveness of educational programs. Addressing these challenges requires systemic reforms and policy support.

2.6 Global Perspectives and Policy Implications

The skills gap in software engineering is a global issue, with variations across regions. In developed countries, the gap is often attributed to the rapid pace of technological change, while in developing countries, additional factors such as infrastructure limitations and resource constraints come into play.

Policy interventions can play a significant role in addressing the gap. Governments can support initiatives such as industry partnerships, curriculum development, and faculty training. Additionally, accreditation bodies can encourage the inclusion of practical and experiential components in educational programs.

The literature suggests that a multi-stakeholder approach is essential for bridging the skills gap. Collaboration between academia, industry, and policymakers can create a supportive ecosystem for skill development and innovation.

3. Theoretical Framework

The present study is grounded in an integrative theoretical framework that combines Human Capital Theory, Constructivist Learning Theory, and Experiential Learning Theory. These frameworks collectively provide a comprehensive lens for understanding the dynamics of skill acquisition, knowledge application, and the alignment between educational outcomes and industry requirements in software engineering. By synthesising these theoretical perspectives, the study seeks to explain the underlying causes of the skills gap and to identify pathways for bridging it through effective educational practices and industry-academia collaboration.

3.1 Human Capital Theory

Human Capital Theory (HCT), as articulated by Becker (1993), posits that education and training are critical investments that enhance individuals' productivity, skills, and economic value. Within the context of software engineering education, HCT underscores the role of academic institutions in developing competencies that are directly relevant to labour market demands. The theory assumes that individuals acquire knowledge and skills through formal education, which subsequently translates into improved employability and economic outcomes.

However, the persistent skills gap suggests a misalignment between the human capital produced by educational institutions and the competencies required by industry. This misalignment can be interpreted as an inefficiency in the educational system's ability to respond to labour market signals (Garousi et al., 2020). In software engineering, where technological advancements occur rapidly, the value of human capital is closely tied to its relevance and adaptability. Graduates who possess outdated or purely theoretical knowledge may struggle to meet industry expectations, thereby reducing their employability.

HCT also emphasises the importance of continuous learning and skill development. In the software industry, where technologies evolve continuously, lifelong learning is essential. This perspective highlights the need for educational programs to instil not only technical knowledge but also the capacity for self-directed learning and adaptability (Feldt et al., 2018). Thus, Human Capital Theory provides a macro-level understanding of the economic implications of the skills gap and the necessity of aligning education with industry needs.

3.2 Constructivist Learning Theory

Constructivist Learning Theory, primarily associated with Piaget (1972), posits that learners actively construct knowledge through interaction with their environment rather than passively receiving information. This perspective emphasises the importance of active engagement, problem-solving, and experiential activities in the learning process. In software engineering education, constructivism supports pedagogical approaches that encourage students to engage in hands-on activities, collaborative projects, and real-world problem-solving.

Traditional lecture-based teaching methods, which focus on the transmission of theoretical knowledge, may not effectively support the development of practical skills required in the software industry. Constructivist approaches, on the other hand, promote deeper understanding by allowing students to apply concepts in meaningful contexts. For example, project-based learning enables students to design and develop software systems, thereby integrating theoretical knowledge with practical application.

Moreover, constructivism highlights the social dimension of learning. Software development is inherently collaborative, requiring effective communication and teamwork. Collaborative learning environments, such as group projects and peer programming, align with constructivist principles and help students develop essential soft skills (Begel & Simon, 2008).

The relevance of Constructivist Learning Theory to this study lies in its emphasis on active and contextualised learning. By advocating for learner-centred pedagogies, the theory provides a framework for addressing the limitations of traditional curricula and enhancing the alignment between academic training and industry practices.

3.3 Experiential Learning Theory

Experiential Learning Theory (ELT), developed by Kolb (1984), builds on constructivist principles by emphasising learning through experience. According to Kolb, learning is a cyclical process involving four stages: concrete experience, reflective observation, abstract conceptualisation, and active experimentation. This model is particularly relevant to software engineering education, where practical experience plays a crucial role in skill development.

