Re Imagining the 4P Framework in Graduate Software Engineering Education: From Project to Product Mindset for Software Process Improvement

Suresh Kumar. S., Dr. R. Preethi

Research Scholar & Assistant Professor, Dean School of Science, Park's College (Autonomous), Tirupur, Tamil Nadu

DOI: https://doi.org/10.51584/IJRIAS.2026.11050176

Received: 18 May 2026; Accepted: 23 May 2026; Published: 12 June 2026

ABSTRACT

The persistent gap between graduate capabilities and contemporary software engineering practice continues to challenge academia and industry despite decades of curricular reform. Traditional models of software engineering education often emphasize plan-driven processes, narrow technical skills, and project-centric thinking that do not fully align with agile, product-centric, and cross-functional modes of real-world software development. This paper proposes an updated interpretation of the classic 4P framework (People, Product, Process, Project) for graduate software engineering education that emphasizes cross-functional team skills, contemporary product design, agile and continuous process models, and a shift from project- to product-oriented thinking. Building on prior work on software process improvement (SPI) in graduate curricula and industry-linked project-based learning, we develop a conceptual curriculum framework and theoretically examine its potential to enhance graduate readiness and support software process improvement outcomes.

The study is guided by three research questions: (1) How can the 4P framework be adapted to better reflect current industrial practices in software engineering? (2) To what extent does an updated 4P framework address known gaps in graduate readiness for cross-functional, agile, and product-centric environments? (3) How can such a framework be operationalized in a graduate curriculum to support software process improvement competencies? From these questions, we derive hypotheses about the relationships between the updated 4P elements and graduate readiness for SPI-oriented roles. The paper adopts a conceptual and design-oriented research approach synthesizing literature on SPI in graduate education, cross-functional collaboration, and product versus project mindsets. We present a structured curriculum model aligned to the updated 4P framework and articulate expected learning outcomes, graduate capabilities, and SPI-aligned competencies. The paper concludes with recommendations for implementation, implications for educators and industry, and directions for empirical validation of the proposed framework.

Keywords: Software Engineering Education, Software Process Improvement, 4P, Project to Product Mindset, SPI-aligned competencies

INTRODUCTION

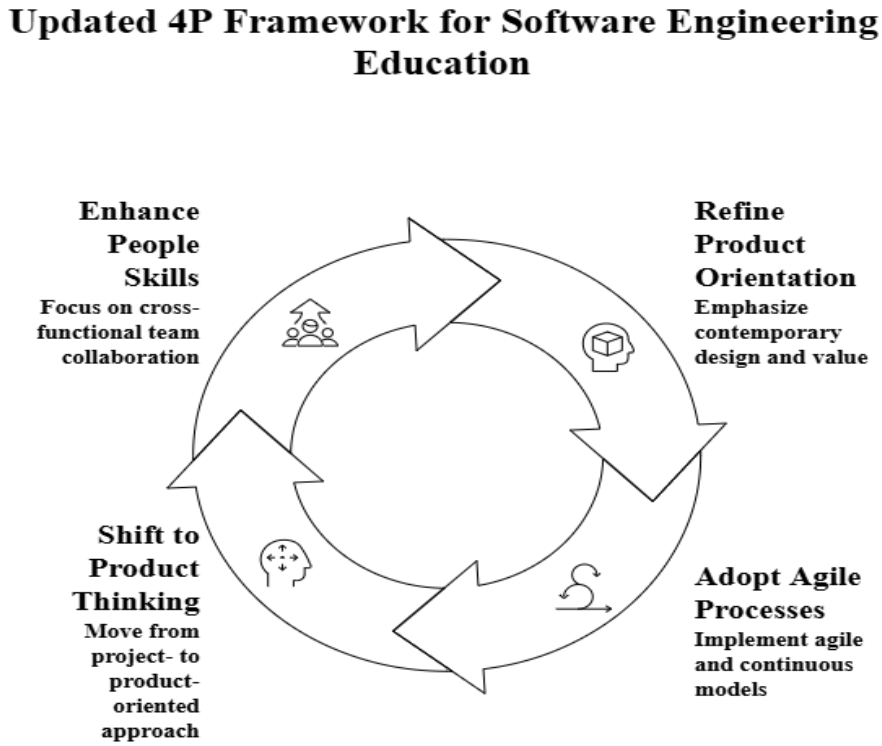
Software engineering education must continually evolve to remain aligned with rapid changes in software development practices, including agile methods, DevOps, and product-centric organizational structures. Recent studies highlight that traditional lecture-centered and project-limited approaches struggle to cultivate the breadth of technical, soft, and process skills required for modern software engineering roles. At the same time, software process improvement (SPI) remains a core concern for organizations seeking predictability, quality, and responsiveness, and graduate programs are increasingly expected to prepare students to participate in or lead SPI initiatives.

