Computer Scientist, Software Engineer, or IT Professional:

Which Do You Think You Are?

Keith W. Miller and Jeffrey Voas

In our first year of publication, *IT Professional* included an article by Bill Lowell and Angela Burgess titled, "A Moving Target: Studies Try to Define the IT Workforce."[1] In that article, the authors complained that job titles were being invented and qualifications were shifting daily. They suggested a "core four" out of the plethora of job titles: computer scientist, computer engineer, systems analyst, and programmer. In a more recent article, Robert Glass, Venkataraman Ramesh, and Iris Vessey broke down the industry using the academic subdivisions of computer science, software engineering, and information systems.[2] Given the fast rate of change in this industry, perhaps it's an appropriate time to revisit what an "IT professional" really is.

One "official" attempt to define the IT workforce is the US Bureau of Labor's list of IT jobs (www.bls.gov/oes/current/oes_nat.htm#b15-0000), which presents the following list under computer and mathematical occupations:

- computer and information scientists, research,
- computer programmers,
- computer software engineers, applications,
- computer software engineers, systems software,
- computer support specialists,
- computer systems analysts,
- database administrators,
- network and computer systems administrators,
- network systems and data communications analysts, and
- computer specialists, all others.

For our purposes, we eliminated classifications such as "actuaries" and "statisticians," which were also included in this group. Yet, the remaining entries still form a long list. As a simplification, we offer three terms that might be a closer fit to somewhat distinct "identities" in IT today:

- computer scientist,
- software engineer, and
- IT professional.

Substantial overlap exists among the three, but each has its distinct emphases, so it seems useful to examine both how we use the terms and what the relationship is among them.

Distinctive Ends for Scientists and Engineers

A broad distinction between scientists and engineers is that scientists study things and engineers build things. Following this logic, computer scientists study things related to computers and computing, and software engineers build systems and software—ideally, following reasonable methods and procedures. Are there practicing software engineers who study things in order to build better systems? Yes, and some people who call themselves computer scientists build large, practical software systems. However, if you identify yourself as a computer scientist, you're more likely to think of yourself as a scholar and a researcher; if you identify yourself as a software engineer, you're more likely to think of yourself as a builder of systems for an employer, customers, and users.

If you identify yourself as a computer scientist, a central question when working on a project is, "what can I learn from this?" A software engineer's central question is, "how can I make this work better?" Both think of software artifacts as means to ends, but those ends are distinctive. A com-

Published by the IEEE Computer Society

puter scientist sees the artifact as an object of study, a source of experiments and data to analyze. A software engineer sees the artifact as a tool to accomplish a customer goal, a method to solve a practical problem. A computer scientist and a software engineer could well be interested in exactly the same piece of software—perhaps even the same aspect of it—but their goals will likely be quite different.

A Bigger Tent

How does an IT professional compare to the first two? To begin, "IT professional" seems to be a broader term—it's broad enough to contain all of the Bureau of Labor categories and more. "Information technology" is far more inclusive than "computer" or "software" because it surely includes both, as well as telecommunications hardware and software, data protection concerns that might be tangential to most scientists and engineers, and the many devices and services dedicated to helping users. Those concerns can differ considerably from the artifact-development issues that are central for computer scientists and software engineers.

"Professional" has at least two distinct connotations. Sometimes it designates a "strongly differentiated occupation"—a phrase coined to refer to jobs in which members are allowed to do things that nonmembers aren't. For example, only doctors can prescribe drugs or perform surgery, lawyers can keep whatever their clients tell them confidential, and police officers can use force and do things for which ordinary citizens would be arrested.[3] Strongly differentiated occupations often have strong professional organizations (such as medical or state bar associations) and strict guidelines for entry into the profession (college degrees and tests for doctors and
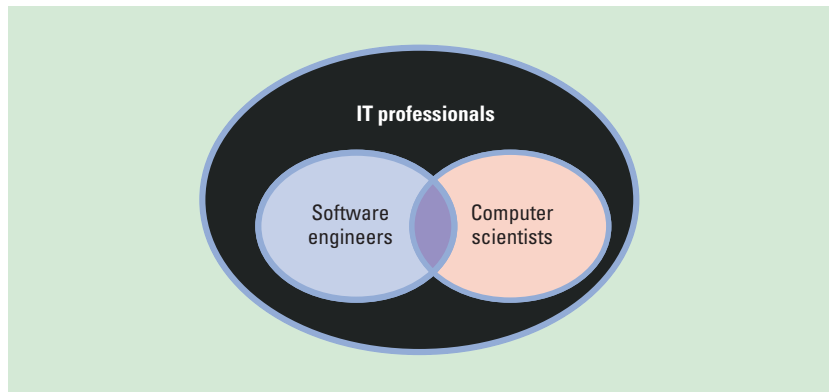
Figure 1. IT workforce classifications. The relationship between our three terms includes considerable overlap.

lawyers, training academies and tests for police, and so on).

Organizations, college degrees, and tests exist for IT professionals, but none are required, and ordinary citizens can't be arrested for doing what most IT professionals do. So, the "strongly differentiated" definition doesn't seem like a good fit. A less stringent definition seems appropriate in this case: "professionals" get paid for doing something, whereas "amateurs" might do the same thing but not for money. Although this definition applies to IT professionals, it seems too informal for what most people mean when they think of IT careers.