ELT supports the integration of experiential learning opportunities such as internships, cooperative education, and project-based learning into academic programs. These experiences allow students to apply theoretical knowledge in real-world contexts, reflect on their experiences, and refine their understanding. For example, internships provide exposure to industry practices, tools, and workflows, enabling students to develop competencies that are directly relevant to their future careers (Feldt et al., 2018).

In addition to technical skills, experiential learning fosters the development of soft skills such as communication, teamwork, and problem-solving. These competencies are essential for success in collaborative software development environments. By engaging in real-world projects, students learn to navigate complex problems, work in teams, and adapt to changing requirements.

The application of ELT in software engineering education also addresses the issue of transferability of knowledge. Students who engage in experiential learning are better able to transfer their academic knowledge to professional contexts, thereby reducing the transition gap between education and employment.

3.4 Integrative Framework and Relevance to the Study

The integration of Human Capital Theory, Constructivist Learning Theory, and Experiential Learning Theory provides a holistic framework for understanding the skills gap in software engineering education. While HCT emphasises the economic value of skills and the importance of aligning education with labour market demands, constructivism and ELT focus on the processes through which these skills are acquired and applied.

Together, these theories highlight the need for educational approaches that are both relevant and effective. Aligning curricula with industry needs (HCT), adopting learner-centred pedagogies (constructivism), and incorporating experiential learning opportunities (ELT) are essential strategies for bridging the skills gap.

This integrative framework guides the analysis of secondary data in this study, enabling a comprehensive examination of the factors contributing to the skills gap and the strategies for addressing it.

4. Research Methodology

This study adopts a qualitative research design based on secondary data analysis to explore the skills gap in software engineering education and the role of industry-academia alignment in addressing it. Qualitative research is particularly suitable for this study as it allows for an in-depth exploration of complex phenomena, including educational practices, institutional dynamics, and stakeholder perspectives (Creswell & Poth, 2018).

Secondary data analysis involves the systematic examination and synthesis of existing data sources, including academic literature, industry reports, and policy documents. This approach is appropriate for the present study, as it enables the integration of diverse perspectives and the identification of recurring themes across multiple contexts (Johnston, 2017).

4.1 Data Sources and Selection Criteria

The study draws on a wide range of secondary data sources, including:

- Peer-reviewed journal articles in software engineering and education
- Conference proceedings from reputable organisations such as IEEE and ACM
- Industry reports and white papers
- Policy documents and educational frameworks

The selection of sources was guided by the following criteria:

- **Relevance:** Sources must address software engineering education, skills gaps, or industry-academia collaboration.
- **Credibility:** Preference was given to peer-reviewed publications and reports from recognised organisations.
- **Recency:** Emphasis was placed on studies published within the last decade to ensure relevance to current industry trends.
- **Diversity:** Sources were selected to represent a range of perspectives, including academic, industry, and policy viewpoints.

This rigorous selection process ensures that the data used in the study are both reliable and relevant.

4.2 Data Analysis Method

The study employs thematic analysis as the primary method for analysing secondary data. Thematic analysis is a widely used qualitative method for identifying, analysing, and reporting patterns (themes) within data (Braun & Clarke, 2006). It is particularly suitable for synthesising findings from multiple sources and generating insights into complex issues.

The analysis was conducted in several stages:

Data Familiarisation: The researcher conducted an extensive review of the selected sources to gain a comprehensive understanding of the literature. This involved reading and re-reading the data to identify initial patterns and insights.

Coding: Relevant segments of the data were coded based on their content. Codes were assigned to key concepts such as "curriculum misalignment," "practical skills," "soft skills," and "industry collaboration."

Theme Development: Codes were grouped into broader themes that captured recurring patterns across the data. For example, codes related to curriculum limitations and outdated content were grouped under the theme of "curriculum misalignment."

