Human Factors, Business Management and Society, Vol. 215, 2026, 305–313
https://doi.org/10.54941/ahfe1007611

Educating for the Unknown: ICT Curricula in a Rapidly Shifting Software Future

Noora Nieminen, Paula Steinby, and Tuomo Helo

Turku University of Applied Sciences, Turku 20520, Finland

ABSTRACT

The rapid transformation of software development—driven by generative AI, AI-augmented workflows, low-code/no-code environments, and the early emergence of quantum computation—challenges the structural stability of ICT curricula. As introductory programming tasks become increasingly automatable, traditional curriculum models risk reactive reform rather than systematic adaptation. This paper proposes a layered curriculum architecture designed to support resilience under conditions of paradigm-level technological uncertainty. Grounded in prior research on AI-assisted programming and LCNC-based CDIO integration, and informed by institutional curriculum mapping within ICT degree programs, the study develops a four-layer educational ecosystem model. The model integrates foundational programming competencies, structured AI-augmented workflows, experimental studio-based environments, and introductory quantum literacy modules. Rather than organizing curricula around specific tools, the framework emphasizes modularity, paradigm agility, and durable cognitive skills such as abstraction, verification, and metacognitive awareness. The proposed architecture enables incremental innovation through pilot modules while preserving long-term structural coherence. By shifting the focus from technology prediction to systematic adaptability, the paper contributes a design-oriented model for future-resilient ICT education.

Keywords: AI-assisted coding, Vibe coding, Quantum computation, ICT curriculum design, Pedagogical sustainability, Curriculum innovation, Adaptive learning, Future skills

INTRODUCTION

Recent advances in generative artificial intelligence have disrupted foundational assumptions in programming education. Large language models can solve most introductory programming tasks, challenging the traditional emphasis on syntax acquisition and manual implementation (Denny et al., 2024; Prather et al., 2025). This development has triggered debate within computing education: whether AI tools should be restricted to preserve learning integrity, or systematically integrated to prepare students for evolving professional practices (Prather et al., 2024).

Research increasingly suggests that the central pedagogical shift is not about tool access, but about redefining learning objectives. As the programmer's role evolves from code author to workflow orchestrator, students must develop competencies in problem decomposition, prompt specification, verification, and critical evaluation of AI-generated output (Bull & Kharrufa, 2023; Prather et al., 2023). Without structured scaffolding, however, AI integration

Received February 19, 2026; Revised April 2, 2026; Accepted April 16, 2026; Available online July 20, 2026

risks cognitive offloading, where learners bypass essential reasoning processes and fail to internalize core concepts (Avouris et al., 2025).

At the same time, emerging computational paradigms further destabilize curriculum assumptions. Quantum computing introduces a fundamentally different computational paradigm compared to classical and AI-augmented programming. Emerging research demonstrates that quantum computing concepts can be meaningfully integrated into higher education curricula through scaffolded and modular approaches that support quantum literacy without requiring full specialization in quantum physics (Mjeda & Murray, 2024). Although large-scale quantum hardware remains limited, accessible simulator environments make early exposure increasingly feasible. Rather than producing quantum specialists, undergraduate ICT programs may aim to cultivate quantum literacy—conceptual awareness that supports cross-paradigm reasoning.

Taken together, these developments indicate a broader structural challenge: ICT curricula are traditionally organized around relatively stable technological paradigms, yet software development is entering a phase of sustained paradigm-level transformation. This raises a fundamental question:

How Can ICT Curricula be Designed to Remain Resilient Under Conditions of Rapid and Uncertain Technological Change?

This paper addresses this question by proposing a layered curriculum architecture that separates durable cognitive foundations from rapidly evolving tools, enabling systematic adaptability rather than reactive reform. From a human factors perspective, this architectural approach addresses cognitive load distribution across the curriculum: by anchoring high-load conceptual work in stable foundational layers, it reduces the risk of learner disorientation during periods of rapid technological transition.

LITERATURE REVIEW

Recent advances in generative AI, AI-augmented development processes, experimental studio practices, and quantum computing collectively challenge established assumptions in ICT education. Although each of these domains has been examined separately within computing education research, their combined implications for curriculum architecture remain insufficiently theorized. Together, they exert structural pressure on curricula that were designed for relatively stable technological paradigms.

