DOI:10.1145/3624729

Rahul Simha, Amruth N. Kumar, and Rajendra K. Raj

Opinion

Undergraduate Computer Science Curricula

First-job readiness versus long-term career preparation.

THERE CAN BE many conflicting goals for the design of a computer science curriculum, including: immediate employability in industry, preparation for long-term success in an ever-changing discipline, and preparation for graduate (that is, post-graduate) study. Emphasis on immediate employability may lead to prioritizing current tools and techniques at the expense of foundational and theoretical skills as well as broader liberal-arts studies that are crucial for long-term career success and graduate work. The implications of these conflicting goals include allocation of finite resources (time, courses in the curriculum), unwillingness of students to invest in the mathematics that they see as irrelevant to their immediate career goals, and reluctance of faculty to have their courses be driven by a continually evolving marketplace of tools and APIs. A balanced curriculum benefits all stakeholders: students, employers, and faculty.

Would a data-driven approach help faculty design curricula that effectively balance these multiple goals? For example, if we ask graduates of computer science programs to reflect on the impact of their undergraduate education, explicitly focusing on short- and long-term impact, will there be enough meaningful data to significantly inform curricular design? A recent survey of industry professionals undertaken by the ACM/IEEE-CS/AAAI 2023 Computer Science Curricular Task Force (CS2023)[a] points the way. This column presents one aspect of that survey—a focus on comparing short-term and long-term views—and calls for similar surveys of industry professionals to be conducted on an ongoing basis to refine our understanding of the role played by various elements of undergraduate computer science curricula in the success of graduates.

IMAGERY FROM SHUTTERSTOCK.COM

The Survey

Approximately every decade, ACM along with its partnering organizations assembles task forces of international experts from academia and industry to assess changing trends in computing and issue curricular guidelines for undergraduate programs in computer science and six allied disciplines. The CS2023 Task Force on Computer Science curricula[2] started by conducting a survey of industry professionals, who were asked to provide their current background and rate various curricular aspects based on their own educational experience. The survey was encoded in Qualtrics, distributed in 2022 to a proprietary list maintained by ACM of its mem-

[a] See https://csed.acm.org/

bers and tech-talk registrants, and was completed by 865 practitioners. We focus here on aspects of the survey related to teasing out the differences between short- and long-term views of industry professionals.

The survey respondents were first asked to select the category representing their number of years of professional experience: 1–2 years, 3–5 years, 6–10 years, 11–20 years, and more than 20 years. Next, to elicit differences between short-term (first-job readiness) and long-term (career preparation) views, respondents were asked to rate the importance of 16 knowledge areas within computer science identified in CS2013[1]: algorithms and complexity, architecture and organization, computational science, discrete structures, graphics and visualization, human-computer interaction, information assurance and security, information management, intelligent systems, networking and communication, operating systems, platform-based development, parallel and distributed computing, programming languages, software development fundamentals, and software engineering.

Next, respondents were asked to rate on a Likert scale (0 = not applicable, 1 = not important, 2 = somewhat important, 3 = very important) how important the following curricular components were for preparing them for their career: capstone course or senior-project; internship or co-op while in college; availability of computer science electives; liberal arts education (including courses on history, languages, literature, philosophy, creative arts, psychology, and others); sciences (physics, chemistry, biology, and others), scientific method, and scientific inquiry; and mathematics (discrete mathematics, calculus, probability and statistics, linear algebra, and other mathematics).

Short- and Long-Term Views

The survey data presents an opportunity to answer two questions—when professionals are explicitly asked to take a short-term view and then a long-term view, are any areas of computer science rated differently? Second, do more experienced professionals (who are presumably able to take the long view) rate items differently than less experienced professionals? The purpose of this column is not to proclaim the importance of certain areas of computer science based on professionals taking one or the other view, but rather to make the case that there is a difference in short-term needs versus longer-term career preparation, and in the future, surveys of professionals should explicitly elicit these differences to better shape computing curricula.

Of the 16 knowledge areas listed, the five greatest differences between short- and long-term importance were reported in architecture, parallel and distributed computing, security, artificial intelligence, and databases. For example, the average rating (on a Likert scale of 0 = Not important, and 3 = Very important) of architecture was 1.93 for the short-term, but 2.50 for the long-term, a difference of nearly 30%. Interestingly, all five areas were rated more important for the long term. In some sense, the first three—architecture, parallel and distributed computing, and security—comprise a broader "systems" viewpoint. We propose that while first-job readiness might demand software development skills, the long-term view might emphasize foun-

dational skills for system design.