Interpretation: The identified themes were analysed in relation to the theoretical framework and research objectives. This stage involved synthesising the findings and drawing connections between different themes.

The use of thematic analysis allows for a systematic and transparent approach to data analysis, enhancing the credibility of the study.

4.3 Validity and Reliability

Ensuring the validity and reliability of qualitative research is essential for producing credible findings. In this study, several strategies were employed to enhance rigour:

Triangulation: Data were collected from multiple sources, including academic literature, industry reports, and policy documents. This triangulation helps to validate findings by cross-verifying information from different perspectives (Creswell & Poth, 2018).

Transparency: The research process, including data selection and analysis procedures, is clearly documented to ensure transparency and replicability.

Use of Credible Sources: Only peer-reviewed publications and reports from reputable organisations were included in the analysis, ensuring the reliability of the data.

Theoretical Grounding: The use of established theoretical frameworks provides a structured basis for analysis and interpretation, enhancing the study's validity.

4.4 Ethical Considerations

As this study is based on secondary data, it does not involve direct interaction with human participants. However, ethical considerations were still observed, including:

- Proper citation and acknowledgement of all sources
- Avoidance of plagiarism
- Accurate representation of original authors' ideas

These practices ensure the integrity and ethical standards of the research (Mannan & Farhana, 2026).

4.5 Limitations of the Methodology

While secondary data analysis offers several advantages, it also has limitations. The study relies on existing data, which may not fully capture current or localised contexts. Additionally, the findings are dependent on the quality and scope of the selected sources.

Despite these limitations, the use of diverse and credible sources helps to mitigate potential biases and enhances the robustness of the findings.

5. Findings and Analysis

The thematic analysis of secondary data reveals a complex and multidimensional set of factors contributing to the skills gap in software engineering education. These factors are interrelated and operate at multiple levels, including curriculum design, pedagogical practices, institutional structures, and industry engagement. The following sections present the key findings organised into major themes.

5.1 Curriculum Misalignment with Industry Needs

One of the most prominent findings across the literature is the persistent misalignment between academic curricula and industry requirements. Traditional software engineering programs continue to emphasise foundational theoretical concepts such as algorithms, data structures, and formal methods. While these are indispensable for building a strong conceptual base, they are insufficient in isolation for preparing graduates for the practical demands of the software industry (Garousi et al., 2020).

A recurring issue is the lag in curriculum updates. Academic institutions often operate within rigid regulatory and administrative frameworks, which delay the integration of emerging technologies such as cloud computing, artificial intelligence, DevOps, and microservices architecture. Consequently, graduates enter the workforce with knowledge that may already be outdated or incomplete in relation to current industry practices (Radermacher & Walia, 2013).

Furthermore, the lack of alignment extends beyond technical content to include methodologies and workflows. Industry practices such as Agile, Scrum, and continuous integration/continuous deployment (CI/CD) are rarely integrated systematically into curricula. This results in graduates being unfamiliar with collaborative and iterative development processes, which are central to modern software engineering.

5.2 Insufficient Practical and Experiential Learning Opportunities

Another critical theme is the limited emphasis on practical and experiential learning within software engineering programs. Although many institutions incorporate laboratory sessions and capstone projects, these experiences often lack the complexity and authenticity of real-world software development environments.

The literature highlights that students frequently graduate without exposure to industry-standard tools and practices, such as version control systems (e.g., Git), issue tracking platforms, and collaborative development environments (Begel & Simon, 2008). This lack of exposure hampers their ability to transition effectively into professional roles.

Internships and cooperative education programs are widely recognised as effective mechanisms for providing practical experience. However, access to such opportunities is uneven across institutions and regions. In many cases, internships are optional rather than mandatory, and their quality varies significantly. This inconsistency limits their effectiveness in bridging the skills gap (Feldt et al., 2018).

Moreover, experiential learning opportunities are often constrained by institutional factors such as large class sizes, limited resources, and insufficient industry partnerships. These constraints reduce the feasibility of implementing project-based and experiential learning approaches at scale.