The classic 4P framework in software engineering—People, Product, Process, Project—has long been used to conceptualize key dimensions of software development and management. However, the way these dimensions are instantiated in curricula often mirrors older, plan-driven paradigms rather than contemporary industrial

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

practice. In many programs, courses focus on individual programming assignments, isolated design tasks, and time-bounded "projects" rather than long-lived product evolution and cross-functional collaboration.

This paper argues that the 4P framework can still serve as a powerful organizing lens for graduate software engineering curricula if it is updated to emphasize four elements: (a) cross-functional team skills under People, (b) contemporary product design and value orientation under Product, (c) agile and continuous models under Process, and (d) a shift from project- to product-oriented thinking under Project. By embedding SPI concepts throughout these four dimensions, the framework aims to improve graduate readiness for modern roles and strengthen their ability to contribute to software process improvement in industry.



The remainder of this paper is structured as follows. Section 2 reviews related work on software engineering education, SPI in graduate curricula, cross-functional collaboration, and product versus project mindsets. Section 3 presents the research questions and hypotheses. Section 4 outlines the methodology and conceptual design approach. Section 5 introduces the updated 4P framework and the proposed curriculum model. Section 6 discusses expected impacts on graduate readiness and SPI capabilities. Section 7 concludes with implications, limitations, and avenues for future research.

LITERATURE REVIEW

Software Engineering Education and Graduate Readiness

Multiple studies have documented a persistent mismatch between software engineering curricula and industry needs, particularly around practical skills, collaboration, and process understanding. Conceptual frameworks for software engineering education increasingly stress the need for project-based learning, industry collaboration, and integration of both hard skills (e.g., design, coding, testing) and soft skills (e.g., communication, teamwork). A recent framework for evaluating software engineering programs emphasizes that future-ready curricula must explicitly cultivate continuous learning, creativity, and solution-oriented thinking to align with industry expectations.

These findings converge on a central theme: graduate readiness requires more than mastery of technical content; it demands the ability to operate effectively in realistic, team-based and process-intensive contexts. Project-based learning with real-world projects and market-oriented aspects has been shown to promote deeper understanding of software engineering concepts and better transition to professional practice.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

Software Process Improvement in Graduate Curricula

SPI has been integrated into some graduate programs through specialized courses that simulate or directly engage with industry environments. One influential approach used SPI projects with real organizations, adopting a "problem–goal–solution" structure where students diagnose process issues, develop business cases, propose process changes, and pilot improvements. Students in such courses learn to identify weaknesses in organizational processes, manage change, and document lessons learned, thereby acquiring practical SPI competencies.

These SPI-focused curricular interventions demonstrate that graduate students can effectively learn process improvement methods when given structured frameworks, authentic problems, and clear deliverables. However, SPI is often confined to a single capstone-style course rather than woven through the broader curriculum, limiting its impact on overall graduate readiness.

Cross-Functional Collaboration and People Dimension

Cross-functional collaboration is now central to modern software development, where teams often include engineers, designers, product managers, and other stakeholders across technical and non-technical roles. Recent work describes interdepartmental teaching approaches in which software engineering students collaborate with students from management or other disciplines on real-world projects using agile frameworks such as Scrum. Such approaches have been associated with higher levels of collaboration, improved understanding of roles and responsibilities, and measurable improvements in code quality and task completion.

This literature suggests that the People dimension of software engineering education should explicitly focus on cross-functional skills, including communication across disciplines, negotiation of priorities, and shared ownership of outcomes. Embedding these skills within the 4P framework aligns the People element with contemporary team-based practice.

