

Microsoft

The new Visual Studio 2008 Beta 2. It's here.
DEFY ALL CHALLENGES



Get it now



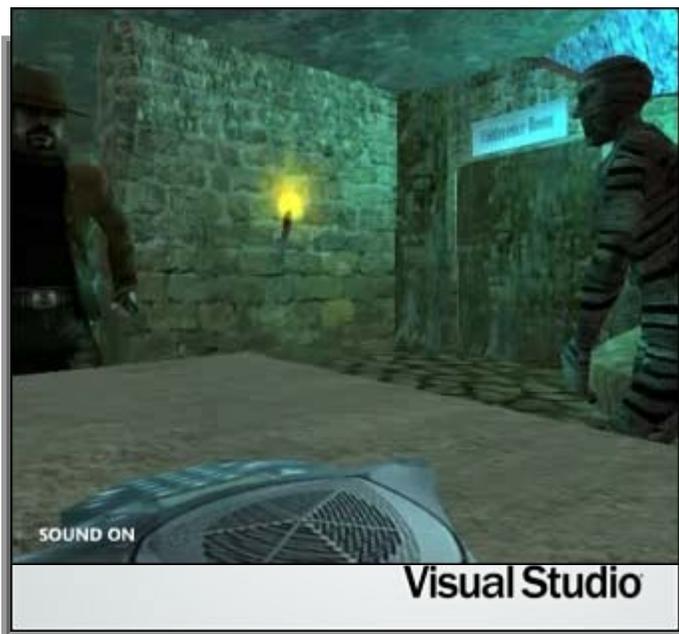
Programming Grads Meet a Skills Gap in the Real World

September 3, 2007

By Darryl K. Taft

Despite the best laid plans of colleges and universities, there remains a skills gap between what computer science graduates learn in their undergraduate years and what they need to become proficient in a typical at-work environment.

In short, many people on both sides of the equation—teachers as well as potential employers—say the educational system is not doing enough to keep pace with the ever-changing needs of IT, and that entering the work force often is as much of an educational experience as is college, particularly for programmers.



Ari Zilka, chief technology officer at Terracotta, in San Francisco, said he knows very well about the skills gap, as he worked his way through college in the high-tech business while attending the University of California, Berkeley.

RELATED LINKS

[State of the Enterprise IT Career](#)

[IT Workers Second-Guess Career Choice](#)

[Microsoft Buys Stake in CareerBuilder.com](#)

[Career Central](#)

[Career Path: Graduate-Degreed Compliance Execs](#)

"I found that UC Berkeley had an excellent curriculum but not only was my schooling lagging behind work, it became very hard to even go to school because work had me learning the concepts and their applicability and nuances that teachers didn't even seem to know."

Zilka noted that many of the new hires he's seen during his career continue to echo the same sentiments as he did.

Some of the things the school didn't teach Zilka and many who are now entering the work force include issues around communication, development skills, and business and product design.

On the communication front, Zilka said, "Presentation skills are critical, and selling and influencing peers is critical."

To read more about the state of the enterprise IT career, click here.

Some of the development skills that schools might emphasize more include design patterns, coding

style and practices, scalability and performance tuning, and a focus on the entire software development lifecycle, Zilka said. He noted that things like quality assurance, unit testing, and stage and release are not usually taught.

Moreover, Zilka said he and his friends used to joke that they would walk out of school knowing mathematical theory, but nothing they would actually use in work.

"In reality, I find that if your school was strong and you learned compilers, OSes [operating systems], hardware design, database design, advanced math and statistics, and the like, you find ways to apply all that theory but you don't learn how to be a developer or a sys admin or any of that," Zilka said.

Bjarne Stroustrup, the creator of the C++ language and a computer science professor at Texas A&M University, said, "There is a gap. It is hard to precisely characterize because there are so many different kinds of jobs," he said. However, "It is important to remember that universities should educate—for a life time of further learning—not train for specific tasks."

Yet, "That said, many students have no clue about software development and couldn't program their way out of a paper bag—that's unnecessary and puts them into a bad situation even if their primary job function is not programming."

For example, said Stroustrup, "inter-face design, testing, use of scripting is often not taught, at best, students are assumed to absorb it by osmosis. Most don't and equate programming with fiddling with tiny programs. At TAMU [Texas A&M University], we are trying to address that. I think a bachelor's is too short time; for most jobs you really need a masters—there is too much to learn for just four years."

Rawn Shah, IBM developerWorks Community Programs Manager, agreed with Stroustrup. Shah works for IBM in Tucson, Ariz.

"Today, some of the most sought after developers are those who work with environments and ideas that are complex," Shah said. "The opportunities to do those are harder to find at the basic university level, but are much more obvious at the masters or Ph.D level."

Bill Scherlis, director of the Institute for Software Research at Carnegie Mellon University, which is home of one of the premier computer science programs in the U.S., acknowledges the skills gap and said CMU is working to help address the gap.