This section synthesizes research across four interconnected strands to clarify how these developments reshape expectations for programming education and to motivate the need for more resilient curriculum models.

Generative AI in Programming Education

Generative AI has disrupted programming education by enabling large language models to solve many introductory tasks, thereby challenging assessment practices and the traditional emphasis on syntax mastery (Denny et al., 2024). A key concern in the literature is cognitive offloading, where learners rely on AI outputs without engaging in underlying reasoning processes (Avouris et al., 2025). Consequently, the debate has shifted from "ban versus allow" toward structured pedagogical integration that safeguards conceptual understanding while acknowledging emerging professional realities (Prather et al., 2023, 2024).

Research increasingly characterizes the programmer's evolving role as orchestration rather than manual implementation. This shift requires competencies in problem decomposition, specification, verification, and iterative refinement of generated artifacts (Bull & Kharrufa, 2023). Case studies suggest that maintaining rigor depends on redesigning assignments, strengthening code review and testing practices, and reframing learning objectives around evaluation rather than production alone. Pilot implementations in higher education further demonstrate that elective modules can function as controlled environments for experimentation prior to curriculum-wide adoption.

AI-Augmented Workflows and Orchestration

Beyond classroom policy debates, literature in software engineering education emphasizes workflow literacy in AI-augmented development. Effective use of AI assistants requires systematic prompt design, incremental refinement, and structured validation of generated code. Rather than diminishing skill requirements, AI integration shifts emphasis toward testing, quality assurance, and critical evaluation.

From a socio-technical perspective, AI-assisted coding represents a human-in-the-loop paradigm in which responsibility for design intent, integration, and risk management remains with the developer. Educationally, this implies that curricula should explicitly teach students how to manage AI-augmented pipelines, including identifying failure modes, verifying correctness, and integrating generated components into coherent systems.

Vibe Coding and Experimental Studios

"Vibe coding" describes an emerging AI-native development practice centered on conversational prompting, rapid iteration, and intent-driven experimentation. Early research suggests that such workflows prioritize flow and speed but may risk superficial understanding if not supported by verification discipline.

Pedagogically, vibe coding aligns with studio-based learning models that emphasize iteration, critique, and reflective practice. Studio environments allow students to prototype quickly while engaging in structured feedback cycles. When embedded within CDIO-aligned project frameworks, these environments can serve as innovation incubators that encourage creativity without sacrificing program-level coherence.

Quantum Computing in ICT Education

Quantum computing introduces a non-classical computational paradigm grounded in superposition and probabilistic reasoning, thereby expanding the conceptual landscape of programming. Educational research acknowledges the difficulty of integrating quantum topics for students lacking advanced mathematical backgrounds. Scaffolded and layered approaches have been proposed to bridge conceptual gaps and ground quantum reasoning in classical analogies (Mjeda & Murray, 2024).

Although industrial demand remains emergent, accessible simulation environments support early exposure. Rather than training quantum specialists, undergraduate programs may aim to cultivate quantum literacy: an awareness of alternative computational models and their implications for future software ecosystems. This literacy-oriented approach emphasizes conceptual flexibility over technical depth.

Supporting this literacy-oriented approach, Seskir et al. (2022) identify conceptual awareness of quantum principles—rather than computational depth—as the appropriate target for broad undergraduate exposure, noting that quantum literacy supports cross-disciplinary reasoning even in contexts where quantum hardware remains inaccessible. Similarly, Gambetta et al. (2023) describe accessible cloud-based quantum simulators as sufficient infrastructure for introductory-level engagement, consistent with the TUAS elective model described in this paper.

Synthesis

Across generative AI, AI-augmented workflows, vibe-coding studios, and quantum education, a common pattern emerges: the locus of expertise is shifting from tool execution toward abstraction, verification, and cross-paradigm reasoning. These developments challenge curricula organized around stable technologies and isolated skill acquisition.

The literature, therefore, motivates curriculum architectures that preserve durable cognitive foundations while enabling structured experimentation and modular renewal. Such architectures must support transitions between paradigms rather than incremental adaptation to individual tools.