Product Versus Project Mindset

The distinction between project mindset and product mindset has attracted increasing attention in both industry and practitioner literature. A project mindset emphasizes fixed scope, timelines, and deliverables, prioritizing on-time and on-budget completion of defined work packages. By contrast, a product mindset emphasizes continuous improvement, outcome orientation, and long-term value creation, with flexible scope and evolving requirements based on user feedback and business goals.

Recent reports indicate that many organizations are shifting from project-centric to product-centric application management to accelerate delivery, improve customer experience, and reduce friction. This shift has significant implications for educational programs that still tend to organize software work as one-off projects with limited focus on lifecycle evolution, product metrics, or long-term stewardship. Training graduates to think in product terms—continuous delivery, experimentation, and value measurement—may better position them for modern roles in DevOps, product engineering, and continuous process improvement.

Summary of Gaps

The literature points to three interconnected gaps:

- Curricula still lean heavily on traditional, lecture-based and project-limited models that do not fully reflect agile, product-centric practice.
- SPI is often treated as a specialized topic rather than as a pervasive competency woven through people, product, process, and project dimensions.
- Cross-functional collaboration and product mindset, while recognized as critical in industry, are not yet systematically embedded in graduate curricula.

These gaps motivate the need to reinterpret and update the 4P framework as an integrative lens for graduate software engineering education.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

Research Questions and Hypotheses

Based on the literature review, this study addresses the following research questions (RQs):

RQ1: How can the traditional 4P framework (People, Product, Process, Project) be adapted to better reflect cross-functional, agile, and product-centric practices in contemporary software engineering?

RQ2: To what extent does an updated 4P framework address known gaps in graduate readiness for SPI-related roles in industry?

RQ3: How can an updated 4P framework be operationalized in a graduate software engineering curriculum to scaffold software process improvement competencies?

From these questions, we derive the following hypotheses regarding the expected relationships between the updated 4P elements and graduate outcomes:

H1: Embedding explicit cross-functional collaboration experiences under the People dimension will positively influence graduates' perceived readiness for team-based and SPI-oriented roles.

H2: Focusing on contemporary product design and value orientation under the Product dimension will enhance graduates' ability to align technical decisions with business and user outcomes.

H3: Integrating agile and continuous process models under the Process dimension will improve graduates' competence in applying and improving modern development and delivery practices.

H4: Shifting the Project dimension toward a product-oriented mindset will strengthen graduates' understanding of long-term lifecycle management and continuous software process improvement.

These hypotheses frame the conceptual design of the curriculum and suggest directions for empirical validation in future work.

METHODOLOGY

This study employs a conceptual and design-oriented research methodology, often described as design science or conceptual framework development in software engineering education research. The approach involves synthesizing existing empirical and conceptual studies, identifying gaps, and proposing a structured framework intended to guide curriculum design and future empirical evaluation.

The methodology consisted of three main stages:

Literature synthesis: We reviewed recent works on SPI in graduate education, conceptual frameworks for software engineering curricula, cross-functional collaboration in teaching, and product versus project mindsets. This synthesis clarified current practice, challenges, and trends.

Framework reinterpretation: We reinterpreted the 4P framework by mapping contemporary industry practices and educational needs onto each dimension. This involved aligning People with cross-functional skills, Product with modern product thinking, Process with agile and continuous models, and Project with product-oriented life cycle perspectives.

Curriculum model design: We sketched a graduate curriculum structure embodying the updated 4P framework and articulated expected learning outcomes and SPI competencies. The model is intended as a design artifact and theoretical contribution, to be evaluated empirically through future case studies and course implementations.

As a conceptual study, the paper does not report original empirical data; instead, it offers propositions and design principles grounded in existing research, suitable for subsequent validation through controlled studies, quasi-experiments, or longitudinal evaluations in academic programs.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

Updated 4P Framework for Graduate Software Engineering Education

Conceptual Overview

This section addresses RQ1 by presenting an updated interpretation of the 4P framework aligned with contemporary industrial practice. The proposed framework retains the four classical dimensions—People, Product, Process, Project—but redefines their focal points to align with modern industry practice and SPI goals. Table 1 provides a high-level comparison.