5.3 Deficiency in Soft Skills and Professional Competencies

In addition to technical skills, employers consistently emphasise the importance of soft skills, including communication, teamwork, adaptability, and problem-solving. However, the findings indicate that these competencies are often underdeveloped among graduates (Feldt et al., 2018).

Software development is inherently a collaborative activity that requires effective communication among team members, stakeholders, and clients. Yet, many educational programs focus primarily on individual performance and technical proficiency, neglecting the development of interpersonal skills. This misalignment is particularly problematic in Agile environments, where collaboration and continuous feedback are essential.

The deficiency in soft skills is further exacerbated by traditional assessment methods, which prioritise theoretical knowledge and individual achievement. Group projects, when included, are often not structured in a way that fosters genuine collaboration or accountability.

Additionally, graduates may lack professional competencies such as time management, adaptability, and the ability to work under pressure. These skills are critical for navigating the dynamic and fast-paced nature of the software industry.

5.4 Weak Industry-Academia Collaboration

The analysis reveals that industry-academia collaboration remains limited and fragmented in many contexts. While the importance of such collaboration is widely acknowledged, its implementation is often constrained by structural and institutional barriers (Petersen et al., 2014).

One key issue is the lack of sustained and formalised partnerships between educational institutions and industry organisations. Collaboration initiatives are often ad hoc and dependent on individual relationships rather than institutional frameworks. This lack of continuity limits their impact and scalability.

Furthermore, differences in objectives and expectations between academia and industry can hinder effective collaboration. Academic institutions prioritise knowledge generation and theoretical rigour, while industry focuses on practical outcomes and efficiency. These differing priorities can create challenges in aligning goals and designing collaborative initiatives.

Resource constraints also play a significant role. Developing and maintaining partnerships requires time, funding, and administrative support, which may be limited in many institutions. Additionally, industry partners may be reluctant to invest in collaboration due to concerns about cost, confidentiality, and return on investment.

5.5 Faculty Skill Gaps and Professional Development

Faculty members are central to the delivery of software engineering education, yet the findings indicate that many educators lack exposure to current industry practices. This gap is particularly evident among faculty who have primarily academic backgrounds and limited industry experience (Garousi et al., 2020).

The rapid pace of technological change further complicates this issue, as faculty must continuously update their knowledge and skills to remain relevant. However, opportunities for professional development are often limited, and institutional support for such initiatives may be insufficient.

This disconnect between faculty expertise and industry practices can result in outdated teaching methods and content, further contributing to the skills gap. Additionally, faculty may lack the resources or incentives to integrate innovative pedagogical approaches, such as project-based learning or industry collaboration, into their teaching.

5.6 Institutional and Policy Constraints

The findings also highlight the role of institutional and policy-level factors in shaping software engineering education. Rigid curricular structures, accreditation requirements, and administrative processes can limit the flexibility of institutions to adapt to changing industry needs.

For example, curriculum revisions often require lengthy approval processes, which delay the incorporation of new technologies and practices. Similarly, accreditation standards may prioritise theoretical knowledge over practical skills, influencing the design of academic programs.

At the policy level, insufficient funding and a lack of strategic focus on industry-academia collaboration can hinder efforts to bridge the skills gap. In many contexts, there is a need for more proactive policy interventions to support curriculum reform, faculty development, and collaborative initiatives.

5.7 Emerging Best Practices

Despite these challenges, the literature identifies several best practices for bridging the skills gap. These include:

- Curriculum co-design with industry stakeholders
- Integration of internships and cooperative education programs
- Adoption of project-based and experiential learning approaches
- Continuous professional development for faculty
- Use of industry-standard tools and technologies in teaching

These practices demonstrate the potential for effective alignment between academia and industry when appropriate strategies and support mechanisms are in place.

