

Survey Report: The Impact of Generative AI on Software Engineering Education

Executive Summary

This survey investigated how academic partners and educators perceive the impact of Generative Artificial Intelligence (GenAI) on software engineering education. The survey collected 15 responses from academic stakeholders involved in software engineering teaching and curriculum development.

The results indicate that Generative AI is already substantially transforming software engineering education. Educators increasingly integrate AI into teaching activities, assessment formats, and curriculum design. However, the survey also reveals significant concerns regarding students' overreliance on AI tools, declining depth of understanding, and the challenge of assessing individual competencies in AI-assisted learning environments.

Rather than viewing AI primarily as a replacement for software engineers, respondents largely perceive AI as a catalyst for changing professional roles and educational priorities. Critical thinking, validation of AI-generated results, architectural reasoning, ethical awareness, and risk assessment emerge as key competencies for future software engineers.

1. Key Findings

1.1 Generative AI is Already Reshaping Teaching

The vast majority of respondents reported that Generative AI has influenced their teaching practices. While only one respondent reported no noticeable impact, 90% indicated moderate to fundamental changes in course delivery. Particularly noteworthy is that 50% reported moderate integration of AI tools into teaching activities, while another 50% described significant redesigns or fundamental transformations of their courses.

These results demonstrate that AI adoption within software engineering education is no longer experimental but has become an established reality.

1.2 Students Are Becoming More Tool-Oriented

Respondents consistently observed changes in student behaviour.

The most frequently reported observations were:

- Increased dependency on AI tools (60%)
- Reduced depth of understanding despite improved efficiency (50%)
- Increased uncertainty regarding future professional relevance (50%)

Only one respondent reported increased motivation and engagement as a dominant effect.

This suggests that educators perceive a growing risk that students may focus on obtaining solutions rather than understanding underlying principles.

1.3 Traditional Assessment Methods Are Under Pressure

Assessment practices are evolving rapidly in response to AI-assisted work.

The most common assessment approaches include:

- Assignments with explicit AI usage guidelines (70%)
- Oral examinations and live coding sessions (60%)
- AI-aware assessment formats (40%)
- Project-based assessments with process tracking (40%)

The results indicate a shift away from evaluating only final artifacts toward evaluating understanding, reasoning processes, and decision making.

1.4 AI Is Expected to Transform Rather Than Replace Software Engineering

Respondents generally reject the idea that AI will simply eliminate software engineering jobs.

Instead:

- 70% believe new roles and competencies will emerge.
- 60% communicate that AI will augment rather than replace developers.
- 50% acknowledge that the long-term impact remains uncertain.

The dominant perception is therefore one of professional evolution rather than displacement.

1.5 Curriculum Adaptation Is Considered Essential

When asked about priorities for iSAQB and academic partners, respondents strongly emphasized:

- Integration of AI topics into software engineering curricula (80%)
- Development of best practices for AI in software architecture (80%)
- Support for educators through training and workshops (50%)

This demonstrates broad consensus that AI-related competencies should become an integral part of software engineering education.

1.6 Critical Evaluation Is More Important Than Prompt Engineering

One of the most interesting findings concerns competency gaps.

Respondents identified the following as the most important deficiencies among students and junior software engineers:

- Critical evaluation and validation of AI-generated results (70%)
- Integration of AI into structured development workflows (60%)
- Understanding limitations and risks of AI systems (40%)
- Adapting software architectures for AI-enabled systems (40%)

Notably, prompt engineering itself received no votes as a major competency gap.

This suggests that educators increasingly view prompt engineering as a tactical skill, whereas critical evaluation and engineering judgment are regarded as strategic competencies.

1.7 Ethical Responsibility Is Viewed as Fundamental

Ethics emerged as one of the strongest themes throughout the survey.

The most relevant ethical concerns identified were:

- Accountability and responsibility (90%)
- Transparency and explainability (60%)
- Security and misuse risks (60%)

Furthermore, all respondents agreed that ethical considerations should be a mandatory professional responsibility for software engineers working with AI systems.

This represents a remarkable level of consensus.

2. Discussion

2.1 The Role of Software Engineering Is Shifting

The survey suggests that software engineering education is moving from a code-centric paradigm toward a judgment-centric paradigm.

