# 21st-Century Toolkit For Information Technology Graduates: Preparing Students with Technical Acumen, AI Awareness and Interpersonal Competence

T.J.Kane C.T Hampton S. Mithun

Purdue University Purdue University Purdue University

**Abstract**

This study explores the evolving demands of the computing workplace in the age of artificial intelligence (AI), focusing on the need to be AI-aware, and also on the critical integration of technical and interpersonal competencies. The goal is to discover how to build our under-graduate Applied Computing courses in support of the best transition from college to career, for our students, and their success in the software development industry. Using survey data collected from 10 employers in the greater Indianapolis area, the research highlights three key insights:

1. There is widespread, but not 100%, adoption of AI technologies across software development companies
2. There is a strong concurrent need for engineering graduates to possess robust interpersonal and business skills
3. There is a need for our graduates to use modern tooling, including AI, effectively, to improve productivity

These findings underscore the importance of preparing students not only with AI proficiency but also with the soft skills required to navigate complex, collaborative professional environments. The paper proposes educational strategies to bridge this gap, ensuring graduates are equipped to meet contemporary workforce expectations.

**Introduction**

The rapid proliferation of artificial intelligence (AI) technologies is transforming industries and reshaping the skills demanded in the engineering workforce. While technical proficiency in AI is increasingly essential, employers also emphasize the importance of interpersonal and business acumen to thrive in collaborative, interdisciplinary environments, and the ability to discern effective and ineffective AI generated code and other content. This study investigates employers needs and expectations, and gives insight into how engineering education can better prepare students for this dual demand, addressing the need to balance technical expertise with the development of critical analytical and soft skills. Drawing on survey data collected from employers, the research identifies key trends in AI adoption and workforce expectations, shedding light on the gaps between academic preparation and industry needs. By exploring these insights, this paper aims to propose actionable strategies to equip information technology graduates with the comprehensive skill set necessary for success in the modern workplace, specifically in the software development domain. Business skills, such as understanding business processes, interpreting client requirements, and communicating strategic technical issues, are increasingly important for computing graduates. Bridging the gap between college and career is one aspect, however there is another gap between technical prowess and successful business endeavors.

**Literature review**

**Importance of AI Awareness:**

AI awareness is crucial for computing graduates to stay relevant in the rapidly evolving tech landscape. It is so important that they understand the ethical implications, potential biases, and societal impacts of AI technologies.

A report from the Harvard Graduate School of Education highlights that students are already using AI in various ways, emphasizing the need for awareness and responsible use [1]

**Current Trends and Challenges:**

The integration of AI in education and the workforce is accelerating, with companies like Microsoft and Pearson collaborating to bridge the AI skills gap. [2]

Awareness about AI is not just about technical skills but also about understanding its broader implications on society and the workforce [3]

**Interpersonal Skills, Significance in the Tech Industry:**

Interpersonal skills, such as communication, teamwork, and emotional intelligence, are essential for success in tech careers. These skills help professionals collaborate effectively, resolve conflicts, and lead teams.

Codecademy emphasizes that interpersonal skills are as important as technical skills for career advancement in tech. [4]

**Key Interpersonal Skills:**

Communication: Clear and efficient communication is vital for conveying technical information to non-technical stakeholders. [5]

Teamwork: Collaboration among team members is crucial for the successful completion of projects [5]

Emotional Intelligence: Understanding and managing one's emotions and those of others can significantly enhance workplace relationships and productivity [5]

**Business Skills, Relevance for Computing Graduates:**

ESSEC Business School's initiatives highlight the need for business students to develop tech-ready skills to succeed in the digital economy [6]

**Essential Business Skills:**

Tech Fluency: Understanding foundational tech concepts and their applications in business (6)

Analytical Rigor: Applying critical thinking and data analysis to solve business problems[6]

Creative Problem-Solving: Developing innovative solutions to complex challenges[6]

**Methods**

This study utilized a mixed-methods approach combining quantitative survey data and qualitative insights drawn from the researcher’s extensive teaching experience, with a specific emphasis on software development.

**Survey Design and Administration**: A targeted survey was developed by the PI, to assess employer perspectives on skills desired in engineering graduates, focusing on the integration of artificial intelligence (AI) and interpersonal competencies within the domain of software development. The survey questions are based on the PI’s industry experience and association with local industry professionals, and familiarity of all investigators in teaching computing and technical education courses. The survey included closed-ended questions (e.g., Likert scale ratings) to quantify employer priorities, as well as open-ended questions to capture detailed feedback. Participants are industry professionals engaged in software development, ranging from startups to large enterprises. Responses were collected and analyzed to identify trends and priorities within the software development field.