"We hear it all the time from our colleagues in industry that university graduates have programming skills and technical knowledge, but they are somehow not equipped to take on software engineering" roles in the corporate environment, Scherlis said.

"When graduates join organizations [after college] they are often shocked to realize they are dealing with limited resources, deadlines, fuzz requirements, requirements that change weekly, applications that scale, the use of frameworks and libraries, existing code—that may be bad code with bad design decisions, issues of interaction within and among teams, and having to develop code that is secure," Scherlis said.

Those are some of the challenges students are faced with that they may not have faced in school, he said. "And we are crafting responses into the curriculum," Scherlis said. "We have to introduce our students to the mission of real engineering and collaboration."

One way to do that is to foster more team development, Scherlis said. "They get experience and deal with the realities of interacting with other individuals." Indeed, Scherlis said, the school encourages computer science students to take a course in social psychology to better understand how teams work because team development is such a key part of the corporate development environment today.

In fact, the school sponsors collaborative projects between an external organization looking for real solutions to IT problems and student teams at CMU. "And education is secondary to the external organization; they are truly looking to our teams to help them with a problem first and foremost," Scherlis said. So both parties benefit.

Page 2: Programming Grads Meet a Skills Gap in the Real World

IBM's Shah said "software development is now more into how people can work together to solve many seemingly common issues."

In other words, "software development is becoming much more of a group activity, and there is a lot of sophistication to that in the industry that isn't being replicated in a smaller closed environment like a college," Shah said. "Very often, they simply can't because of the time limitations of the semester-based programs."

Moreover, Scherlis said the onslaught of open source technology has been a big benefit to students, as they are able to get access to source code more easily and get students up and familiar with technology they are bound to see after graduation.

Byron Sebastian, CEO of SourceLabs, in Seattle, said many less experienced developers can learn necessary workplace skills by actively working on open-source projects in their free time.

"A developer with a Computer Science degree as well as hands-on experience in shipping software through an open source project has a running start in being successful in the workplace," Sebastian said. "They get experience working with complex and distributed teams, real 'ship cycles,' real customers, and real usage of their code. My advice to a student in a university studying software engineering would be to also actively work on open source projects to acquire more 'real world' skills they can apply when they enter the job market."

To read about why some IT professionals are second-guessing their career choice, [click here](#).

John McMullen, professor of Information Systems at Monroe College, in Bronx, New York, said, "A gap exists for a few reasons. One is that colleges are mandated by accrediting agencies to teach theory and to not be 'training schools.' Most teach courses like 'Programming Logic Using Java (or C++, etc.)' to try to sneak in some practical work but students do not get intensive programming experience."

In addition, "The rapidly changing landscape of development languages and tools—Python, PHP, LSL [Linden Scripting Language, the language used in Second Life], etc.—make it really impossible for curricula to keep up with," McMullen said. "Colleges must teach would-be programmers to 'learn-how-to-learn' new languages and skills."

Eric Newcomer, CTO at Iona Technologies, in Waltham, Mass., said, "You would think that researchers in the universities would be on the forefront of technology, and while that is the case at some universities and with some professors, the universities are generally behind software companies."

Newcomer who has taught database design at the graduate level, said if you look at a typical computer science curriculum you see plenty of fundamentals—programming languages, algorithms, operating systems, database management, etc.

"But you do not usually see things like XML, Web Services, Ajax [Asynchronous JavaScript and XML, Ruby, OSGi [Open Services Gateway Initiative], Eclipse, open source, WCF [Microsoft's Windows Communication Foundation], BizTalk, SOA [Service Oriented Architecture], ESBs [Enterprise Service Bus], orchestration engines, integration strategies, etc.," Newcomer said. "And these are the topics IT departments are most concerned with, not the fundamentals. They want to know what new technologies to adopt and what they're good for."

Newcomer said he travels to universities to deliver guest lectures and has witnessed students complain about the skills gap. "I guess the summary is that the schools cover more theory than practice, and focus on teaching well established technologies rather than new ones," he said.

John Montgomery, group program manager for Microsoft's Popfly mashup tool, who has looked closely at the phenomenon of beginning developers as well as grooming developers out of college to work in his teams at Microsoft, said, by and large, the computer science curricula at the top-tier schools do a good job of graduating students with a basic understanding of things like data structures and computational algorithms.

"But it's rare that anyone gets hired to go and write a better bubble sort algorithm or that graduates are called upon to perform array arithmetic," Montgomery said.

For the most part, these are "solved problems"—the frameworks and tools that anyone would give a developer will have already implemented what's necessary to solve those problems, he added.

Yet, "Mostly, developers seem to be presented with ambiguous technical problems, complex development processes, and unfamiliar code bases to work with," Montgomery said. "They'll be likely to be handed some chunk of existing code and expected to add a new feature or resolve a bug. Some schools try to train in the abstract for this kind of problem set, presenting students with classes in development processes."