Case Context and Methodology: Institutional Context - Turku University of Applied Sciences

Turku University of Applied Sciences (TUAS) offers multiple degree pathways in information and communication technology (ICT) designed to prepare students for professional software development and digital transformation roles. The core programs include the Bachelor of Engineering in ICT (offered in both Finnish and English) and the Bachelor of Business Administration in ICT, each comprising foundational software engineering, systems design, and professional practice modules structured around CDIO (Conceive–Design–Implement–Operate) principles. Within the TUAS ICT programs, CDIO-aligned pedagogy is formally embedded in core courses, particularly *Foundational ICT and Study Skills* and *Low-Code Software Development Basics*. These courses provide a structured environment in which AI-assisted and LCNC-supported learning practices can be iteratively developed and evaluated without requiring structural curriculum redesign.

Within these degree structures, several elective and pilot modules have been introduced to address emerging technological trends:

- AI in Software Development – an elective course focusing on integrating AI tools into software workflows.
- Python Programming Project with AI – a project-based elective where students apply AI-assisted development tools in authentic coding tasks.
- Introduction to Quantum Computing – an introductory elective exposing students to quantum concepts and simulator-based programming environments.

These elective studies serve both as curricular innovations and as controlled environments for experimenting with emerging tools and practices before considering broader integration into mandatory coursework.

Methodological Approach

This study adopts a design-based research (DBR) approach combined with curriculum mapping within the TUAS ICT context. The objective is not to conduct an empirical evaluation of student outcomes, but to develop a conceptually grounded and practically informed curriculum design model that supports resilience under conditions of rapid technological change.

The research process involved several interrelated activities:

1. Systematic analysis of existing ICT curricula: The B.Eng. and BBA program structures were systematically reviewed to identify current course sequences, learning outcomes, and competency frameworks.
2. Mapping of AI-related and emerging technology modules: Elective modules addressing AI-assisted development, LCNC (low-code/no-code) tools, and quantum computing were mapped against core program outcomes to locate potential integration points.
3. Alignment with CDIO principles: Integration opportunities were evaluated in relation to CDIO standards, emphasizing experiential learning, iterative design, and authentic problem solving.
4. Synthesis of prior research and pilot insights: Findings from earlier work on AI-assisted programming and LCNC integration within CDIO contexts (Nieminen & Reunanen, 2024; Nieminen & Reunanen, 2025) were synthesized with recent literature on vibe coding and quantum literacy to inform a resilient curriculum framework.

Concretely, the curriculum mapping covered twelve core modules and five elective modules across the two TUAS ICT degree programs (B.Eng. and BBA). Each module was reviewed for its stated learning outcomes, assessment methods, and CDIO stage alignment. Mapping was conducted by two of the authors independently and then reconciled through structured discussion to reduce individual bias. Integration points for emerging technologies were identified where elective content reinforced, extended, or introduced competencies absent from mandatory coursework. This process yielded the layered architecture, which we will now describe.

This methodological orientation emphasizes iterative refinement of design concepts in dialogue with both institutional practice and broader research evidence. The proposed framework extends beyond tool-specific recommendations toward structural curriculum strategies that preserve durable cognitive competencies while enabling systematic adaptation to novel paradigms.

Curriculum Structure, Learning Outcomes and CDIO Integration

To clarify how emerging technologies are currently positioned within the TUAS ICT programs, Table 1 distinguishes between foundational core modules and experimental elective modules.

Table 1: Core and emerging curriculum components in TUAS ICT programs.

Curriculum Layer	Core Modules (Mandatory)	Emerging / Elective Modules
Programming Foundations	Introductory programming, Data structures, Software architecture	Python Programming Project with AI
Software Engineering	Systems design, Testing, DevOps basics	AI in Software Development
Project-Based Learning	CDIO-based implementation projects	LCNC / experimental studio modules
Advanced Paradigms	—	Introduction to Quantum Computing

Core modules emphasize stable competencies (algorithmic reasoning, structured development, system integration), while electives function as innovation spaces where emerging paradigms can be explored without destabilizing the curriculum. This layered positioning supports controlled experimentation prior to broader structural reform.

To avoid tool-centric integration, emerging modules were mapped against program-level learning outcomes. Table 2 illustrates how AI-, vibe-, and quantum-related modules reinforce durable competencies rather than replace them.

Table 2: Alignment of emerging modules with core competencies.