**Qualitative Insights from Teaching**: To complement the survey findings, the PI incorporated qualitative insights drawn from 13 years of software engineering experience, and 10 years teaching information technology students with a focus on software development. These insights included classroom observations, student feedback, and first-hand experiences implementing curriculum changes aimed at enhancing AI-related technical skills and soft skill development. Challenges and successes in preparing students for software development careers were analyzed to highlight areas of alignment and misalignment with employer expectations.

**Analysis**: Quantitative survey responses were analyzed to identify dominant trends specific to software development. Open-ended responses were coded thematically, emphasizing insights related to AI usage, interpersonal skills, and business acumen within the software industry. Findings from the survey were cross-referenced with classroom experiences to synthesize a comprehensive understanding of employer expectations and recommendations for educational practices in software development.

**Results**

The survey results provide valuable insights into employer expectations for engineering graduates, specifically within the domain of software development. The responses, feedback and participant diversity lend credibility to the data, allowing for meaningful analysis of trends in AI adoption, interpersonal skills, and business acumen within the software development field. These findings form the foundation for addressing gaps in educational preparation and proposing strategies to align academic outcomes with industry demands.

**Quantitative Data Presentation**

A blue rectangular sign with white text

AI-generated content may be incorrect.Do employers use AI? 70% of respondents stated that they use AI to assist in the software development process, with 30% of companies included in the survey, using a proprietary AI engine. Of note, 30% of respondents indicated that they did not use AI.

Respondents were asked what **assists** their software developers used, and given choices to select from, with an optional ‘other’ to expand on this**.** The choices given are based on the PI’s own experience as a software developer.

Here, it can be seen that AI engine output is used as much as peer code reviews, indicating the importance of clear and constructive communication for such a code review, and the ability to discern issues in the code, both AI generated and developer created.

The additional ‘other’ prompt indicated the Agile methodology. It is interesting that Agile was mentioned in reference to assisting the software development process. Agile is a widely used project management methodology, that stresses the importance of open, effective communication via regular meetings, collaboration between team members, and support of any difficulties that need to be resolved. This highlights the importance of culturing those soft skills that teams need, to best deliver a robust software product, efficiently and effectively. The reference to the Agile methodology also points to alignment of assigned tasks within a group of developers, with the expectation that these tasks be completed within a certain time frame (Sprint), and to the standards of the company.

A screenshot of a computer

AI-generated content may be incorrect.This focused question gathered data about code development strategies in the current workforce, the tooling commonly used and the content that could and perhaps should, be included in a university course that prepares students for a career in industry. AI and code reviews come out on top here. Software development templates, and internet searches help software developers create code, as has been the case for many years. Respondents indicated that the Agile methodology is a valuable assist used by software developers.

Responses demonstrate the need for an integrated approach – it is not all about AI, companies use their own protocols and standards, and have a defined development team hierarchy. Structures are in place to assess the quality of code developed by the team, and team members should work with the company protocol, using company developed templates and strategies as they build or maintain a software product.

Respondents were asked if their developers trust AI generated content, where ‘trust’ is described as using the AI content ‘without concern’. On a scale of 0(least trustworthy) to 5(absolute trust), the average response was 2.7. There is potential positive bias from the companies with their own internally developed AI engine, which if anything, would increase the average here. However, it is clear that there is a need for developers who can discern what is ineffective or inaccurate, in AI generated code.

When asked if employers believe that AI has a positive impact on the software development process, the response correlated with a belief that AI is useful – average rating of 4 on a scale of 0 (not useful) to 5(useful). The corollary, has AI made a *negative* impact on the development process’, supported the belief that AI is a useful tool, with an average rating of 1.43. Later questions in the survey relating to the specific fears about AI, may explain these leanings in the survey group.

Productivity is an important metric in software development. Respondents were asked if software developers are more productive as a result of using AI. That result was overwhelmingly positive, 4.2, on a scale from 0(least) to 5(most productive as a result of AI), one survey participant citing ***reduced cognitive load and increased code production***. Another envisioning the power of AI enhancing the role of the developer.

***Using AI as an "engine" to generate code is one side of the equation, versus using AI that generates code for an app or program that technically becomes a product within the ecosystem. Productivity of software developer will increase however from a job requirement standpoint, their role will also include product management.***

An optional survey question asked respondents to share their reflection on the impact of AI on the software development process. The responses given support the extensive use of AI across respondent companies, the positive view of AI as a tool in particular for starting a project, the necessity to discern content generated by AI, and the critical need to identify mistakes made by AI.