6. Discussion

The findings of this study underscore the multifaceted nature of the skills gap in software engineering education and highlight the need for a comprehensive and integrated approach to addressing it. This section interprets the findings through the lens of the theoretical framework and situates them within the broader literature.

6.1 Reinterpreting the Skills Gap Through Human Capital Theory

From the perspective of Human Capital Theory, the skills gap represents a misalignment between the competencies produced by educational institutions and the demands of the labour market (Becker, 1993). This misalignment reflects inefficiencies in the development and utilisation of human capital, with significant implications for both individuals and organisations.

The findings suggest that current educational practices do not fully equip students with the skills required for productive participation in the software industry. As a result, graduates may face challenges in securing employment, while employers incur additional costs for training and onboarding.

Addressing this issue requires a shift from a supply-driven model of education, where curricula are determined primarily by academic considerations, to a demand-driven model that incorporates industry needs. This shift aligns with the principles of Human Capital Theory and emphasises the importance of aligning educational outcomes with economic objectives.

6.2 Constructivist Perspectives on Learning and Skill Development

The findings also highlight the limitations of traditional, lecture-based pedagogies in developing practical skills and competencies. From a constructivist perspective, learning is most effective when students actively engage with content and construct knowledge through experience (Piaget, 1972).

The lack of experiential and project-based learning opportunities identified in the findings suggests that many programs do not fully embrace constructivist principles. As a result, students may struggle to apply theoretical knowledge in real-world contexts.

Integrating constructivist approaches into software engineering education can enhance learning outcomes by promoting active engagement, critical thinking, and collaboration. For example, project-based learning allows students to work on real-world problems, thereby bridging the gap between theory and practice.

6.3 The Central Role of Experiential Learning

Experiential Learning Theory provides a powerful framework for understanding the importance of practical experience in skill development (Kolb, 1984). The findings clearly indicate that experiential learning opportunities, such as internships and project-based learning, are essential for bridging the skills gap.

These experiences enable students to engage in the full cycle of learning, from concrete experience to reflection and application. They also facilitate the development of both technical and soft skills, which are critical for success in the software industry.

However, the uneven availability of experiential learning opportunities highlights the need for institutional and policy-level interventions. Ensuring that all students have access to high-quality experiential learning experiences is essential for achieving equitable outcomes.

6.4 Rethinking Industry-Academia Collaboration

The findings emphasise the importance of industry-academia collaboration as a mechanism for aligning educational outcomes with industry needs. However, the limited and fragmented nature of existing collaborations suggests that more systematic and sustained efforts are required.

Effective collaboration requires the establishment of formal partnerships, clear communication channels, and shared objectives. Institutions must move beyond ad hoc initiatives and develop strategic frameworks for engagement with industry partners.

Moreover, collaboration should not be limited to internships and guest lectures but should extend to curriculum design, research, and faculty development. Such comprehensive engagement can create a more integrated and responsive educational ecosystem.

6.5 Addressing Faculty and Institutional Challenges

The role of faculty in bridging the skills gap cannot be overstated. As the primary agents of curriculum delivery, educators must possess both theoretical knowledge and practical expertise. The findings highlight the need for continuous professional development to ensure that faculty remain updated with industry practices.

Institutional support is critical in this regard. Universities must provide resources and incentives for faculty to engage in professional development and industry collaboration. This may include funding for training, opportunities for industry placements, and recognition of industry engagement in performance evaluations.

At the institutional level, greater flexibility in curriculum design and implementation is needed. Streamlining administrative processes and revising accreditation standards can facilitate more responsive and adaptive educational practices.

6.6 Policy Implications and Systemic Reform

The findings and analysis point to the need for systemic reform at the policy level. Governments and regulatory bodies play a crucial role in shaping the landscape of higher education and industry engagement.

Policy interventions can support the development of industry-academia partnerships through funding, incentives, and regulatory frameworks. For example, governments can provide grants for collaborative projects, tax incentives for industry participation, and guidelines for integrating experiential learning into curricula.

