



Hands On Technology Transfer, Inc.

# White Paper

---

## The Problem of Pseudo-Training

### *Abstract*

*Many computer programmers and IT professionals are largely self-taught, but the sort of knowledge that one can get from Q&A on the WWW is no substitute for formal, comprehensive, in-depth training. But don't believe us. Look at history.*

---

### **THE SELF-TAUGHT PROGRAMMER**

There are a great many programmers who are self-taught<sup>1</sup>, and there is nothing wrong with that in and of itself. These folks, typically millennials who grew up in a world where computers of one type or another were everywhere,<sup>2</sup> have an almost instinctive feel for logic and languages. But a piecemeal approach to learning to program is a dangerous approach if it is not employed in conjunction with a structured methodology. A lot of major programming errors -- ones that result in huge costs to businesses and people -- occur because of amateurish programming, programming done by programmers who are intelligent, but not professional. That is, they are bright, hard-working, effective, thorough, and dedicated, but far too often have learned their trade haphazardly, without any formal thought or planning.

As an educator, I am frequently shocked at the basic mistakes that even experienced programmers make, often because their "training" has consisted almost entirely of grabbing answers off the web (especially in the cases of HTML, CSS and JavaScript), and from sites like Quora, Stack Overflow, and various technical communities. Don't get me wrong -- these are fabulous resources. But they provide opinions and answers to specific questions from individuals, and this is not even remotely the same thing as training.

### **FLIMSY FOUNDATIONS**

In a recent Python Programming course that I was auditing (well, supervising)<sup>3</sup>, one student, a senior programmer in a large firm, made it clear with his questions that he did not have the slightest understanding of whether and why it was important to know the amount of space a language reserved for floating point numbers. He had never encountered a case where it mattered. Well, of course it matters! He and his employers were fortunate that he had never run across a programming challenge where he could have corrupted a large value or wasted reams of memory by storing 8 bit values in 64 bit spaces, or vice versa.

I also cannot tell you how many times (many!) I have seen programmers mucking about with SQL while knowing nothing at all about RDB design, and then scratching their heads when their "improvements" slowed the system to a crawl. And virtually any experienced software engineer can cite instances where an upfront understanding of underlying software architecture would have cut hardware costs and sped up application performance by several orders of magnitude.

## **THE COST OF EASY ANSWERS: DISASTERS CAUSED BY AMATEURISH PROGRAMMING**

It's a real problem, this lack of in-depth knowledge, with real consequences.

Recently, a bit of sloppy programming appears to have caused at least 21 suicides in India.<sup>4</sup> It seems that in the Indian educational system, test results have a profound effect on the future of students,<sup>5</sup> since India's higher education system does not have the capacity to educate all the students who are qualified. A suboptimal score eliminates students from consideration for admittance to many institutions, and students who fail to achieve the necessary scores often feel, right or wrong, that they have no future. In early 2019, about 1,000,000 students took compulsory government-sponsored placement tests, and 350,000 failed. The number of failures was far higher than expected, was unlike any previous result, and was, it turns out, wrong. But before the error was identified and publicized, at least 21 students committed suicide. A government committee set up to investigate the situation concluded that errors in the software, coupled with failed attempts to rectify the errors, were to blame.<sup>6</sup>

There are many similar cases. In 1990, over 50 percent of AT&T's network crashed, and 75 million calls failed to connect, because of a standard software update that had an undetected bug.<sup>7</sup> We have no idea how many emergency calls went unanswered and how many died as a result.

In 1978, the Hartford Coliseum collapsed due to a glitch in the CAD software used to design the roof.<sup>8</sup> If the roof had fallen just six hours sooner, hundreds or thousands would have died.