***still too many errors to trust AI***

***It has reduced cognitive load when writing code. We still find that AI makes mistakes and human review is required (not blindly trusting the result) but we've found it is generally easier to review a mostly correct piece of code than to handcraft the same code Finding the AI tools are particularly useful as people jump between project domains and coding languages***

***the majority of software developed is for maintenance or small features. LLM and the ability to ask the LLM to create a code base for a specific spec, becomes a prompt engineering challenge. So we expect developers jump start their code using Co-pilot and such and then continue with the current devops methodology.***

***AI has transformed the software development process, streamlining workflows, enhancing productivity, and pushing the boundaries of what's possible. The largest impact is in code generation and assistance. Debugging and testing have also evolved as the toolsets have evolved - examples include debugging, testing, vulnerability prediction, and recommended fixes.***

***It is used as an enhancement/guide but all code is human centered, tech enables. Everything is reviewed and tested by us.***

Not all respondents answered a question relating to the potential for reduced need for software developers in their company, as a result of AI. However, of those who did respond, there was a split – 67% indicated a belief in a decreased need for software developers, whereas 33% indicated no impact would be seen.

Given the undisputed presence of AI, respondents were asked what their vision is for the role of software developers as AI progresses. Feedback points to the need for appropriate and effective development of AI prompts, in order to get the most effective answer from the AI engine, specifically how ‘adjusting AI prompts’ and ‘Agent orchestration’.

These responses support the use of AI as a tool,

***We will still need good developers. They will need to understand the overall full scale of the software product which is currently a short fall of the AI models. The AI model is expected to help them become more productive.***

***We are a human centered tech enabled firm so AI is used as an assistant, but we will not be fully dependent on AI.***

Other responses expand their view of AI as a tool and envision software developers in the role of designers or managers, somewhat removed from the coding process, as a result of AI.

***Software developers will work more in a distributed toolset environment that will require prompting for code outcomes versus developing lines of code. Relying less on coding knowledge vs. understanding the usage and outcomes.***

***Using AI as an "engine" to generate code is one side of the equation…. Productivity of software developer will increase however from a job requirement standpoint, their role will also include product management.***

Client requirements inform the UI/UX design process. Interestingly, respondents had a favorable opinion of AI when used in product design. Participants were asked if AI could be used to support the UX/UI design process. On a scale of 0 to 5(most supportive), the average recorded response was 4.0. There is a leap from product design to developing software that supports the front-end and back-end of the product. When asked if AI could be used to translate client requirements into tasks in the software development process, respondents were split, with 67% stating that this was possible and 33% disagreeing entirely.

The survey concerns the next generation of IT professionals. Universities continue to support technical degree programs that supply graduates for the IT industry. Given AI, there is another skillset that best prepares graduates to be successful professionals. Respondents were asked to consider the role of the software developer in a modern organization, using AI and other tooling.

Most respondents indicated that software developers would play an active role in the design process, signifying a need for interpersonal and analytical skills, as well as the ability to be creative in terms of software design and implementation of the client requirements, ie if working with a UI design team, software developers would work to best *implement* product design, striking a balance between technical feasibility (not all frameworks support the exact user interface features requested by the designers) and delivery of an effective solution for the client. Other software developer responsibilities might include robust testing of the software product.

Very often, software development companies maintain an existing codebase, legacy applications that are still in use, having been created many years ago. This is an important consideration, as legacy code bases are often less open to the use of modern AI engines, given that the language and the development methodology may have preceded LLMs in use today. 83% of respondents had responsibility for a legacy code base, and 100% of these stated that AI was ineffective in helping maintain and modify this important part of the software development industry. This finding underlines the necessity to prepare students to develop code independently of AI, and to analyze the impact of code. To best prepare to work on existing applications, students should master the traditional coding constructs, synthesis of program code and logic, development of effective loops and conditionals, and more, that will enable them to dive in and take care of these legacy applications.

There were concerns raised regarding the use of AI engines to generate code. These concerns were rooted in security, code bloat, over-reliance on the engine on the part of the developer, and injection of malicious content. Based on the PI’s industry experience, specific questions were posed to the respondents, and feedback indicates a high level of concern in using AI.

The results indicate a lack of trust in the use of AI engines to generate code, and the importance of preparing today’s engineers to verify and problem solve, using AI generated content as an input.

