BALANCING FOUNDATIONAL KNOWLEDGE AND REAL-WORLD PRACTICE IN COMPUTING EDUCATION DURING TECHNOLOGICAL DISRUPTION

by

JON VENTULETT

B.S., Westfield State University, 2006

M.S., Rensselaer Polytechnic Institute, 2009

A Research Paper Submitted to the School of Computing Faculty of

Middle Georgia State University in

Partial Fulfillment for the Requirements for the Degree

DOCTOR OF SCIENCE IN INFORMATION TECHNOLOGY

MACON, GEORGIA

2026

Balancing foundational knowledge and real-world practice in computing education during technological disruption

Jon Ventulett, ***Middle Georgia State University,*** ***Jon.Ventulett@MGA.edu***

Abstract

This qualitative practitioner study examines how undergraduate computing faculty, early career graduates, and hiring managers interpret foundational knowledge and real-world development practice during a period of technological disruption. Using semi-structured interviews and thematic analysis, the study explores how stakeholders define, experience, and evaluate this balance across the education-to-employment pipeline. Findings suggest that foundational knowledge is understood less as a fixed technical skill set and more as a transferable mindset involving problem-solving, adaptability, independence, and conceptual understanding. Real-world practice is described as performance within workplace constraints, including existing systems, team workflows, organizational expectations, and professional judgment. Across stakeholder groups, balance is not experienced as a fixed formula. Faculty construct it within institutional limits, graduates encounter it as a transition gap, and hiring managers evaluate it through workplace performance. Generative AI further complicates this relationship by increasing the importance of reasoning, validation, and evaluative judgment. The study points to the need for stronger alignment between undergraduate preparation and entry-level professional expectations.

Keywords*:* computing education, foundational knowledge, curriculum design, generative AI, workforce readiness

Introduction

Problem Statement

Undergraduate computing education finds itself in a period of intense technological disruption not seen in decades (ABET, 2024). The forces of automation and generative AI, coupled with the rapid shifts in professional development practices that have emerged from these changes, have left undergraduate computing programs and educators scrambling to maintain curricula that adequately prepare students for the occupational landscape they will soon enter. Teaching-focused institutions, especially those emphasizing direct industry readiness, are increasingly tasked with preparing students with both a solid theoretical grounding and a job-ready skill set. However, there is little consensus on how to achieve this balance (Karataş et al., 2025). Foundational theory, coupled with a heavy complement of mathematics, has long been the emphasis of Computer Science (CS) programs, whereas Information Technology/Information Systems (IT/IS) programs tend to emphasize hands-on techniques and proficiency with industry tools. However, as emerging technologies increasingly blur the line between these domains, many institutions are struggling to define what “real-world readiness” truly is and whether their programs can, or should, evolve at a pace approaching the technologies themselves. Despite this ongoing debate, few studies have examined how specific stakeholders, such as faculty, early career graduates, and hiring managers working with entry-level talent, interpret and navigate these pedagogical and curricular decisions and expectations in day-to-day practice. This knowledge gap leaves program coordinators and department chairs without clear guidance on how to address the gap between academic preparation and workplace expectations while still maintaining the discipline’s longstanding intellectual core. As a result, some scholars have advocated for the creation of conceptual frameworks to aid curriculum and instructional decision-making when integrating rapidly evolving technologies into curricula (Dexter, 2023).

Purpose of the Study

The objective of this qualitative study is to explore how key stakeholders across the education-to-employment pipeline (undergraduate faculty, early-career graduates, and hiring managers) conceptualize and evaluate the balance between foundational knowledge and real-world development practice amid rapid technological change. Specifically, this study seeks to document how these stakeholders perceive the gap between undergraduate computing preparation and workplace expectations, including how they define "foundational" versus "real-world" knowledge and skills, how they prioritize competencies for entry-level readiness, and where perspectives align or diverge across the pipeline. Generative AI tools are also examined as an important contextual factor shaping expectations and learning practices. Through semi-structured interviews and thematic analysis, the study will produce a cross-stakeholder view of the foundations/practice balance and identify implications for feasible instructional and programmatic adjustments within teaching-focused CS/IT education. Conceptually, the study is informed by Diffusion of Innovations theory as a lens for understanding how new tools and practices are adopted and normalized within educational and professional contexts (Rogers, 2003).

This study will examine the following primary research question and supporting subquestions:
How do key stakeholders across the education-to-employment pipeline perceive the gap between undergraduate computing preparation and workplace expectations, and what themes emerge regarding challenges, shaping factors, and possible solutions?
SQ1: How do key stakeholders across the education-to-employment pipeline (faculty, early career graduates, and hiring managers) define and evaluate the balance between foundational knowledge and real-world development practice during a period of intense technological disruption?
SQ2: Where do stakeholder perspectives align or diverge regarding entry-level job readiness, and what contextual factors appear to shape these perspectives (e.g., role type, institutional constraints, workplace expectations)?
SQ3: What factors do stakeholders describe that shape how foundational knowledge and real-world practices are taught, learned, and evaluated across undergraduate programs and early-career roles?
SQ4: What strategies or recommendations do stakeholders propose to improve alignment between undergraduate preparation and entry-level expectations, including the influence of emerging tools such as GenAI?

