



Lifelong Learning for Lifelong Employment

Philippe Kruchten

SOMEONE ASKED ME recently, “How do we keep 40-year-old software developers employed?” At first I was puzzled. I had little clue this was a problem. Isn’t there more demand than supply for software developers?

However, imagine a software developer who graduates from a good engineering school and gets a good job in a large high-tech company. He marries and raises a family, is good at barbecue, runs in triathlons, doesn’t miss a single French movie in town, has a large collection of rock LPs, and so on. Then, after 20 years at his job, owing to a series of economic changes, restructurings, and acquisitions, he finds himself suddenly unemployed. He realizes that much of the technology he’s been working on for the past 15 years was valued in his company but is hardly sellable in today’s job market. Actually, none of the technologies he has mastered are in use in any of the companies he’s interviewing with.

“What do I know about the MEAN stack? What ‘mean’ stack?”

What can this developer do to improve his employment chances?

The State of Software Development Work

Several forces are at play in the software development world:

- *Software churn.* The half-life of good software engineering knowledge, tools, and technologies is about five years. This means that after 20 years, less than 10 percent of what you know is still useful.
- *A contingent workforce.* We’re moving rapidly into a situation in which teams are assembled rapidly out of a contingent workforce: skilled freelancers who offer their services for the duration of a project or less and aren’t on the payroll.
- *The global economy.* Your competitors aren’t in your city,

didn’t study at the same school as you, and so on. They might be far away, they don’t need a visa to compete with you, and they can live very well with half your salary.

- *A cloud-based, connected world.* You might even never be in the same room as your teammates. Today I just had two meetings with people in three time zones, over the Internet, with video, shared screens, and so on, all from the comfort of my home. All our “belongings” are in Google docs and cloud repositories.

These forces are converging to a critical need for lifelong, or continuous, learning.

The Lifelong Learner

What is lifelong learning, and how do we learn to be lifelong learners? Lifelong learning is peoples’ “ability to identify and to address their own educational needs in a chang-

ing world in ways sufficient to maintain their competence and to allow them to contribute to the advancement of knowledge.”¹ Or, at least it’s the ability to perform adequately in your job or, more simply, to remain employable. Until 50 or so years ago, we had assumed that the learning phase of our lives was at school: college, university, and engineering school.

points for prompting self-reflection. If you don’t get performance reviews or they’re too far apart or just a vacuous ceremony, ask for feedback. (But then listen to the feedback; don’t become defensive and belligerent!)

Understand Your Work Style, Especially in Groups

There are easy ways to better understand how you work:

a book, some examples, a demo or two, or some practice? In my case, I can’t just learn by doing or watching someone doing something if I don’t have some kind of mental (or conceptual) model of the process.

Build a Network

Be a part of the professional community. Don’t just be a bystander, a consumer of knowledge when in need. Contribute. Make presentations. Help organize learning events. Write your experience and share. Blog. You’ll get some feedback, and the simple process of writing forces you to clarify your ideas. (I write to better understand what I think.)

Find mentors and role models. What people in your organization (or, better, outside your organization) seem successful to you? Try to emulate them. If they’re approachable, ask them to be your mentor. Give back by mentoring younger colleagues.

Be So Good They Can’t Ignore You (to Paraphrase Cal Newport²)

You need to be very good at least at one thing, process, practice, or technology. That thing you’re supergood at must also be in demand, so don’t lose track of the world around you. Being the superguru of MS-DOS or supergood at APL programming doesn’t garner many brownie points nowadays. I was once very good at Ada programming!

Aim to be a “T-shaped” individual, combining deep disciplinary knowledge along with a keen ability to communicate across social, cultural, and economic boundaries. Have enough breadth of knowledge to understand how the world works, and be able to plunge into and adapt to different situations. However, master some discipline thoroughly,

We had assumed that the learning phase of our lives was at school. Not any more.

Becoming a constant learner involves the following ingredients, many of which could be summarized by the ancient aphorism “know thyself” and the neologism *metacognition*.

Self-Reflection

Once in a while, say, four to eight times a year, stop and reflect. Do this at least after any negative event, personal or team related. Wait a few hours or days until any anger has dissipated. What happened, why did it happen, and what knowledge would have prevented it from happening?

Also do this for positive events. You’ve been congratulated? Good. Reflect then on what elements contributed to your success. They might not always be what you think.