Durable Competency	AI-Assisted Programming	Vibe / Studio Learning	Quantum Literacy
Abstraction	Prompt design & specification	Intent-driven prototyping	Conceptual modeling of qubits
Verification	Code review & testing of AI output	Iterative critique cycles	Circuit simulation validation
Problem Decomposition	Task structuring for LLMs	Feature-based iteration	Algorithmic reasoning beyond determinism
Systems Thinking	Integration of generated modules	Collaborative artifact design	Awareness of alternative computational models
Metacognition	Reflection on AI limitations	Feedback-driven iteration	Cross-paradigm comparison

Emerging technologies do not introduce entirely new competencies; rather, they intensify the importance of abstraction, verification, and systems-level reasoning.

The TUAS ICT programs follow CDIO principles. Emerging technologies were analyzed through the CDIO lifecycle to evaluate structural compatibility.

Table 3: CDIO integration of AI, vibe coding, and quantum modules.

CDIO Phase	AI-Augmented Workflows	Vibe Coding Studios	Quantum Modules
Conceive	Problem framing for AI-assisted development	Ideation through conversational prompting	Conceptual comparison of classical vs quantum logic
Design	Prompt engineering, architecture planning	Rapid prototyping & feature iteration	Quantum circuit design in simulators
Implement	AI-assisted code generation with verification	Collaborative build-first implementation	Basic quantum algorithm simulation
Operate	Testing, integration, lifecycle management	Iterative refinement & user feedback	Reflection on scalability & paradigm limits

AI and vibe coding strengthen CDIO implementation and iteration phases, while quantum literacy primarily enriches the conceptual (Conceive/Design) phases by expanding abstraction frameworks.

DISCUSSION

The proposed layered curriculum architecture offers a structured yet adaptable approach to ICT education in the face of technological uncertainty. Its primary strength lies in separating durable cognitive foundations from rapidly evolving technological layers. By insulating core competencies—such as abstraction, decomposition, and verification—from short-term technological fluctuation, the model supports long-term curricular coherence.

At the same time, integrating AI-augmented workflows, experimental studio environments, and quantum literacy modules enables controlled innovation. Rather than replacing foundational programming education, emerging technologies are positioned as extensions that intensify higher-order reasoning skills. This alignment with CDIO principles further strengthens the framework by grounding experimentation in structured, experiential learning processes.

However, implementing such a model is not without challenges. Faculty development becomes critical, as instructors must acquire sufficient literacy in AI-assisted tools, vibe-coding environments, and quantum simulators to scaffold learning effectively. Institutions with limited resources may also face constraints in maintaining modular updates and experimental learning spaces.

Strategically, the architecture reframes adaptability as a core educational objective. Instead of preparing students for specific technologies, it cultivates paradigm agility—the ability to operate across classical, AI-augmented, and emerging computational models. Early exposure to quantum concepts, for example, is not justified by immediate labor-market demand but by its contribution to conceptual flexibility and abstraction skills.

Nevertheless, systemic risks remain. Technological hype cycles may encourage premature curricular shifts, potentially overshadowing foundational learning. Without careful scaffolding, modular layers could fragment rather than reinforce program coherence. The framework, therefore, depends on deliberate integration and sustained institutional reflection.

Ultimately, curriculum resilience cannot be achieved through isolated tool adoption. It requires architectural thinking: designing educational structures capable of absorbing paradigm shifts without losing conceptual continuity.

CONCLUSION

Rapid advances in generative AI, AI-augmented development environments, and emerging computational paradigms such as quantum computing are reshaping the conditions under which ICT education operates. These developments do not merely introduce new tools; they destabilize assumptions about programming practice, assessment, and curriculum structure.

This paper has argued that reactive adaptation to individual technologies is insufficient. Instead, ICT programs require architectural approaches that distinguish between durable cognitive foundations and evolving technological interfaces. Based on literature synthesis and curriculum mapping within the TUAS ICT programs, we proposed a layered curriculum architecture consisting of foundational competencies, AI-augmented orchestration practices, experimental studio environments, and exploratory frontier modules.

The framework demonstrates that resilience in ICT education is not achieved through the prediction of future technologies, but through a structural design that supports paradigm transitions. By preserving abstraction, verification, and systems thinking as core competencies while enabling modular experimentation, institutions can maintain coherence amid technological acceleration.