Review of Literature

Computing education has long grappled with a core tension between teaching foundational theoretical concepts and providing up-to-date practical skills. This long-standing skills gap between academia and industry has been repeatedly documented for over a decade and remains a significant concern among stakeholders (Radermacher & Walia, 2013; Valstar et al., 2020). Recent graduates often leave their undergraduate education with strong conceptual knowledge but struggle to apply real-world industry tasks and often require significant on-the-job training to acquire missing skills and adapt to company practices and culture (Lundberg et al., 2020). This tension has also been observed internationally, with studies of undergraduate computing programs identifying continued concerns about alignment between curriculum structures and industry expectations (Alabdulkarim et al., 2021). This skill gap spans both specific technical proficiencies (frameworks, tools, cloud technologies) and “soft” skills like communication, teamwork, and project management, which employers increasingly expect in addition to programming expertise.

Faculty Philosophies and Strategies

Faculty often differ in how they navigate the theory-vs-practice balance. One philosophy continues to place emphasis on foundational knowledge, arguing that universities should teach clarity of thought and a true understanding of what is happening behind the scenes, rather than which tool to use.

These educators worry that shifting to every new framework can dilute core computing principles and create a "mile wide, inch deep" skill set. Another common caution is that too much focus on surface-level tools might leave students without deep mastery of underlying concepts (Valstar et al., 2020). Faculty holding this view note that current technologies evolve rapidly and can be learned while working in industry, while a solid foundational grounding in algorithms development and analysis, systems architecture, and math remains vital. However, another group contends that certain modern development practices, such as version control, agile methods, cloud platforms, are now essential knowledge for undergraduates. Proponents of this view argue that integrating such real-world skills throughout the curriculum is crucial to producing job-ready entry-level hires. Recent studies suggest that many industry tools and practices are increasingly difficult to separate from professional computing competence, and that limited exposure to current workflows can leave students with an incomplete understanding of real-world software development (Valstar et al., 2020; Exter et al., 2018). These studies have highlighted gaps between what academia emphasizes and what practitioners value in practice, bolstering the case for greater applied content in courses (Exter et al., 2018; Valstar et al., 2020). In practice, many instructors strive to balance the two schools of thought by teaching theory while offering hands-on projects, while simultaneously acknowledging the zero-sum reality of limited classroom contact hours. Faculty at teaching-focused colleges, who often feel direct pressure to prepare students for employment, tend to place higher value on including current tools and project experiences than research-university faculty (Valstar et al., 2020). However, even these practitioners stress that practical additions should not take precedence over computing fundamentals. The consensus in recent faculty surveys is that both dimensions matter; strong problem-solving and conceptual skills remain the top priority, yet there is a growing expectation that students encounter at least some industry-relevant content by their senior year (Valstar et al., 2020; Prather et al., 2023).

Integrating Real-World Development Practices

In response to the skills gap, educators have experimented with various methods to infuse real-world development experience into the curriculum without sacrificing core theory. Capstone courses, for example, are widely used to simulate industry-like projects, allowing students to apply their theoretical learning to larger-scale, team-based software development tasks. Such capstones and other project-based learning approaches have been shown to increase student engagement and improve job preparedness (Whalley et al., 2025). Beyond capstones, many programs are embedding practical skills training into earlier courses, including tools such as Git and Docker, collaborative workflows, and unit testing. Industry-facing research has identified skills such as version control, continuous integration (CI/CD), debugging, and teamwork as desirable qualifications for entry-level software engineers (Stamm, 2023). Likewise, top-performing software engineers distinguish themselves not only through coding ability, but also through maintainability-oriented decision-making, collaboration-sensitive work habits, and continuous learning (Li et al., 2020). This alignment of curriculum with workplace expectations extends to professional skills development, including communication, project management, and ethical considerations. Student perspectives also suggest interest in clearer guidance on emerging tools, particularly generative AI, and how such tools should be used to support learning rather than bypass it (Zastudil et al., 2023). Such student feedback reinforces the need for curricula to evolve. However, significant challenges persist; faculty must constantly update content in a rapidly and unceasingly changing field. Adding new core content is severely restricted by the limitations of classroom time and resources. The ongoing task for educators is to discover and apply innovative ways to integrate modern practices while maintaining the substantial theoretical foundation that has historically defined the discipline.