Page 2: Programming Grads Meet a Skills Gap in the Real World

To Todd Williams, vice president of technology at Genuitec, a Plano, Texas, software tools maker, the best employment candidate fresh out of school is the one who realizes the code they do not have to write may be the best code there is.

Williams said he thinks some schools focus too much on writing software as an art form for the developers to express themselves. But finding a way to accomplish a task without writing a bunch of new code pleases Williams, he said.

"New code equals new bugs equals maintenance burden, equals higher life-cycle costs," he said.

"In fact, if you have to hand-write code you've already admitted that you're going to take the slowest most expensive route available to implement a solution," Williams said. "Most software isn't unique and the issues it addresses can easily be solved by either finding, reusing and integrating existing software or through the use of productivity-enhancing tools that generate most of the code for you. Then, the clever developer only has to string the solution together with the minimal 'code wiring' that makes the pieces function well together."

To read about how to get ready for fall job hunting season, click here.

Williams' employer, Genuitec, sells the MyEclipse IDE (integrated development environment). MyEclipse 6.0, the latest version of the technology, supports both code generation as well as old-fashioned hand coding.

Joe Ponczak, CEO of Codign Software, in Baltimore, said of the programming skills issue: "That's a pretty wide gap there. I think the developers that excel after college are the ones who continuously challenge themselves to learn new patterns, languages and methodologies while in college."

Moreover, although colleges do a good job at teaching the basics, "I think the basics are now 'too' basic and need to be improved," Ponczak said.

For instance, a look at the curricula at several local universities showed a listing of courses that have been taught the same way for years, Ponczak said.

But, "There were few, if any, courses on development methodologies, TDD [test-driven development], emerging technologies and languages, multi-layer development, security, scripting, refactoring, metrics or open source."

Meanwhile, SourceLabs' Sebastian said shipping software from a commercial vendor requires "ruthless pragmatism and customer focus" to be able to quickly deliver features customers require in a high-quality but timely fashion. And being able to make rapid design, feature, and bug triage decisions based on these factors is a skill that is usually acquired through hands-on experience and work alongside professional engineers with years of experience doing so, rather than in the classroom, he said.

While some developers have a natural "knack" for this, it's not a skill that is easily acquired in school.

In addition, large complex systems in the real world have multiple integration points, dependencies,

and are often built by multiple teams, Sebastian said. "Learning to work in this environment, and having the skills to design and execute on the integration, assembly, and testing of components is another skill that tends to be acquired in the work-place rather than in school," he said.

Chris Stephenson, executive director of the Computer Science Teachers Association, in New York, said she too sees the skills gap between what students learn in school and what they need in the work force, "but what is really exciting is that I have seen more and more educators (both at the K-12 level and the university level) willing to make these skills part of their curriculum."

Like others, Stephenson said computer science should no longer be taught as a solitary and isolated discipline.

"There is little effort made to address issues such as effective team work, project planning and time management, and conflict resolution let alone helping students gain the cultural competencies and effective communication skills that are the key to success in a global economy," Stephenson said.

"Also, not enough effort has been made to show students how computing connects to problem solving in the real world," Stephenson said. "The good news, however, is that an increasing number of educators are building these skills into the classroom experience. Teachers now have students work in teams on real world projects where the failure to plan together, work together, and communicate effectively are a big part of the evaluation that the students receive."

Microsoft's Montgomery listed a series of things he would like to see in higher education computer science programs.

One is technology education programs that integrate communication skills. "The best developers are often the ones who can explain problems and solutions the most clearly to others," he said

The second is technology education programs that emphasize teams. "Very few developers really work alone," Montgomery said.

The third is graduates with analytical skills, particularly around ambiguous problems.

"It's important that developers understand the intention of what they're being asked to do as well as the implications of a solution they're thinking of and can weight and communicate these," Montgomery said.

The fourth is graduates with an understanding of development processes. "Not a theoretical one—they need to work on teams that use formal, top-down development process, agile development, teams with other developers, teams with test processes, and so on," he said.

The fifth is graduates with an ability to learn on the fly.

And the sixth is graduates with competence in several programming languages. "C++ is typically a must; C# or some other managed-code language is also mandatory," Montgomery said. However, competence in one dynamic language, such as JavaScript, should also be present, he said. And the graduate should have the ability to know which to use when.

In addition, Montgomery said he believes that for the U.S. to compete on global level people in various domains will have to attain some level of technical expertise.

"The diffusion of technical skills into many domains means that finance majors, doctors, and so on also need to have strong technical skills—what may have been considered development skills a few years ago—to compete on a global stage," Montgomery said. "This new demand is causing leading schools to nurture the growth of MIS/CIS/IT minors as rich programs unto their own."

Check out eWEEK.com's Application Development Center for the latest news, reviews and analysis in programming environments and developer tools.