Table 1. Traditional vs. updated 4P emphases

Dimension	Traditional curricular focus	Updated curricular focus
People	Individual skills, intra-team communication	Cross-functional collaboration, inter professional skills, shared ownership
Product	Requirements and design documents, static artifacts	Product outcomes, user value, product discovery, UX and experimentation
Process	Plan-driven life cycle, basic SDLC phases	Agile, DevOps, continuous delivery, continuous improvement practices
Project	Time-bounded course project, fixed scope	Product-oriented mindset, long-lived codebase, lifecycle and metrics

The updated framework treats SPI as a cross-cutting concern that emerges when graduates can jointly reason about people, product, process, and project (product) dynamics.

"The overall curriculum architecture based on the updated 4P framework and integrated SPI activities is illustrated in Figure 1."

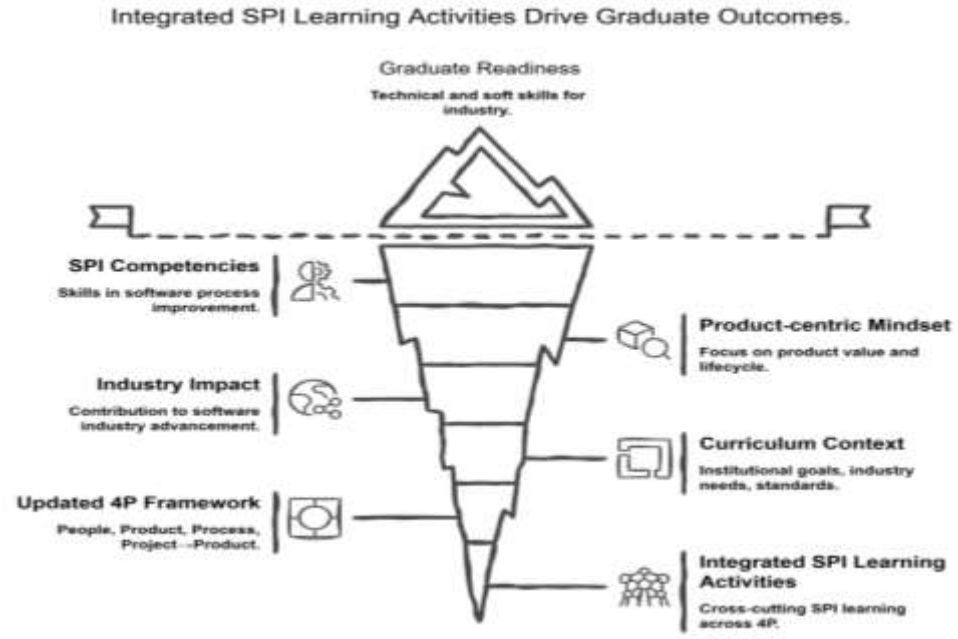


Figure 1. Updated 4P–SPI Curriculum Architecture

People: Cross-Functional Team Skills

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

In the updated framework, the People dimension emphasizes cross-functional interaction and inter professional learning. Courses are designed so that software engineering students work with peers from management, design, or other related programs on shared projects, reflecting industry patterns where diverse roles collaborate toward common goals.

Key curricular elements include:

Structured collaboration with non-technical or semi-technical stakeholders through interdepartmental courses or industry-linked projects.

Explicit learning outcomes targeting communication across roles, conflict resolution, negotiation of priorities, and understanding of organizational structures.

Assessment of team effectiveness, role clarity, and collaboration quality alongside technical outcomes.

Evidence from interdepartmental approaches indicates that such designs improve collaboration, understanding of roles, and software development performance, suggesting that a People dimension built on cross-functional skills can enhance graduate readiness for complex SPI contexts.

Product: Contemporary Product Design and Value Orientation

The Product dimension is reframed from static artifacts to dynamic, value-centric product thinking. Instead of focusing solely on specifications and design diagrams, courses emphasize product discovery, user experience, market fit, and measurable outcomes.

Illustrative components include:

Integration of product management concepts such as problem framing, hypothesis-driven development, and outcome metrics.

Exposure to user research methods, usability evaluation, and iterative product refinement.