Generative AI as a New Disruptor

Emerging generative AI coding tools (e.g., GitHub Copilot, OpenAI's ChatGPT) have recently hastened and prioritized the theory-versus-practice debate. In the last two years, these AI assistants have shown they can automatically generate code and solutions, raising new questions about what computing skills students truly need. Initial faculty reactions to AI tools have been mixed, but largely cautious. Many instructors, concerned about academic integrity and student overreliance, have considered discouraging, restricting, or temporarily banning AI-assisted coding on assignments while they determine how these tools should fit into programming education (Lau & Guo, 2023). Beyond fears regarding cheating, educators are split on the pedagogical response. Some advocate "doubling down" on fundamental concepts and manual coding skills in light of AI. If students can offload coding to a tool, these instructors argue that it is even more critical to solidify human understanding of algorithms, problem-solving, and code reading (Lau & Guo, 2023). This view echoes the traditional stance of prioritizing foundational learning, now as a guard against over-reliance on automation. On the other side, a more integration-oriented view holds that AI tools are becoming part of professional practice, and as such, students may need to learn to work with them as part of their broader development process. Proponents in this camp suggest that outright resistance may be ineffective and that curricula may need to incorporate AI-assisted development alongside classical theory (Lau & Guo, 2023; Prather et al., 2023). For example, some instructors now permit tools like Copilot in assignments but require students to critique or verify the AI's output, turning it into a learning opportunity rather than a shortcut. Others perform portfolio reviews, where students have a chance to explain, answer questions about, and modify their code in real time in order to better simulate the code interview experience and ensure that deep understanding is being developed alongside tool use. Early studies indicate that students are already using these tools and are interested in receiving guidance on their responsible use. In one survey, students had slightly more experience with AI coding assistants than their professors and expressed interest in instruction on responsible use (Prather et al., 2023). Faculty are also rethinking assessment strategies, with some considering more in-person exams, oral explanations, or code reviews to ensure students still attain fundamental skills (Zastudil et al., 2023). In short, generative AI presents a disruptive new element in computing education that both widens the theory–practice divide if students rely on AI for all "practical" coding and offers potential to bridge it if used to support routine work while emphasizing deeper conceptual learning. At the very least, this new paradigm has forced educators to continually recalibrate curriculum priorities, further underscoring that the balance between foundational theory and applied professional skills is a constantly moving target in the face of technological change.

Methodology

This study employed qualitative practitioner research design using semi-structured interviews and researcher reflection. A qualitative practitioner study is an inquiry conducted by a researcher who is also an active practitioner within the professional area being studied, using that experience as a source of contextual insight while managing potential bias through reflexive practice. This approach is appropriate given the open-ended nature of the research questions, which sought to understand how undergraduate computing education navigates the relationship between foundational knowledge and real-world practice amid rapid technological change (Braun & Clarke, 2006). Rather than measuring predefined outcomes or evaluating specific curricula, the study focused on capturing how key stakeholders interpret, navigate, and reason about these instructional tensions in practice. The practitioner research framework was well-suited to this investigation, as the researcher is an active faculty member working within the educational context being examined. This positionality provided relevant contextual insight into instructional decision-making while also requiring intentional reflexivity. Reflexivity was addressed through systematic journaling and transparency in methodological decision-making throughout the study.

Participants and Sampling

The study included 10 adult participants (age 18 or older) drawn from three participant groups: undergraduate computing faculty at teaching-focused four-year institutions, recent graduates (within the past 1-2 years) currently employed in computing-related roles, and industry professionals or hiring managers who work with entry-level computing professionals.

Participants from each group were recruited to provide perspectives across the education-to-employment pipeline. Faculty, recent graduates, and hiring managers were included in comparable numbers to support cross-stakeholder comparison of themes related to preparedness, expectations, and the interpretation of foundational versus applied skills. Participants were selected using purposive convenience sampling based on relevance to the research questions and accessibility to the researcher. All participation was voluntary, and no compensation was provided.

Data Collection

Data was collected through individual semi-structured interviews lasting approximately 30–45 minutes. Interviews were conducted over video conferencing. A semi-structured interview guide was used to ensure consistency across interviews while allowing flexibility for follow-up questions and elaboration. Although the wording of questions varied slightly by participant group (faculty, recent graduates, or hiring managers), all interviews shared a set of core thematic areas, including:

- Definitions of foundational computing knowledge
- Understandings of real-world development practices
- Perceived balance between theory and applied skills
- Decision-making or experiential influences related to curriculum and skill development
- Responses to technological change, including AI-assisted tools

A small number of role-specific questions were included to capture perspectives that do not apply to all groups. Recent graduates and industry professionals were asked to reflect on workforce transitions and preparedness gaps, while faculty were asked to describe the logic and constraints that influence curriculum inclusion decisions. With participant consent, interviews were recorded to ensure accuracy and transcribed verbatim for analysis.

In addition to interview data, a reflective practitioner journal was maintained throughout the study. This journal documented instructional observations, decision points, and reflections related to the research focus. These entries served as a secondary data source and were analyzed alongside interview transcripts to provide contextual grounding and reflexive insight.

Data Analysis

Interview transcripts and reflective journal entries were analyzed using thematic analysis following the six-phase framework described by Braun and Clarke (2006): Familiarization with the data through repeated reading, generation of initial codes across the dataset, identification of broader themes by grouping related codes, review and refinement of themes for coherence and consistency, definition and naming of final themes, and lastly synthesis of findings into a narrative account.