Historically, educational success was often associated with the ability to produce correct software artifacts. In the age of Generative AI, the ability to critically evaluate, validate, govern, and integrate AI-generated outputs becomes increasingly important.

This transition mirrors broader industry developments where software engineers are expected to supervise AI-assisted processes rather than manually produce every implementation detail.

2.2 The Emerging Assessment Challenge

One of the most significant challenges highlighted by respondents is the difficulty of reliably assessing student competence when AI tools are widely available.

Traditional take-home assignments no longer guarantee that the submitted work reflects the student's own understanding. Consequently, educators increasingly rely on oral examinations, project defenses, and process-oriented assessments.

Future educational models will likely need to focus more strongly on assessing reasoning, architectural thinking, and decision-making capabilities rather than only technical outputs.

2.3 The Risk of Superficial Learning

A recurring theme throughout the survey is the concern that AI may accelerate task completion while simultaneously reducing conceptual understanding.

Students may successfully generate code, architectures, or documentation without fully understanding the underlying principles.

This creates a potential educational paradox: AI can increase productivity while simultaneously weakening deep learning if used without appropriate pedagogical safeguards.

The challenge therefore lies not in restricting AI but in designing educational approaches that preserve critical thinking and conceptual understanding.

2.4 Prompt Engineering May Not Become a Separate Discipline

Interestingly, respondents expressed substantial uncertainty regarding whether prompt engineering will evolve into an independent software engineering discipline.

Most respondents selected either "uncertain" or "unlikely".

This finding aligns with the observation that prompt engineering itself was not identified as a major competency gap. Instead, respondents appear to view prompt engineering as one component of a broader set of AI engineering skills rather than as a standalone discipline.

2.5 Ethics Is Becoming an Engineering Discipline

The survey indicates that ethical considerations are increasingly viewed as an engineering responsibility rather than merely a legal or organizational concern.

Software engineers are expected not only to develop functional systems but also to understand transparency, accountability, safety, fairness, and societal impact.

This suggests that ethical reasoning should become a recurring theme throughout software engineering curricula rather than being treated as an isolated topic.

3. Limitations

Several limitations should be considered when interpreting the results.

First, the survey is based on a relatively small sample of respondents. Consequently, the findings should be interpreted as exploratory rather than statistically representative.

Second, all respondents are associated with software engineering education and iSAQB-related activities. Their perspectives may therefore differ from those of educators in other disciplines or industrial practitioners.

Third, the survey captures perceptions and expectations rather than objective measurements of educational outcomes. Some observed effects, such as reduced depth of understanding or increased dependency on AI tools, require further empirical investigation.

4. Recommendations and Future Opportunities

Based on the survey findings, the following recommendations can be derived.

Recommendation 1: Integrate AI Across Existing Software Engineering Topics

AI should not be taught solely in isolated courses. Instead, AI-related aspects should be incorporated into requirements engineering, software architecture, software testing, quality assurance, and project management.

Recommendation 2: Prioritize Critical Evaluation Skills

Educational programs should place greater emphasis on evaluating, validating, and challenging AI-generated outputs. Critical assessment skills may become more important than prompt construction itself.

Recommendation 3: Redesign Assessment Strategies

Institutions should increasingly employ oral examinations, project defenses, design reviews, and reflective reports that make student reasoning visible.

Recommendation 4: Develop AI-Oriented Architecture Guidance

Professional organizations such as iSAQB should consider developing best practices and educational materials addressing AI-enabled software architectures, governance mechanisms, and quality attributes.

Recommendation 5: Strengthen Ethical and Risk-Awareness Education

Future software engineers should receive systematic training in accountability, transparency, explainability, risk assessment, and regulatory compliance.

Recommendation 6: Support Educators

Universities and professional organizations should provide teaching materials, assessment templates, training programs, and communities of practice to support educators in adapting to AI-driven educational changes.

Conclusion

The survey reveals that Generative AI is already fundamentally influencing software engineering education. While respondents recognize significant opportunities for increased productivity and new forms of learning, they also express concerns regarding superficial learning, assessment integrity, and overreliance on AI tools. The findings suggest that future software engineering education should focus less on the production of software artifacts and more on critical evaluation, architectural reasoning, ethical responsibility, and governance of AI-assisted development processes. These competencies are likely to define the next generation of software engineers and software architects.