

# Certification

Adrian Kosmaczewski

2008-08-05

While several other professions have a long, established and standard procedure of certification, the title “software engineer” is applied to both self-made developers, turned into experts of some technique, or to people with PhD degrees, and a long history of both academic and professional achievements.

When in some situations it is not legally possible to use the title “software engineer” without an engineering degree of some kind (for example, in some states of the USA or some institutions like the IEEE - <http://www.ieeeusa.org/policy/positions/titleengineer.html>), the term “software developer” is usually applied to people in charge of designing, writing and / or maintaining software-based systems. I will use the terms developer and engineer interchangeably in this discussion, which some people might think is not correct.

The discussion about the need of a formal certification process is a relatively new one:

Professional certification in the IT industry is a relatively recent phenomenon. It was begun in the late 1980s by Novell, Inc., an upstart networking vendor from Provo, Utah, in an effort to build market share and manage support costs for its products by building the skill levels of the people who worked with those products. Novell was one of the first companies to recognize the links between education/skills and product success. They knew that they could not build an education infrastructure that would support their worldwide marketing plans with their own resources. However, they also recognized that if they did not provide for skills acquisition for their highly technical products, they could never meet their product revenue goals.

(Shore)

However, no consensus about whether or not certification is needed has been reached yet. This article will highlight some of the problems raised by software engineering certification, which might explain the lack of consensus cited before:

- The first one has to do with the inherent extension of the software engineering field: are all software developers equal?

- The second one has to do with the large number of available certifications: which one to choose? Which ones are “reliable” indicators of expertise, and in which fields?

## What is a “Software Engineer”?

In my career, I’ve found self-made people (I’m one of them, actually), real-estate architects, lawyers, mathematicians, economists and even geophysicists writing code for a life. What I’ve seen so far is that the most successful software developers are those who like doing it, no matter which profession they’ve followed. And the opposite is also true: many guys with a computer science degree discover, some time after they start their careers, that they definitely do not like that code thing.

One of the biggest problems with certifications is that there is not such thing as a “single kind” of software developer:

- There are those who write games, and spend most of their time writing in low-level languages for game consoles, optimizing for speed and space, and creating three-dimensional worlds using as little memory as possible...
- There are those who write web-based applications, and spend their time creating 3-tier architectures, talking to a database, using some kind of object-oriented platform, and luckily exposing some data using XML web services, dealing with cross-browser issues, and wondering what is all that fuss about Web 2.0...
- There are those who write operating systems, and work for some embedded software company, or hack Linux kernel device drivers every night, or work for Microsoft or Sun or Apple, and spend most of their time discussing whether microkernels are better than monolithic architectures...
- There are those who have the ill fate of working as a consultant, and spend more time switching from project to project every day, or dealing more with corporate politics, rather than with code...
- There are those who manage projects and spend more time in their mailing list or in Microsoft Project rather than being able to code (and then complain about this in their blogs)...
- There are those who have a software engineering degree, but work for ZDNet writing about industry trends...
- There are those who turn into human resource consultants, and try to keep up to date on the new trends, but feel completely lost given what they learnt in university...
- There are those who do a little bit of all what I’ve mentioned above, and are or not really good at all of them...
- And finally there are those who might fit any of the characteristics above, but would have preferred not to listen their parents and rather open that scuba-diving shop in Honolulu.

## Available Certifications

This diversity explains the existence of more common product-specific certifications: you can be certified to use Microsoft technologies (<http://www.microsoft.com/learning/mcp/default.msp>), MySQL databases (<http://www.mysql.com/certification/>), Apple servers (<http://www.apple.com/xserve/raid/certifications.html>), various IBM products (<http://www-03.ibm.com/certify/certs/index.shtml>), Java development stacks (<http://www.sun.com/training/certification/java/index.xml>), Cisco routers (<http://forums.cisco.com/eforum/servlet/CCNP?page=main>), RedHat Linux installations (<https://www.redhat.com/training/certification/>) or UML diagrams (<http://www.omg.org/uml-certification/index.htm>)

However, given that technology companies have interest in having many people taking their certifications, their affordability and low-entry barriers to get them, many of these become much easier to get than they should be, and as a result, they lose credibility, and do not help IT recruiters to filter properly software developers during the selection processes. I've heard many complaints of project managers regarding these certifications, and I think it's a generalized feeling:

“Certified skills pay has not just flat lined, it's in the negative. This is big news if you're certified and you're thinking about getting re-certified,” said Foote. “This trend is in the fourth quarter, that pay for certifications is on the wane, while non-certified skills are growing in pay.”(...) Certifications are losing value because employers are looking for more in their workers than the ability to pass an exam; they want business-articulate IT pros.”

(Rothberg, 2006)

Bruce Schneier, a well-known security researcher, has written about security certifications as well, with a mixed feeling:

In the end, certifications are like profiling. They work, but they're sloppy. Just because someone has a particular certification doesn't mean that he has the security expertise you're looking for (in other words, there are false positives). And just because someone doesn't have a security certification doesn't mean that he doesn't have the required security expertise (false negatives). But we use them for the same reason we profile: We don't have the time, patience, or ability to test for what we're looking for explicitly.

(Schneier, 2006)

The conclusion of all of this is that the debate is pretty much still open, and that there is not a simple answer to it.

## Market Fragmentation

There is an interesting anonymous comment in Schneier's website as well:

Another thought on certification is they are not all equal.

There are Vendor Certs. Microsoft's MCP/MCSE, CISCO CCNA/CCNP/CCIE Pro: The candidate is likely to know how to work on your specific platform. Con: The candidate is likely to think in only the vendor's interest.

There are Certs to assure knowledge of standard security terminology. ISC CISSP Pro: Can talk strategy and evaluate the nine domains to evaluate how the company is doing overall Cons: Most likely could not tell you what the ninth byte of an ip packet means or if OpenSSL is out of date on Red Hat Linux.

Topic specific, vendor neutral. SANS GIAC Pro: Vendor neutral. A lot of focus on specific skills in NIDS or Hardening Windows, Incident Handling, etc. Con: Concentration on open source tools since they are easily available, but it does not seem to impress all employers.

(@nonymou5, in Schneier, 2006, spelling mistakes not corrected)

I think that this comment summarizes pretty well another problem with certifications: there is a great level of fragmentation in today's market. Every single important technology in the IT world requires a huge investment in time and practice in order to master it, and this translates in a huge complexity for the developers to choose the right certification. All of these without taking into account the large number of IT-related university degrees available, online or not.

## Conclusion

The term "software engineer" is sufficiently vague, and the number of "certifications" sufficiently large, as to allow a single "yes or no" answer to whether professionals in the software sector should be certified or not. I personally think that I would rather avoid vendor-specific certifications as far as possible, and choose university-related or problem domain related certifications instead, to keep my career options open, and my mind free of marketing.

## References

Rothberg, Deborah; "Another Nail in the IT Certification Coffin", November 3rd, 2006, [Internet] <http://www.eweek.com/article2/0,1895,2051272,00.asp> (Accessed June 3rd, 2007)

Schneier, Bruce; "Security Certifications", July 20th, 2006, [Internet] [http://www.schneier.com/blog/archives/2006/07/security\\_certif.html](http://www.schneier.com/blog/archives/2006/07/security_certif.html) (Accessed June 3rd, 2007)

Shore, Julie; "Why Certification? The Applicability of IT Certifications to College and University Curricula", [Internet] <http://www.developer.ibm.com/university/scholars/certification/ebu-certification.pdf> (Accessed June 3rd, 2007)