Coding was supported by qualitative data analysis software to assist with organization, traceability, and transparency. To protect confidentiality, participants are referenced using anonymized identifiers indicating stakeholder group and initials: F = faculty, RG = recent graduate, HM = hiring manager (e.g., HM-CJ).

Given the researcher's dual role as practitioner and investigator, reflexivity was intentionally integrated into the analytic process. The reflective journal functioned both as a data source and as a mechanism for examining assumptions, positionality, and interpretive influence.

Results

The results are divided into four key themes that collectively describe how stakeholders define, evaluate, and balance foundational knowledge and real-world development practice during a period of technological disruption. The alignment and divergence of opinions on entry-level readiness, the factors shaping classroom and workplace realities, and the proposed strategies of stakeholders to better align are also discussed. Quotes are attributed using anonymized participant IDs.

Primary Research Question and Summary of Findings

The primary research question of this study was: How do key stakeholders across the education-to-employment pipeline perceive the gap between undergraduate computing preparation and workplace expectations, and what themes emerge regarding challenges, shaping factors, and possible solutions?

The findings indicate that stakeholders perceive the gap less as a simple absence of specific tools or content and more as a problem of translation between academic preparation and workplace performance. Faculty described the challenge primarily through curriculum structure, sequencing, and institutional constraints. Early-career graduates described the gap through the transition into existing codebases, professional workflows, and team-based environments. Hiring managers evaluated the gap through applied judgment, troubleshooting, communication, independence, and performance under real organizational constraints.

SQ1: How do key stakeholders across the education-to-employment pipeline define and evaluate the balance between foundational knowledge and real-world development practice during a period of intense technological disruption?

Stakeholders did not define balance as a fixed ratio between theory and practice. Faculty tended to describe balance as a sequencing problem, with foundational concepts emphasized early and applied practices introduced later through projects, upper-level courses, or capstone work. Early-career graduates described the balance in terms of the transition from coursework to professional expectations, especially the need to work with existing codebases, workflows, and team structures. Hiring managers evaluated balance through workplace performance, particularly whether new hires could reason through unfamiliar problems, communicate effectively, and apply knowledge under realistic constraints.

SQ2: Where do stakeholder perspectives align or diverge regarding entry-level job readiness, and what contextual factors appear to shape these perspectives?

Stakeholders aligned around the importance of problem-solving, adaptability, and conceptual understanding. However, they diverged in how they identified readiness gaps. Faculty emphasized curricular limits, sequencing, and the difficulty of adding applied content without weakening foundations. Early-career graduates pointed to specific gaps in workflow experience, including collaboration, existing codebases, APIs, frameworks, and workplace processes. Hiring managers focused less on specific tools and more on independent reasoning, troubleshooting, communication, and the ability to operate at organizational scale.

SQ3: What factors do stakeholders describe that shape how foundational knowledge and real-world practices are taught, learned, and evaluated across undergraduate programs and early-career roles?

The main factors were curricular structure, course level, faculty autonomy, student readiness, workplace expectations, institutional constraints, and the rapid emergence of generative AI tools. Faculty described balancing these areas within existing course objectives and limited instructional time. Early-career graduates described learning real-world practice through internships or early employment, where they encountered professional workflows not fully replicated in coursework. Hiring managers evaluated readiness through observable workplace behavior, especially the ability to investigate problems, make decisions, and justify actions.

SQ4: What strategies or recommendations do stakeholders propose to improve alignment between undergraduate preparation and entry-level expectations, including the influence of emerging tools such as GenAI?

Stakeholder recommendations pointed toward better integration rather than simply adding more content. Suggested improvements included making the connection between foundational concepts and applied practice more explicit, introducing collaborative workflows earlier, giving students more experience with existing systems and realistic constraints, and designing assessments that require explanation, validation, and accountability. In relation to GenAI, stakeholders emphasized the need to treat tool use as a matter of reasoning and judgment rather than simple access to automated code generation.

Theme 1: Framing "foundations" as a mental model rather than a toolset.

Across all groups, "foundational knowledge" was routinely described more as a mental model rather than a set of specific skills. Reasoning acumen, high-level conceptual understanding, problem-solving strategies, and a baseline ability to tackle unfamiliar problems were described as much more "foundational" than traditional computational areas of study. Faculty SB defined foundations as "a set of core concepts that go beyond programming languages", emphasizing that skills transfer once students "realize that all computers work the same, regardless of programming language, regardless of technology, even technologies that change". Hiring manager CJ prioritized a troubleshooting mindset as the key foundational skill: "working a problem that you do not know the answer to, and that you're not expected to know the answer to... What thought would you process next to try and find an answer, to try and move forward".

Recent graduates grounded foundational skills in the context of being able to work with and adapt to an existing, unfamiliar systems. HG stated the most foundational skill as "being able to look at code that's already there and being able to understand it…You can't just build, obviously you're not just building something from the ground up". Recent graduate TB also pointed to the importance of being able to adapt to an existing code base versus the academic model of creating programs from scratch: "that's essentially what an entire software engineer's job is, unless you're at the architect's level or the senior level where you're doing the creation of stuff".