Preparing students for the unknown, therefore, demands more than technological literacy. It requires cultivating epistemic flexibility—the capacity to reason across computational paradigms and adapt to shifting development practices. Curriculum resilience emerges not from stability of tools, but from stability of conceptual foundations combined with structured adaptability. In human factors terms, the framework prioritizes learner-centered design: matching the complexity of the educational environment to the evolving cognitive demands of professional software practice, and ensuring that curriculum architecture itself functions as a resilience-supporting scaffold rather than a source of additional uncertainty.

REFERENCES

Avouris, N., Sgarbas, K., Caridakis, G., & Sintoris, C. (2025). *Teaching introduction to programming in the times of AI: A case study of a course redesign*. arXiv. https://doi.org/10.48550/arXiv.2508.06572

Bull, C., & Kharrufa, A. (2023). Generative artificial intelligence assistants in software development education: A vision for integrating generative artificial intelligence into educational practice, not instinctively defending against it. *IEEE Software, 41*(2), 52–57. https://doi.org/10.1109/MS.2023.3300574

Denny, P., Prather, J., Becker, B. A., Finnie-Ansley, J., Hellas, A., Leinonen, J., Luxton-Reilly, A., Reeves, B. N., Santos, E. A., & Sarsa, S. (2024). Computing education in the era of generative AI. *Communications of the ACM, 67*(2), 56–64. https://doi.org/10.1145/3624720

Gambetta, J. M., Chow, J. M., & Totzauer, M. (2017). Building logical qubits in a superconducting quantum computing system. npj Quantum Information, 3(1), 2. https://doi.org/10.1038/s41534-016-0004-0

Mjeda, K., & Murray, L. (2024). *Teaching quantum computing to undergraduate students: A layered, scaffolded approach*. arXiv. https://arxiv.org/abs/2405.09265

Nieminen, N., & Reunanen, T. (2024). Artificial intelligence as a catalyst: A case study on adaptive learning in programming education. *Human Factors, Business Management and Society, 135*, 303–312. https://doi.org/10.54941/ahfe1004957

Nieminen, N., & Reunanen, T. (2025). Combining AI tools, low-code platforms, and product development in ICT education: A reflective study on educational and practical outcomes. *Human Factors, Business Management and Society, 176*, 228–241. https://doi.org/10.54941/ahfe1006320

Prather, J., Denny, P., Leinonen, J., Becker, B. A., Albluwi, I., Craig, M., Keuning, H., Kiesler, N., Kohn, T., Luxton-Reilly, A., MacNeil, S., Petersen, A., Pettit, R., Reeves, B. N., & Šavelka, J. (2023). The robots are here: Navigating the generative AI revolution in computing education. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education (SIGCSE '23)* (pp. 108–114). ACM. https://doi.org/10.1145/3623762.3633499

Prather, J., Leinonen, J., Kiesler, N., Benario, J. G., Lau, S., MacNeil, S., Norouzi, N., Opel, S., Pettit, V., Porter, L., Reeves, B. N., Šavelka, J., Smith, D. H., Strickroth, S., & Zingaro, D. (2024). Beyond the hype: A comprehensive review of current trends in generative AI research, teaching practices, and tools. *ACM Transactions on Computing Education*. https://doi.org/10.1145/3689187.3709614

Sengul, C., Neykova, R., & Destefanis, G. (2024). Software engineering education in the era of conversational AI: Current trends and future directions. *Frontiers in Artificial Intelligence, 7*, 1436350. https://doi.org/10.3389/frai.2024.1436350

Seskir, Z. C., Migdal, P., Weidner, C., Anupam, A., Case, N., Davis, N., Decaroli, C., Ercan, I., Foti, C., Gora, P., Jankiewicz, K., Johnson, B. R., Kohnle, A., Krueger, A., Langford, N., Macaluso, A., Plewa, M., Smith, F. A., Surer, E., … Chiofalo, M. L. (2022). Quantum games and interactive tools for quantum technologies outreach and education. Optical Engineering, 61(8), 081809. https://doi.org/10.1117/1.OE.61.8.081809

Thorne, D. J., & Sarkar, D. (2025). Leveraging test-driven development with large language models for reliable and verifiable spreadsheet code generation: A research framework. arXiv. https://doi.org/10.48550/arXiv.2510.15585