The PI drew on prior industry experience to ask respondents to select the top 3 soft-skills they want to see in new hires, our college graduates. The responses indicate a need for a well-rounded graduate, with good inter-personal aptitude, an ability to function as an innovative thinker and communicator, while also working well in teams and under the direction of leadership. Other skills listed included a demonstration of passion for software engineering and technology,

***During the interview process we are trying to identify passion. Passion for software engineering and technology, helps the developers to be better at what they do.***

Collaboration, innovative thinking, as well as importance of following directions, are all part of a soft-skills subset that cannot be ignored as we prepare our engineers of the future.

Respondents were asked to indicate the 3 most important technical skills, not based on any specific framework, although targeted at current tooling and practices in IT and specifically related to AI. The technical skills employers look for reveal the importance of being able to effectively *use* AI as a tool, and not rely solely on an AI engine to develop code.

Respondents were given an opportunity to list additional technical skills they would like to see in new hires. One respondent indicated that testing is an important area, and much testing is done through the use of AI.

The transition from college to career is an important one. A successful first couple of weeks in a new job can set the stage for a more comfortable and successful outcome. This in mind, respondents were asked what a typical set of onboarding activities might be. CBTs (Computer Based Training), peer mentoring and exposure to customer requirements and feedback all featured in the written feedback.

***CBTs, interviews with existing staff, introduction to common issues, and exposure to customer communications.***

***we use peer mentoring to on board new developers. they are expected to take and run independently with small task in the first three months. We also offer many recorded videos about the product and product architechture***

***Three days of virtual onboarding initially, then onto working with an HR team and managers to get new hires onto projects, design development goals, and through the first year of employment.***

**Qualitative Data Insights**

While the goal posts have shifted in terms of technical preparation of computing graduates, employers specifically look for a combination of these technical skills, with the soft skills that are needed to coordinate successful teams, and innovate and respond to customer and industry needs. Effective use of AI is definitely seen as a desired skillset. Effective use of AI as a tool, not only involves prompt creation and agent orchestration, but also hinges on awareness of AI’s inherent security risks and potential to deliver a sub-optimal solution. A surprising finding from the survey is that not all employers use AI to assist in the software development process, but instead use traditional code starter templates, existing code, peer support. While only 30% of the respondents stated that they did not use AI, all respondents indicated that students should be prepared to use other assists, like training videos, tech support, documentation, peer reviews.

AI is here and is a component of the software developer toolkit. The findings from this survey indicate that employers view AI as a valuable tool, given the responses to the impact of AI on productivity. However, results also reveal that AI generated code is not trusted, the output of an AI engine should be tempered by human insight and experience, and may need to be manipulated in order to represent an effective solution. Students should know how to refactor AI generated code, and to remove unnecessary parts. Students should recognize code bloat and be prepared to analyze generated code for unsafe content.

It is clear that the survey group expect their software developers to use AI, in the context of finding answers to development questions, or to scaffold a project. However, the need to discern effective code is critical. We can deduce this from the responses to questions about the impact of AI, positive and negative, and the feedback regarding employer concerns about AI. Given a mission critical application, security concerns may dictate an avoidance of AI engines, in favor of traditional approaches, like starter code templates and code repositories, with less reliance on AI generated responses. Students should know how to develop code independently of AI, graduates should have the confidence and the acumen to develop solutions independently. This means that a university course should stress the importance of using these traditional approaches, and should include ample practice both on and off the AI court.

An open response question about software developer roles and expectations indicate that employers are looking for innovative thinkers, who can use AI to best effect. This goes beyond developing prompts to get the most effective response from an AI engine – one employer surveyed specifically mentioned ‘Agent Orchestration’. AI agent orchestration employs a network of AI agents, each designed for specific tasks, working together to automate complex workflows and processes. Agentic AI is becoming more important. Rather than focus on the specifics here, our students should be prepared to continuously learn about new technology, and implement solutions accordingly. Perhaps this is an area in which direct industry involvement in the college curriculum is of great benefit to the student, and could include possible industry rotations, coordinated by the college and industry partners.

According to the respondents of the survey, software professionals must be comfortable and effective as part of a team, be open and able to take directions from a team lead or supervisory role, be receptive to feedback from code reviews. Companies expect employees to observe company and/or group protocol. The Agile methodology was specifically referenced by the respondents. This means that students should be exposed to working within structured groups, with respect for a hierarchy, and communicating effectively. They should also understand the importance of a time line, in delivering effective software that forms part of a larger project.