While mental models were emphasized, specific technologies were referenced, but solely among the faculty group interviewed. Faculty TB referred to the classical Computer Science backbone of "fundamental knowledge of how data structures work, operating systems, compilers…Database is also very important". Faculty GC stated that "hardware is also something that should not be ignored…knowledge like digital logic will be important."

Theme 2: "Real-World Practice" defined as workflow and collaboration

"Real-world development practice", much like "foundational knowledge", was tied much less to specific tools than to collaboration, dealing with problems at scale, organizational process, and performance under pressure. Hiring manager CJ described the contrast as "the difference between sterile, clean environment of a textbook problem…and working that exact same problem while four or five people stand around a nurse's desk staring at you (…) because they need that computer to be online and you're preventing that". The focus on task breakdown versus the more academic "from scratch" design was emphasized by recent graduate TB: "it's a lot of collaboration and a lot of really small bite sized parts…You ask for confirmation for what to do and what the intended design is from tons of different people". Recent graduate HG was one of the few to name a specific technique, albeit a ubiquitous one: "We have a software development lifecycle that uses Agile… that is kind of huge in my role", and that their exposure to it was almost entirely on the job: "I learned more in internships than I did in school".

From the faculty perspective, "real-world practice" was often described as a late-stage concept in curriculum. Faculty BH stated "…it's in a few of my junior, senior level courses, those 300 levels, maybe 400 levels." BH also noted how important students appeared to value this content, even when the evaluation of the resulting work was less enthusiastic: "The students were very happy…the students felt like they were doing something real", even when the project was small in scope and "a little lackluster".

Theme 3: The readiness gap as a skill and scale translation problem

Across hiring manager interviews, job readiness evaluation criteria were less focused on specific tools or skills, and more about whether the employee could bridge the gap between conceptual understanding and actionable results under realistic constraints. When asked how well new hires are prepared for the job, hiring manager SI stated: "Not well. They don't understand how to put the knowledge they gain in college to use…No, it's translating that information, translating that learning into something that your company can use". SI also referenced the level of supervision that was required, and the reluctance of new hires to take ownership of decisions: "They will need handholding through every stage… They don't like to make decisions… They just don't have a good process of how to think through things to come to some conclusions". Hiring manager CJ referred to the readiness gap more as a lack of "real world polish in dealing with people" than a lack of experience with any specific technique or toolset: "I see people who are significantly stronger technically than they are dealing with the real-world problems".

While hiring managers focused on the lack of less tangible interpersonal and decision-making skills, recent graduates noted more specific technologies or technical concepts as the primary hurdle. Recent graduate HG stated, "there was this topic of APIs that I had no idea about…I don't know if it was something I was supposed to be learning that early, but…API is the building block now…like everything it's API to a backend." Recent graduate TB emphasized that undergraduate coursework did not lack in complexity or required rigor, but fell short in regard to replicating real-world programming experience: "I would say the coursework was a lot more complex than what is usually asked of me on a day-to-day basis (…) but one big thing that I felt like I might have lacked during my education was exposure to a large existing code base and being asked to do modifications on it."

Theme 4: GenAI is forcing change in how competence is taught, observed, and evaluated

There was little consensus across stakeholder groups - or even within them - on how AI tools are influencing the new hire skills gap, or how they contribute to the balance of fundamentals versus real-world development practices in undergraduate education. However, there was consensus centered around the idea that GenAI and similar tools increase the need for new hires to be skilled in reasoning, interpretation, and accountability, rather than just code production.

From a faculty perspective, the reactions tended toward nuance rather than hard lines. Faculty SB described a staged policy: "In the beginner courses… I simply don't allow them to use any AI… it's all about restriction… because I want them to gain the core." In advanced contexts, SB shifted to permissive use: "use as much AI as you want… because the problems are so meaningful in real life that in real life they would have to use AI too." Faculty TDB responded by shifting grading weight away from take-home work: "I have reduced the weight from homework assignments and put more weight on in class quiz… those quizzes usually will test their concepts… mostly conceptual questions.” Faculty BH described a resistance to moving to high-stakes written exams as a response to the difficulty in assessing unsupervised work: “I'm not a really believer in… paper and pencil… because it seems so not what you do in computer science or IT… and I don't want to go back to that.”

The employer perspective was almost entirely results-driven. If the employee is producing work that they understand, can explain, and are accountable to, the method of producing it was of secondary concern. Hiring manager TD stated: “when we tell an entry-level person, do not execute until you understand what you're doing. We mean that… I don’t really care what tool you use.” Recent graduates also had varied responses to the effectiveness and impact of GenAI on their new positions. HG strongly resisted copilot-like tools, finding them intrusive: “it writes really, really crappy code… it is not an engineer… it kind of just feels like it's like this guy on my back that’s micromanaging.” Recent graduate TB describes how quickly the workplace response is changing alongside the maturation of GenAI tools: “it's evolving day by day… at least once a week, we have a meeting about AI and new tools that are being developed in different departments that are now available company wide.” TB also describes the disconnect between management and those in technical roles regarding what AI tools can realistically be used for, their reliability, and the amount of effort required to take advantage of them: “Everyone on the business side just assumes you can automate absolutely everything without any oversight, whereas developers, you know how much you have to clean up the code that is generated by AI, especially if you want it to not break everything. There's a lot of oversight that's needed, but I would say it's a huge, huge accelerator when used properly.”