Did more experienced professionals (21+ years in the industry) rate knowledge areas differently than less experienced ones (1–2 years or 3–5 years)? The biggest differences were seen in human-computer interaction and security, both rated more important by experienced professionals. Similarly, those with less experience rated artificial intelligence and databases more important than did more experienced professionals.

Ratings of the importance of curricular components (listed earlier) yielded more interesting results. Overall, the five components that were rated most important (averaged over both long- and short-term) were: probability and statistics, science, discrete math, linear algebra, and calculus. Yet, differences were found in the short- and long-term views: capstones and internships were rated more important for the short-term whereas probability/statistics and liberal arts were rated more important for the long-term. More experienced professionals rated science and liberal arts more important than did their less-experienced counterparts, who rated internships and capstones as being more important.

In summary, those taking the long-term view and those most experienced appeared to value knowledge areas and curricular components often seen as less immediately relevant by computer science students (architecture, probability, linear algebra, science, and liberal arts). *To prepare students for the long-term, we recommend that computing departments find ways to persuasively share this perspective with them.*

Where to Go from Here: Curricular Adaptation

Although a survey dataset can numerically identify features of interest with clarity, what is less obvious is how curricula can be adapted in response. How exactly should a curriculum include science (for the long-term) and what kinds of capstone projects are sufficient (for the short-term)? How should a balance be struck between the two? We suggest a few principles to consider:

▸ Since a baccalaureate education is meant to last a lifetime and serve a variety of career paths, one should be careful not to sacrifice any area rated highly for the long-term. This is where a periodically administered survey such as this one that includes a long-term view comes in useful.

Those taking the long-term view and those most experienced appeared to value knowledge areas and curricular components often seen as less immediately relevant by computer science students.

▸ It is incumbent upon educators to, at the very least, sufficiently prepare students for graduate study in the discipline.

▸ A curriculum that prioritizes foundational skills must also include development of market-ready skills. An example might be a capstone course that includes many first-job skills, such as software tool proficiency, professional ethics, writing, and presentation.

▸ When it comes to the development of theoretical (including mathematical) knowledge, many students are either averse to it or enter college underprepared. Yet, it is difficult for them to acquire this knowledge in the workplace—it is best acquired during undergraduate study. One way to motivate students to learn theoretical knowledge might be to combine it with applications. So, courses should be reformulated to include computing applications that use theoretical knowledge. Incidentally, some areas of theoretical knowledge might be more useful in industry than others, as the ratings for probability, statistics, and linear algebra indicate.

▸ While striving to craft curricula that balance first-job readiness with long-term career preparation, institutions should also try to effectively utilize their local expertise and communicate the particular student profile of their institution to potential employers.

▸ For recruitment, employers might want to use a combination of theoretical (for the long-term) and practical (for the short-term) assessment, which would signal to students the importance of also developing foundational skills.

A Call to Action: Future Data Collection

In closing, we make the case for periodically administering a national survey of computing industry professionals focused on curricular feedback. (A similar survey of faculty on the goals of computer science education was recently reported elsewhere.[3]) A rich and growing dataset from such surveys will help educators balance long-term value with first-job preparedness. More importantly, repeated surveys have the potential to dive deeper into rapidly changing areas such as artificial intelligence and data science, and elicit in sufficient detail, particular subtopics of consequence that may have arisen since prior surveys. Multiple rounds of surveys over multiple years also have a better chance of reaching a variety of computing practitioners. Finally, surveys, their analysis, and subsequent curricular refinement will lead to a better mutual understanding between employers and academia and reduce the gap in employers' expectations of computing graduates.

References

1. ACM/IEEE-CS Joint Task Force on Computing Curricula. Computer Science Curricula 2013. *Technical Report.* ACM Press and IEEE Computer Society Press (2013); 10.1145/2534860
2. *Raj, R.K. and Kumar, A.N. Toward computer science curricular guidelines 2023 (CS2023). ACM Inroads 13*, 4 (Nov. 2022); 10.1145/3571092
3. *Valstar, S. et al. A quantitative study of faculty views on the goals of an undergraduate CS program and preparing students for industry. In Proceedings of the 2020 ACM Conf. on Intern. Computing Education Research (ICER '20).* ACM, NY (2020); 10.1145/3372782.3406277

Rahul Simha (simha@gwu.edu) is a professor in the Department of Computer Science at The George Washington University, Washington, D.C., USA.

Amruth Kumar (amruth@ramapo.edu) is a professor of computer science at the School of Theoretical and Applied Science, Ramapo College, Mahwah, NJ, USA.

Rajendra Raj (rkr@cs.rit.edu) is a professor in the Department of Computer Science at Rochester Institute of Technology, Rochester, NY, USA.