Ask for Feedback

Most people dread the annual or semiannual performance review. However, these events are great

- You can start with the Myers-Briggs Type Indicator (MBTI; www.myersbriggs.org) personality test (or some of its more recent imitators) to understand how you perceive the world around you and make decisions.
- Gallup’s StrengthsFinder (www.gallupstrengthscenter.com) is better for seeing how you perform in teams.
- The Belbin model (www.belbin.com) is more complex and richer, and uses your coworkers’ input; it’s not based merely on your self-assessment, with its inherent bias.

I’ve found the Belbin model more useful by far for reflecting on my working style in teams.

Know Your Learning Style

Into which of David Kolb’s learning styles do you fall—concrete experience or abstract conceptualization, reflective observation or active experimentation? Need a lecture,

with analytic thinking and problem-solving skills so that you can add value, not just be there for decoration. Then grow another strength, more forward looking, and become a “pi-shaped” (π) individual.

Develop Critical Thinking

Learn more about critical thinking (which isn’t the same as critiquing, something everyone seems good at). Critical thinking is the objective analysis and evaluation of an issue to form a judgment. According to Richard Paul and Linda Elder, we reason

*for a purpose, within a point of view, based on some assumptions, leading to implications and consequences. We use data, facts, and experiences to make inferences and judgments, based on concepts and theories to answer a question or solve a problem.*³

Learning, in Practice

How much learning do you need to do, then? That’s hard to define precisely. But looking at the rules for various professions, I’d say that 80 hours a year is a minimum. That’s about 10 full days. “I can’t spend that much outside of work,” someone told me. Well, you should, really. Paradoxically, current employment is often the major obstacle to lifelong learning, although you might be able to sell using work time to explore new technologies to your current employer. Employers also need to realize that their best employees will bail on them if those employees can’t remain current.

To satisfy the requirements for continuous development, you need the following.

Adequate Programs

There are plenty of adequate programs to choose from. Start with

your local community and look for user groups: Java user groups, Linux user groups, and so on. Then look for larger societies that might have something in town (or on the Web): ACM, the IEEE Computer Society, the Project Management Institute, the Scrum Alliance, and so on. Start something new with a Startup Weekend or MakerFaire.

Thanks to the Internet, you can access MOOCs (massive open online courses) from reputable sources (such as Coursera, Udacity, edX, the Khan Academy, Codecademy, and Code School). And yes, not all of this is necessarily free. The really good stuff has a price. Aim at some certification, but aim for the programs that are recognized and demanding. Align them to the vertical bar of your T (or pi).

Incentives

To maintain my certification as a professional software developer (IEEE Certified Software Development Professional), I need about 90 hours a year. To maintain my Canadian professional-engineer license, I must be able to demonstrate an average of 80 hours a year. In general, any professional certification I respect would comply with ISO 17024:2012. It would also require proof of currency; that is, if you don’t keep your competencies current, you lose your certification.

To Get Organized

If you want to benefit from all this learning, you might want to get organized for the long haul. How do you take or keep notes, and where? From little notebooks to online diaries, to folders full of oddly named PDFs, with some tooling such as Evernote? Sometimes you have to review the good learning you did two or three

years ago. But if you can’t retrieve the information easily, you haven’t fully benefitted from it, have you? You can’t rely solely on your memory.

Read, Read, Read

You need to read. Read more than Twitter-length sound bites (although Twitter is a good way to find what to read). You also need to read more solid stuff than collections of blog entries, which usually only scratch the surface. They don’t give enough foundation or concepts and might be just someone’s opinions based on a sample of one. When I was interviewing candidates, I always asked, “What technical books or publications have you read in the last 12 months?” Attending a conference is a good complement or alternative; you’ll discover new technologies and have more opportunities to develop your network.

Well, if you’ve at least read this article, you’re off to a good start! ☺

References

1. Canadian Eng. Accreditation Board, *Accreditation Criteria and Procedures*, Engineers Canada, 2008.
2. C. Newport, *So Good They Can’t Ignore You: Why Skills Trump Passion in the Quest for Work You Love*, Business Plus, 2012.
3. R.W. Paul and L. Elder, *The Miniature Guide to Critical Thinking: Concepts and Tools*, Foundation for Critical Thinking, 1999.

PHILIPPE KRUCHTEN is a professor of software engineering in the University of British Columbia’s Electrical and Computer Engineering department. Contact him at kruchten@ieee.org.



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.