Analysis

The balancing of foundational knowledge and real-world practice as described by stakeholders presents itself less as a proportional tradeoff - fewer foundational concepts to make room for more applied knowledge, or vice versa - and more of an issue of the proper sequencing of content, and the translation of that content into practice. Faculty are understandably focused more on the sequencing aspect. Real-world practice is framed as applicable mostly at a late stage in the student’s education, rather than something running as a parallel track since day one. F-SB and F-BH both described real-world practice as something that is best introduced in 300-400 level or capstone courses, where strong fundamentals can finally be applied to more difficult real-world situations. Others frame it as something that can be introduced as single focused topics throughout curriculum (F-TB) or as something used as illustrative examples while teaching foundational content (F-GC). If a throughline exists, it’s to establish strong and lasting conceptual mastery early, then progressively apply real-world constraints (collaboration, existing external systems, delivery and production workflows) in ways that produce demonstrable evidence of readiness. Technological disruption, especially GenAI, intensifies this by changing the standard of what can be produced quickly, which in turn raises the bar on what counts as trustworthy demonstration of understanding.

While non-faculty stakeholders tended to agree that there should be more real-world practice in undergraduate education, they pointed to different targets in identifying what form that would take. Recent graduates pointed primarily to workflow standards and dealing with the constraints of large established codebases and procedures. Hiring managers described the metric of “real-world” skills primarily as the ability to solve problems under various external constraints: without constant supervision and encouragement (HM-SI; HM-CJ), with scale and security in mind (HM-TD), under pressure (HM-CJ), and without a clear roadmap to a solution (HM-SI; HM-TD; HM-CJ). Most strikingly, when discussing what new hires were often missing as real-world development practice, not a single specific tool or framework was mentioned by a hiring manager interviewed. There appears to be a broad understanding that specific tools and workflows needed are job-specific, but that the ability to adapt to them, and to the environment that they are used in, is the primary indicator of job readiness.

Regarding the effect of GenAI in the undergraduate curriculum, the throughline that emerges is not whether to restrict its usage, or even how much it should be incorporated into undergraduate education, but rather a shift in what counts as credible evidence of learning. Faculty members TDB, BH, and SB all referred to reducing the weight of take-home exercises in grading rubrics, but differed on what the replacement was. TDB referred to an increase in in-class work, SB focused on oral examination and code review, while BH discussed strategies for making the thought process more visible during coding, such as code-history tracing and runtime experimentation. On the employer side, TD treated AI as comparable to other information tools but restated the requirement that action requires understanding: "do not execute until you understand what you're doing." SI's critique of unedited, generic AI output and weak prompting and editing practices points to the same issue: AI makes output easier to generate, so value shifts toward judgment, reasoning, and accountability. In this context, the titular "balancing" includes balancing assessment integrity with realistic tool use, not pretending tools do not exist, but designing tasks that force explanation, validation, responsibility, and a genuine demonstration of understanding.

Discussion

This study examined how the balance between foundational knowledge and real-world development practice is developed, evaluated, and experienced across the education-to-employment pipeline. While purposeful balance of competing areas of study is most directly relevant to course and curricular design, the findings demonstrate that its effects extend beyond the classroom and are interpreted differently depending on stakeholder position.

The primary responsibility of establishing this balance and delivering content to support it falls on instructors. However, the level of influence that instructors have on the overall balance varies significantly. Some instructors described making only course-level decisions about when and how to introduce applied practices, choosing when real-world content would be presented. Others described establishing this balance through course sequencing, for example choosing to focus only on fundamentals during early courses and include what they define as real-world practice only later in capstone or upper-level courses. Even at the course level, these decisions are shaped by existing course structures, time limitations, and predefined learning objectives, limiting the extent to which any individual faculty member can intentionally balance content across a full program. Only those serving in both faculty and supervisory roles, such as department chair, had the authority to oversee and implement allocation of time and content at the curricular level.

From the perspective of recent graduates, the tradeoff between fundamentals and real-world practice is perceived not as a deliberate design or a percentage allocation of content in their undergraduate education, but as a set of practices to which they must adapt. Their accounts emphasize the need to interpret existing systems, work within large established codebases, and be productive in collaborative environments, areas that they often found lacking in coursework. The transition into the workforce is perceived as a shift into an entirely different set of expectations rather than a checklist of which content was or was not covered.