Perhaps we should instead follow Deborah Johnson's suggestion that IT professionals are actually somewhere in between the two definitions we just considered—more differentiated than some occupations, but not yet strongly so.

Licensing and Differentiation

Although IT professionals seems too broad a designation to be strongly differentiated, a movement is under way (particularly in the IEEE) to recognize software engineering as a strongly differentiated profession. The IEEE and the ACM have already approved a

code of ethics for software engineers, and the IEEE is establishing a software engineering body of knowledge and curriculum.

Part of the move to professionalize software engineering relates to advocacy for licensing software engineers—a somewhat controversial aspect of the project. In fact, after the software engineering code of ethics was approved, the ACM dropped out of the project, largely because its leadership opposes licensing.[4]

This very public debate highlights an important difference between software engineers, on the one hand, and computer scientists and IT professionals on the other: there's no high-profile controversy about the potential licensing of computer scientists or IT professionals. This is largely because only software engineers have become sufficiently differentiated to merit a public discussion of the possibility. (One US state, Texas, already licenses software engineers.)

Under the IT Pro* Tent

Figure 1 displays our understanding of the relationship among the three terms. IT professionals is the "big tent," but many people who belong under it think of themselves as something else first: software engineers, database

administrators, Web designers, and so on. If you identify strongly enough with a subspecialty, you might not think of yourself as an IT pro at all. For example, a computer engineer designing a very-large-scale integration (VLSI) chip for a cell phone might not spend much time thinking about his or her place in the world as an IT professional, even though the definition fits. On the other hand, if you don't feel a tight affiliation to another description, you might think of IT professional as the most useful description—your "home" in the IT world.

throughout them. For example, jobs such as "computer support specialist" and "system administrator" suggest service orientations. In contrast with software engineers and computer scientists, many people identified in IT pro* focus less on hardware and software artifacts and more on how humans interact with those artifacts.

Returning to the US Bureau of Labor list, the descriptions of such jobs as "system administrator" and "database analyst" emphasize the need for people who can build and maintain systems that work

Of course, how you define yourself as a professional is more than a game about what you read. Your professional identification influences how you approach your job, and it matters in how you plan your career. Our lives are more than our work, but our work is surely a major part of our lives. Having a clearer vision about where you want to belong in the world of IT could help you get where you want to go. 🎛

References

1. B. Lowell and A. Burgess, "A Moving Target: Studies Try to Define the IT Workforce," *IT Professional*, vol.1, no. 3, 1999, pp. 33–37.
2. R.L. Glass, V. Ramesh, and I. Vessey, "An Analysis of Research in Computing Disciplines," *Comm. ACM*, vol. 47, no. 6, 2004, pp. 89–94.
3. D.G. Johnson, *Computer Ethics*, 3rd ed., Prentice-Hall, 2000.
4. J.C. Knight and N.G. Leveson, "Should Software Engineers Be Licensed?" *Comm. ACM*, vol. 45, no. 11, 2002, pp. 87–90.

> **Of course, how you define yourself as a professional is more than a game about what you read.**

The Venn diagram might suggest that IT professionals is too broad to be useful, but the term can be helpful in excluding, for example, people who use computers in their jobs but don't focus on the IT aspects. For example, a teacher who daily uses computing to perform research and develop and disseminate information uses IT continually but doesn't fit our definition of an IT professional.

Another way to look at Figure 1 is to focus on the IT professionals who are neither software engineers nor computer scientists. We think this subset, which we call IT pro* = {IT professionals} – {software engineers} – {computer scientists}, is a significant group of individuals who identify most with the term IT professionals, whereas software engineers and computer scientists identify more with their more specialized concerns. Although IT pro* is a large, heterogeneous collection of occupations, certain themes run

for specific groups or organizations. IT pro* members are less inventors and scholars, and more administrators and facilitators. These occupations require up-to-date information on systems, software, and new ideas—but usually in the context of solving or avoiding problems that users are having or might have in the future.

Perhaps a quick-and-dirty way to locate yourself on the Venn diagram in Figure 1 is to identify your favorite IEEE publication. If you're most eager to receive *IEEE Transactions on Software Engineering*, you probably think of yourself as a computer scientist. If your favorite periodical is *IEEE Software*, you probably see yourself primarily as a software engineer. If *IT Professional* is your favorite (thank you very much), our first guess is that you see yourself as an IT pro*.

Keith W. Miller is a professor of computer science at the University of Illinois at Springfield. His professional interests include software testing and computer ethics. Miller is editor in chief of IEEE Technology and Society *magazine, and he helped write the joint IEEE and ACM Software Engineering Code of Ethics and Professional Practice. Contact him at miller.keith@uis.edu.*

Jeffrey Voas is the director of system assurance at SAIC. His professional interests include software testing, reliability, fault tolerance, and safety. Voas is past president of the IEEE Reliability Society and a current member of the IEEE Computer Society board of governors. Contact him at jeffrey.m.voas@saic.com.