Additionally, national strategies for digital skills development can help align educational outcomes with economic priorities. Such strategies should involve multiple stakeholders, including educational institutions, industry organisations, and policymakers.

6.7 Toward a Holistic Model of Alignment

Ultimately, bridging the skills gap requires a holistic and integrated approach that addresses multiple dimensions of the problem. This includes:

- Aligning curricula with industry needs (Human Capital Theory)
- Adopting learner-centred pedagogies (Constructivism)
- Integrating experiential learning opportunities (Experiential Learning Theory)
- Strengthening industry-academia collaboration
- Supporting faculty development and institutional reform

By combining these elements, it is possible to create a more responsive and effective system of software engineering education that meets the needs of both students and employers.

7. Conclusion

The skills gap in software engineering education represents a critical challenge with far-reaching implications for individual employability, organisational productivity, and broader economic development. This study has examined the nature and causes of this gap through a qualitative analysis of secondary data, drawing on an integrative theoretical framework that combines Human Capital Theory, Constructivist Learning Theory, and Experiential Learning Theory.

The findings demonstrate that the skills gap is not the result of a single factor but rather a complex interplay of curricular, pedagogical, institutional, and collaborative shortcomings. Traditional curricula, while strong in theoretical foundations, often fail to keep pace with rapidly evolving industry requirements. This lag, coupled with insufficient emphasis on experiential learning, leaves graduates underprepared for the practical realities of software development. Furthermore, the limited development of soft skills and professional competencies further exacerbates the transition challenges faced by graduates entering the workforce.

The study also highlights the critical role of industry-academia collaboration in bridging this gap. However, existing collaborations are often fragmented and lack the strategic depth required for sustainable impact. Strengthening these partnerships through structured and long-term initiatives is essential for aligning educational outcomes with industry needs. Additionally, the role of faculty as mediators of knowledge and practice underscores the importance of continuous professional development and institutional support.

From a theoretical perspective, the study reinforces the importance of aligning human capital development with labour market demands, adopting learner-centred pedagogies, and integrating experiential learning into educational programs. These approaches collectively contribute to a more responsive and effective system of software engineering education.

In conclusion, bridging the skills gap requires a holistic and coordinated effort involving educational institutions, industry stakeholders, and policymakers. By fostering dynamic curricula, promoting experiential learning, strengthening partnerships, and supporting faculty development, it is possible to create an educational ecosystem that equips graduates with the skills and competencies needed to thrive in the modern software industry. Future research may focus on empirical validation of these strategies and the development of context-specific models for industry-academia alignment.

References

Becker, G. S. (1993). *Human capital: A theoretical and empirical analysis* (3rd ed.). University of Chicago Press.

Begel, A., & Simon, B. (2008). Novice software developers, all over again. *Proceedings of the Fourth International Workshop on Computing Education Research*, 3-14.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77-101.

Creswell, J. W., & Poth, C. N. (2018). *Qualitative inquiry and research design: Choosing among five approaches* (4th ed.). Sage Publications.

Feldt, R., Torkar, R., Angelis, L., & Samuelsson, M. (2018). Towards individualised software engineering education. *IEEE Transactions on Education*, 61(3), 194-203.

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.

Johnston, M. P. (2017). Secondary data analysis: A method whose time has come. *Qualitative and Quantitative Methods in Libraries*, 3(3), 619-626.

Kolb, D. A. (1984). *Experiential learning: Experience as the source of learning and development*. Prentice Hall.

Mannan, K.A., & Farhana, K.M. (2026). *The Principles of Qur'anic Research Methodology: Deriving the Process of Knowledge from Revelation.* KMF Publishers. Open Access (CC BY 4.0). DOI: https://doi.org/10.64907/xkmf.book.pqrm.26.02.12

Piaget, J. (1972). *The psychology of the child*. Basic Books.

Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M. (2014). Systematic mapping studies in software engineering. *Information and Software Technology*, 50(1-2), 71-81.

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.