From the perspective of hiring managers, the definitions of real-world practice varied significantly. Interpretations of real-world practice included collaborative coding, interpersonal dynamics, professionalism, comfort with AI usage, thinking at scale, and factoring in security concerns to daily workflows. Notably, not a single subject in this group specifically referred to a tool, framework, or specific technology as what was lacking. From the perspective of foundational skills, their definition was more consistent. They described foundational knowledge more as whether graduates can apply knowledge in context, particularly in areas such as troubleshooting, independent reasoning, and navigating ambiguity. From this vantage point, gaps are most visible not in what academia would consider technical foundations, but in the ability to operationalize that understanding within real-world constraints.

Collectively, these perspectives suggest that the issue is not simply how much emphasis is placed on foundational knowledge versus applied practice, but how effectively the relationship between the two is carried through the pipeline. The academic model often presents these elements piecemeal through sequencing and selective course content, while workplace stakeholders expect them to be integrated earlier and more explicitly.

The emergence of generative AI tools has further clarified this dynamic, rather than introducing a fundamentally new problem. AI appears to reduce the importance of surface-level implementation tasks and place greater emphasis on interpretation, validation, and decision-making; tasks that hiring managers increasingly consider baseline competency. This shift makes existing gaps more visible, particularly where students are able to produce functional outputs but struggle to explain or adapt them.

Conclusions

The findings of this study indicate that the balance between foundational knowledge and real-world practice is not centrally designed or consistently implemented but instead emerges from fragmented decisions and constraints at different points in the pipeline. The primary disconnect identified in this study is not a lack of foundational knowledge, but a gap in how that knowledge is applied, integrated, and evaluated across contexts. Efforts to improve alignment may therefore be better directed toward making this relationship more explicit within the curriculum, rather than attempting to increase coverage of specific technologies or practices.

Limitations

While this study examines how the balance between foundational knowledge and real-world practice is constructed, experienced, and evaluated, it does not capture the full set of structural forces that shape that balance. Particularly, faculty participants' level of control in enacting decisions that shape this balance varied significantly. These choices were primarily described as course-level decisions rather than program-level design, limiting visibility into how balance is coordinated across entire degree programs. Interviews with a larger set of faculty that also includes decision makers who shape curriculum, such as department chairs or divisional deans, would give greater insight into how these choices are applied at the program level. Additionally, the study draws on perspectives from stakeholders who engage with and define this concept in fundamentally different ways - faculty as creators and definers, graduates as recipients, and hiring managers as evaluators, making cross-group comparison interpretive rather than directly equivalent. As a qualitative study based on a small, purposive sample, the findings are not intended to be generalizable, but instead to provide a grounded account of how this issue is experienced in practice across the employment pipeline. Although a faculty-centered study would provide greater insight into the creation and logic of curricular balance, it would not capture the broader view of how this balance is interpreted across stakeholder groups.

Recommendations for Future Research

Future research should examine how the balance between foundational knowledge and real-world practice is shaped at the program and institutional level, particularly in contexts where faculty have limited influence over curriculum structure and sequencing. Studies that follow students longitudinally from coursework into early career roles would provide deeper insight into how this balance is experienced and how effectively knowledge transfers across domains. Targeted investigations of course-level adaptations - such as early integration of collaborative workflows, work in existing codebases, or applied problem contexts - could help identify approaches that improve the connection between foundational understanding and real-world application. Finally, as generative AI tools continue to evolve, further research is needed to understand how their use reshapes expectations for entry-level competence, particularly in relation to reasoning, validation, and independent problem-solving.

References

ABET. (2024). *Criteria for accrediting computing programs, 2025–2026*. ABET. https://www.abet.org/wp-content/uploads/2024/11/2025-2026_CAC_Criteria.pdf

Alabdulkarim, Y., Almukaynizi, M., Almaymoni, M., Alhadlag, A., Alsalamah, S., & Alsalamah, H. (2021). Saudi undergraduate computing programs in light of the 2020 Curricula Report. *International Journal of Education and Information Technologies*, *15*, 137–146. https://doi.org/10.46300/9109.2021.15.14

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Dexter, S. (2023). Developing faculty EdTech instructional decision-making competence with principles for the integration of EdTech. *Educational Technology Research and Development*, *71*(1), 163–179. https://doi.org/10.1007/s11423-023-10198-0

Exter, M., Caskurlu, S., & Fernandez, T. (2018). Comparing computing professionals' perceptions of importance of skills and knowledge on the job and coverage in undergraduate experiences. *ACM Transactions on Computing Education*, *18*(4), 1–29. https://doi.org/10.1145/3218430

Karataş, F., Eriçok, B., & Tanrikulu, L. (2025). Reshaping curriculum adaptation in the age of artificial intelligence. *British Educational Research Journal*, *51*(1), 154–180. https://doi.org/10.1002/berj.4068

Lau, S., & Guo, P.J. (2023). From "ban it till we understand it" to "resistance is futile": How university programming instructors plan to adapt as more students use AI code generation and explanation tools such as ChatGPT and GitHub Copilot. In *Proceedings of the 2023 ACM Conference on International Computing Education Research Vol.1*, 106–121. https://doi.org/10.1145/3568813.3600138