The comment that made the biggest impact on the PI concerned looking for passion for the discipline in prospective software development hires. The question about developing passion is one that applies to many professions. How can this be nurtured in a college software development pathway? Perhaps the answer lies in getting to know the students and their interests, and showing them how to use computing to express themselves. This ties into Self Determination Theory, SDT. SDT was popularized in the 1970s, and is rooted in the theory of bringing forth intrinsic motivation. SDT emphasizes the importance of intrinsic motivation and the fulfillment of three basic psychological needs: autonomy, competence, and relatedness. It suggests that when these needs are met, individuals are more likely to experience growth, well-being, and sustained motivation, which would be very beneficial for a course framework.

**Comparative Analysis**

* Discuss connections between your survey findings and broader industry trends or academic literature.
* For example, if employers highlight AI and communication skills, relate this to known workforce demands in software development.

**Key Takeaways**

The findings illustrate that employers want to hire software developers who are technically proficient, as would be expected, but that is not all. Employers place emphasis on the ability of their developers to respect company protocols, work well with teams and deliver software products that are compliant with customer needs. They want these new hires to be open to the use of modern tooling, of which AI has emerged as an important component.

Employers also look for analytical skills, skills that translate customer needs to effective software offerings, and may involve the use of AI as an assist. The ability to verify AI output is also critical, as is the awareness of the security implications of using an AI engine. Most software organizations in the survey use AI, but not all. Some employers do not use an AI engine due to concerns about security, code bloat, malicious code injection. Most employers in the survey had responsibility for legacy code bases. As per the respondents, AI is not helpful in maintaining legacy code base content, due to the fact that available LMS are not ‘trained’ in legacy code, development platforms and methodologies. Graduates joining such companies should be prepared to use traditional software development techniques , like skeleton code, peer reviews and team coding protocols. Finally, soft skills are as important now as they were in the past, as software professionals must navigate interpersonal and business challenges as they work in software development teams to create and maintain software.

Graduates prepared to meet the demands of the workforce on day 1, will live up to the expectations of the employers in the survey group, and beyond.

**Discussion**

The key takeaways are worthy of discussion and can be used to inform college courses, to best benefit students and to prepare them for the expectations of industry.

Employers look for relevant technical proficiency, in an ever-changing tech landscape. The industry constantly evolves to incorporate new IDES, new libraries and new languages. However the basic principles of software design do not change. All program code, whether procedural, functional, Object Oriented, scripting or logic, incorporates Boolean algebra, uses common programming constructs – loops and conditionals , and manipulates data. These basics should be stressed, so that students are best prepared to adapt to the industry, with a mastery of the basics in software development. It is important for A computer screen shot of a computer screen

AI-generated content may be incorrect.students to see how applicable these basic skills are. The PI suggests incorporating visual demonstration of logic in basic programming concepts classes for Freshmen and Sophomores. An example here would be the use of Python, and the Turtle module to stress how to use the language in a visual way that students may find entertaining. One such project demonstrating the use of loops, conditionals and functions, effectively creates a city scape, using a single function, loops and conditionals.

Employers surveyed clearly indicated the importance of customer requirements. It is vital that team leads, and team members alike synthesize customer needs, to ensure that an effective software product results. Here, the ability to discern what should needs to be built into robust products, should come from a holistic understanding of what the customer wants. That includes interviewing the customer, reading through requirements documents, building prototypes, taking feedback and translating need to integration of initial requirements and modifications throughout the development life cycle. Here, the PI believes that it is important to give students an opportunity to build software products based on a general specification. Students at any level, can be asked to build a product that accomplishes a task. Freshman students can be asked to build a simple interface, more advanced students can build a full-stack software offering that achieves an objective – such as a meeting coordination app tied to a database. A screenshot of a computer

AI-generated content may be incorrect.

In this, AI can be used to scaffold the application, but the students should take the task to completion and ensure that requirements are met. There are multiple opportunities in college for students to build solutions, for instance, a syllabus generator tool that uses syllabus updates from a college specific repository. A student food pantry inventory and clothing donation application, an escort home safely application.

Course content should include such opportunities for students. Students should be given the chance to think through a development effort, instead of being told what exactly to do to achieve the objective. This ties into Self Determination Theory.

Employers work with legacy code bases, that are not amenable to interpretation via AI. The PI believes that the best way to expose students to this area of the industry is via industry rotations, and any opportunities to work on college software tooling, requiring involvement of the college tech support department.