Use of product-level metrics (e.g., adoption, engagement, defect trends) to guide technical and process decisions. Such a shift helps students connect software design choices with business and user value, aligning with industry trends that favor product-centric structures and continuous delivery of value. This orientation is critical for SPI, as process decisions are justified in terms of their impact on outcomes rather than mere compliance.

Process: Agile and Continuous Models

Under the updated framework, the Process dimension foregrounds agile methods, DevOps practices, and continuous improvement cycles instead of primarily teaching sequential life cycles. Students are expected to not only practice but also analyze and improve processes.

Core elements include:

- Hands-on practice with agile frameworks (e.g., Scrum, Kanban) combined with reflection on process effectiveness.
- Introduction to CI/CD pipelines, automated testing, and monitoring as enablers of frequent, reliable change.
- Activities that require students to measure process performance (e.g., lead time, deployment frequency, defect rates) and propose improvements, echoing SPI practices observed in industry-focused graduate courses.
- By engaging students in iterative process experimentation and data-informed improvement, the Process dimension develops competencies directly relevant to SPI, where continuous measurement and adaptation are central.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue V May 2026

Project: From Project to Product Mindset

The Project dimension is reconceived as a product-oriented dimension that emphasizes long-term stewardship over short-term completion. While coursework will still use semester-bounded structures, projects are framed as episodes in an ongoing product lifecycle rather than one-off tasks.

Practical implementations might include:

- Multi-course product lines, where teams inherit and evolve an existing codebase developed by prior cohorts, simulating long-lived systems.
- Evaluation criteria that value maintainability, extensibility, and operational readiness in addition to functionality and timeliness.
- Reflection assignments that compare project versus product perspectives, drawing on industry analyses of their differences in scope, lifecycle, deliverables, and roles.
- Such experiences aim to cultivate a product mindset in which graduates think in terms of outcomes, lifecycle, and sustained improvement instead of merely completing a deliverable, aligning their mental models with modern product engineering cultures.

Proposed Curriculum Model and Expected Outcomes

Curriculum Structure

This subsection addresses RQ3 by outlining a curriculum structure that operationalizes the updated 4P framework through four integrated course pillars.

Building on the updated 4P framework, we outline a notional graduate curriculum structure with four integrated course "pillars":

People & Collaboration Studio: An interdepartmental course with mixed teams (e.g., software engineering and management students) executing an agile project with real stakeholders, emphasizing cross-functional collaboration outcomes.

Product Design and Strategy: A course focusing on product discovery, UX, lean experimentation, and aligning technical decisions with business value.

Agile Process and DevOps: A practice-heavy course where students implement, monitor, and improve agile and DevOps practices, including CI/CD and automated testing, with explicit SPI assignments.

Product Lifecycle and SPI Capstone: A capstone where students inherit an existing product, implement new features, manage technical debt, and conduct a structured SPI initiative, culminating in a retrospective and improvement plan.

SPI topics—such as diagnosing process problems, developing business cases, and planning and piloting improvements—are embedded across all pillars, building on established SPI curricula.

Expected Impact on Graduate Readiness

This subsection contributes to answering RQ2 by discussing the expected impact of the updated 4P–based curriculum on graduate readiness.

The updated framework and curriculum model are expected to positively affect graduate readiness in several dimensions:

Technical and process skills: Graduates gain hands-on experience with agile, DevOps, and measurement-driven improvement, corresponding to current industrial expectations.

Cross-functional collaboration: Interdepartmental experiences and explicit People-focused outcomes prepare graduates to work effectively with diverse stakeholders, reflecting growing evidence of the value of such designs.

Product-centric thinking: Exposure to product strategy and lifecycle management fosters a product mindset that aligns with organizations' shift from project to product structures.

SPI capabilities: Through repeated, scaffolded engagements with process diagnosis, improvement planning, and pilot evaluation, graduates develop practical SPI skills beyond abstract process models.

These expected impacts correspond directly to the hypotheses articulated in Section 3 and can be operationalized as measurable learning outcomes in future empirical studies.

Implications for SPI

Building on the preceding analysis, this subsection further elaborates the implications of the framework for software process improvement (RQ2).