Li, P. L., Ko, A. J., & Begel, A. (2020). What distinguishes great software engineers? *Empirical Software Engineering*, *25*(1), 322–352. https://doi.org/10.1007/s10664-019-09773-y

Lundberg, G. M., Krogstie, B. R., & Krogstie, J. (2020). Becoming fully operational: Employability and the need for training of computer science graduates. *2020 IEEE Global Engineering Education Conference (EDUCON)*, 644–651. https://doi.org/10.1109/EDUCON45650.2020.9125188

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., & Savelka, J. (2023). The robots are here: Navigating the generative AI revolution in computing education. In *Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education*, 108–159. https://doi.org/10.1145/3623762.3633499

Radermacher, A., & Walia, G. (2013). Gaps between industry expectations and the abilities of graduates. In *Proceedings of the 44th ACM Technical Symposium on Computer Science Education*, 525–530. https://doi.org/10.1145/2445196.2445351

Rogers, E. M. (2003). *Diffusion of innovations* (5th ed.). Free Press.

Stamm, S. (2023). Desired qualifications sought in entry level software engineers. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, 854–860. https://doi.org/10.1145/3545945.3569849

Valstar, S., Krause-Levy, S., Macedo, A., Griswold, W. G., & Porter, L. (2020). Faculty views on the goals of an undergraduate CS education and the academia-industry gap. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 577–583. https://doi.org/10.1145/3328778.3366834

Whalley, J., Imbulpitiya, A., & Clear, T. (2025). Computing capstone courses as preparation for practice: A global survey of instructors. *ACM Inroads*, *16*(1), 26–39. https://doi.org/10.1145/3715880

Zastudil, C., Rogalska, M., Kapp, C., Vaughn, J., & MacNeil, S. (2023). *Generative AI in computing education: Perspectives of students and instructors*. arXiv. https://doi.org/10.48550/arXiv.2308.04309

Appendix

Interview Questions

Interview questions were organized around eight key topics. Each topic had minor revisions in wording to best apply to the target stakeholder group: faculty (F), hiring managers (HM), and recent graduates (RG).

1. Background
F: Can you briefly describe your teaching role, the types of computing courses you teach, and the student population you work with?
HM: Can you briefly describe your role, and your experience working with or hiring entry-level computing professionals?
RG: Can you briefly describe your academic background in computing and your current role in the field?

2. Defining "Foundational Knowledge"
F: When you think about the phrase "foundational knowledge and skills" in computing education, what does that mean to you, and which elements do you consider most essential for undergraduates?
HM: When you think about foundational knowledge and skills in computing, what does that mean to you when evaluating entry-level candidates?
RG: When you think about "foundational knowledge" and skills in computing, what does that mean to you based on your current role?

3. Defining "Real-World Practice"
F: When you think about the phrase "real-world development practice" in computing, what does that mean to you in the context of teaching or curriculum design?
HM: When you think about "real-world practice" in computing, what does that mean to you in your professional environment?
RG: When you think about "real-world development practice" in computing, what does that mean to you based on your experience entering the workforce?

4. Balance Between Foundations and Practice
F: How do you approach balancing foundational knowledge and real-world development practices in your courses?
HM: From your perspective, how well do recent graduates balance foundational knowledge with real-world practices?
RG: How did your coursework balance foundational knowledge and real-world development practices, and how effective did that balance feel once you entered the workforce?

5-A. Knowledge Gaps (RG / HM only)
HM: When you work with or hire entry-level candidates, what tools, workflows, or development practices are most commonly missing or underdeveloped?
RG: When you started your first role, what specific tools, workflows, or development practices did you realize you weren't adequately prepared for?

5-B. Curriculum Inclusion Logic (F only)
F: Have you had any feedback from students regarding their desire to learn about certain real-world or applied practices?
F: When foundational concepts are the primary focus of a course or program, how do you decide if and where real-world development practices are incorporated?

6. Technological Change & Emerging Tools

F: How have recent technological changes, including AI-assisted tools, influenced how you think about teaching, assessment, or curriculum design?
HM: How have recent technologies, including AI-assisted tools, influenced expectations for entry-level developers in your organization?
RG: How have recent technologies, including AI-assisted tools, influenced how you approach your professional work?

7. Academic Rigor & Learning Integrity
F: Given the availability of tools that can automate coding and problem-solving, how do you design assessments that ensure students are genuinely developing and demonstrating understanding?
HM: Given the availability of tools that can assist with coding or problem-solving, how do you evaluate whether an entry-level candidate is genuinely demonstrating understanding rather than surface-level or tool-assisted familiarity?
RG: Given the availability of tools that could assist or automate coursework, what helped you determine that you were genuinely developing and demonstrating understanding during your studies?

8. Looking Forward
F: Looking ahead, how do you see undergraduate computing education evolving in terms of integrating foundational knowledge with real-world development practices?
HM: Looking ahead, what changes would you like to see in undergraduate computing education to better combine foundational knowledge with real-world development practices?
RG: Looking ahead, based on your experience transitioning into the workforce, what changes would you recommend to better integrate foundational knowledge with real-world development practices for future students?