Employers want developers to work with AI but not depend on it. They want developers to be able to discern code bloat and refactor AI generated code to conform to lean expectations and requirements. The PI believes that students should be given multiple opportunities to create AI prompts as the basis for code generation, and should then be asked to refactor the code to use only course based content. AI code generation will often result in content that includes exception handling where this is not required, or a specific protocol that is not compatible with a specification. Refactoring AI generated content is a powerful way to frame AI as a tool and assess student acumen at the same time.

Soft skills can be nurtured in the classroom. Collaboration can be encouraged and emphasized by rewarding positive sharing and collaboration with a positive grade, and/or positive affirmations. Opportunities to present to faculty, and honest, constructive feedback, can demonstrate the impact of effective work. However, the downside here is that students who do not contribute professionally should be shown that work and effort, as well as accuracy and professionalism are requirements of staying in a job in the industry. This is often a difficult area – and would require a set of solid expectations, both from the course and the department, as well as an appreciation of why a student may not have demonstrated professional effort. Too often, as in any job and any industry, professionals who do not deliver professional results, and who are not seen as effective contributors, will lose their jobs. We want to emphasize the importance of putting forth effort and maintaining professional composure and practice, starting from Freshman year. In this, demonstrations and expectations are key.

Employers state that the AGILE methodology is used extensively. Student classwork and group work can use the AGILE methodology to foster communications skills and respect for time management and delivery dates.

Students benefit from witnessing a well-functioning department, staffed by faculty who are respectful of each other, and proud of the department’s accomplishments. Faculty collaboration can model professionalism and passion for computing. In co-developing content, in forging links between the different courses in a pathway, and in obtaining grants and industry involvement, faculty demonstrate the power of professional sharing and collaboration.

This study reveals that Employers typically have a line-up of on-boarding activities and training exercises for new hires. From personal experience, the PI believes that successful handling of these onboarding activities can propel the new hire to success. Very often employers look for experience, which only a handful of graduates have. If colleges offered internship programs, rotations in industry, and the opportunity to complete typical industry on-boarding tasks, like a data management standard CRUD (Create, Read, Update, Delete) project, students would feel more comfortable in transitioning to industry. This requires industry partner input into college courses, content and delivery.

* Compare your findings with existing literature.
* Address limitations and potential areas for future research.

**Conclusion**

Employers need graduates to be equipped with 21st century skills – technical skills that incorporate the ability to develop code and use AI to best support the development of robust code. AI can be used as a tool across the software development platform, with the exception of maintaining legacy applications, in which case AI is deemed ineffective in support of making changes to that code base. Employers want confident hires, who can use AI but are not dependent on it. On the contrary, software developers should use AI, and then make the changes necessary for a specific solution – which includes removing code bloat and ensuring the integrity of the developed code.

Technical skills alone are not enough to give our students the best start in the tech industry. They should be able to collaborate, solve work place difficulties and innovate.

Employers want graduates who are passionate about their chosen field, because ‘***Passion for software engineering and technology, helps the developers to be better at what they do’.***

The graduate equipped with strong basic skills in software development, who has been trained to use AI as a tool, and who knows how to communicate clearly and effectively, is in a great position to succeed. We, as faculty, can help create this success by modeling our courses on industry needs, listening to industry needs and demonstrating professionalism within our departments, as well as passion for the content.

**References**

[1] Students Are Using AI Already. Here’s What They Think Adults Should Know <https://www.gse.harvard.edu/ideas/usable-knowledge/24/09/students-are-using-ai-already-heres-what-they-think-adults-should-know>

[2] **Transforming the future of learning and work with AI skilling** <https://www.microsoft.com/en-us/education/blog/2025/03/transforming-the-future-of-learning-and-work-with-ai-skilling/>

[3] <https://dailytitan.com/opinion/opinion-ai-awareness-is-key-for-post-graduate-success/article_8d3b4368-10b2-11ef-b0e7-774070df3b2b.html>

[4] **Code Academy, The Interpersonal Skills That You Need In Tech Careers** <https://www.codecademy.com/resources/blog/interpersonal-skills-for-tech-careers-examples/>

[5]

<https://www.fortray.com/blog/interpersonal-skills-in-it/>

[6] https://knowledge.essec.edu/en/innovation/equipping-graduates-tech-success.html

[7] Papier, J. (2017). Improving college-to-work transitions through enhanced training for employment. *Research in Post-Compulsory Education*, *22*(1), 38–48. https://doi.org/10.1080/13596748.2016.1272084