By blending updated 4P elements with SPI principles, the framework positions graduates not only as competent practitioners but also as emerging change agents. They are equipped to:

Identify people, product, process, and project (product) factors contributing to organizational challenges.

Formulate improvement hypotheses grounded in product outcomes and team dynamics rather than solely in prescriptive models.

Collaborate across roles to design, pilot, and refine process and practice changes, echoing proven SPI education approaches.

Thus, the framework links educational design to organizational outcomes, suggesting that curriculum reform can contribute to broader software process improvement goals.

CONCLUSION AND FUTURE WORK

This paper proposes an updated interpretation of the 4P framework for graduate software engineering education, emphasizing cross-functional People skills, value-centric Product thinking, agile and continuous Process models, and a product-oriented reinterpretation of the Project dimension. The framework is grounded in contemporary literature on software engineering education, SPI in graduate curricula, cross-functional collaboration, and product versus project mindsets.

We formulated research questions and hypotheses regarding the potential of this updated framework to improve graduate readiness and build SPI-related competencies. A conceptual curriculum model illustrates how the framework can be instantiated through integrated courses and SPI-focused learning activities. While the present work is conceptual, it offers a structured basis for empirical evaluation in real programs.

Future research should empirically investigate the proposed hypotheses through mixed-method studies, including pre- and post-course surveys of graduate readiness, analysis of project and product artifacts, and longitudinal tracking of graduates' roles and effectiveness in SPI initiatives. Collaborations between universities and industry partners will be essential to validate and refine the framework, ensuring that graduate software engineering education remains aligned with evolving professional practice.

REFERENCES

1. Alenezi, M. (2025). A framework to evaluate software engineering program using SWEBOK v4. TEM Journal, 14(1), 1–12. https://doi.org/10.18421/TEM141-01
2. Alenezi, M., & Qureshi, M. R. J. (2025). A framework to evaluate software engineering program using SWEBOK v4. TEM Journal, 14(1), 1–12. https://doi.org/10.18421/TEM141-01

3. Laporte, C. Y., & O'Connor, R. V. (2016). Software process improvement in industry in a graduate software engineering curriculum. In R. V. O'Connor, A. Mitasiūnas, R. V. Messnarz, & A. Kaindl (Eds.), Systems, software and services process improvement (pp. 1–12). Springer. https://doi.org/10.1007/978-3-319-30264-8_1
4. Laporte, C. Y., & O'Connor, R. V. (2015). Software process improvement in graduate software engineering programs. In R. V. O'Connor, A. Mitasiūnas, R. V. Messnarz, & A. Kaindl (Eds.), Systems, software and services process improvement (pp. 1–12). Springer. https://doi.org/10.1007/978-3-319-24647-8_1
5. Mecs Press. (Publisher). (2021).
6. Razali, R., & Hashim, N. L. (2021). A conceptual framework for software engineering education: Project based learning approach integrated with industry collaboration. International Journal of Engineering and Manufacturing (IJEME), 11(5), 56–68. https://doi.org/10.5815/ijeme.2021.05.05
7. NIX United. (2023, April 5). Cross-functional teams in software development: Roles, responsibilities & examples. https://nix-united.com/blog/cross-functional-teams-in-software-development-principles-and-examples/
8. Net Solutions. (2026, February 1). Product mindset over project mindset: Benefits and roadmap. https://www.netsolutions.com/insights/product-mindset-vs-project-mindset/
9. Nguyen-Duc, A., Cruzes, D. S., & Abrahamsson, P. (2021). The development and validation of a framework for software engineering education in a startup context. IEEE Global Engineering Education Conference (EDUCON), 170–177. https://doi.org/10.1109/EDUCON46332.2021.9453997
10. TechTarget. (2025, May 27). Product vs. project mindset in software development. https://www.techtarget.com/searchsoftwarequality/feature/Compare-a-product-vs-project-mindset-for-software-development
11. Vajpai, J., & Magda, M. (2023, June 24). A novel interdepartmental approach to teach cross-functional collaboration in software engineering. In Proceedings of the 2023 ASEE Annual Conference & Exposition. American Society for Engineering Education. https://peer.asee.org/a-novel-interdepartmental-approach-to-teach-cross-functional-collaboration-in-software-engineering