There are endless cases where amateurish errors on ecommerce sites -- poor UX design, sites unable to handle peak traffic, a simple failure to debug, the accidental use of test code in a production environment -- have had the result of confusing or driving away customers and costing sales from hundreds to thousands of dollars.<sup>9</sup>

## **PSEUDO-TRAINING**

Clearly, one of the major causes of poor programming is the phenomenon of *pseudo-training*. When enterprises allow untrained or minimally trained "programmers" to develop code, the coders all too often figure out how to solve a particular problem by finding code examples or answers online, at one of the many excellent technical communities that are available free.<sup>10</sup>

**But Q&A is not training.** Lacking a more robust understanding of the software frameworks employed or the possible side effects of certain coding practices, such programmers create apparent solutions that pass routine QA but fail in the long run. And even if the software works, performance suffers due to poor underlying architectural design.

## WHAT'S TO BE DONE?

A large part of the solution is for enterprises to provide real training that delivers architectural understanding and expert guidance in the application domain. In-depth coverage, comprehensive hands-on labs and personal facilitation help programmers develop the thorough, practical competence that saves time and boosts the quality of work product. That is why we at Hands On Technical Training focus on all three of those elements in every course we deliver, whether the course is delivered live face-to-face, remote-live, or via facilitated on-demand streaming.

## CONCLUSIONS

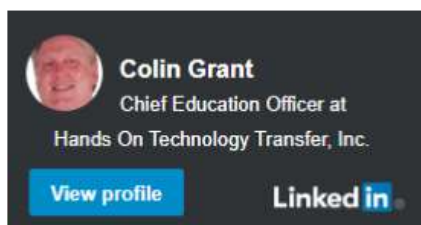
We laugh out loud at the idea of a self-taught medical doctor or self-taught structural engineer or even a self-taught beautician (the chemicals used in coloring hair can be dangerous!). But we allow self-taught or mostly self-taught programmers to create computer code that does everything from keep our money secure to assure that the car stops when we step on the brake. This is not a good idea.

Even the best, brightest programmers in the world need formal training and mentoring from real subject matter experts and trained technical educators. The pseudo-training that is so useful for answering certain questions on the WWW is no substitute for actual technical training.

*Colin Grant*

Chief Education Officer

[Hands On Technology Transfer, Inc.](#)



Colin Grant, the CEO (Chief Education Officer) at [Hands On Technology Transfer, Inc.](#), has worked in the field of technical education for almost 40 years as a technical trainer, programmer, course developer, media developer, manager, executive, entrepreneur, editor, and author. He welcomes your comments, criticism and input. Please contact him at [Colin.Grant@traininghott.com](mailto:Colin.Grant@traininghott.com).

---

## ENDNOTES

- <sup>1</sup> Student Developer Report 2018. (2018). *HackerRank*. [Online]. Available: <https://info.hackerrank.com/rs/487-WAY-049/images/HackerRank-Student-Developer-Report-2018.pdf>, accessed December, 2019.
- <sup>2</sup> Eadicicco, Lisa. (April 13, 2015). Silicon Valley's obsession with youth, summed up in one chart. *Business Insider*. [Online]. Available: <https://www.businessinsider.com/silicon-valley-age-programmer-2015-4>, accessed December, 2019.
- <sup>3</sup> The course: <https://www.traininghott.com/Courses/Python-Programming-Course.htm>
- <sup>4</sup> Shanker, M.S. (April 27, 2019). NHRC Steps In As Telangana Student Suicides Climb to 21; Private IT Firm Being Probed For Bungling Of Results? *Outlook* (magazine). [Online]. Available: <https://www.outlookindia.com/website/story/india-news-nhrc-steps-in-as-telangana-student-suicides-climb-to-21-private-it-firm-behind-bungling-of-results/329420>, accessed December, 2019.
- <sup>5</sup> Kapur, Manavi. (October 24, 2019). India's culture of high-stakes testing needs to be dismantled. *QuartzIndia*. [Online]. Available: <https://qz.com/india/1728666/indias-high-stakes-testing-culture-needs-to-be-dismantled/>, accessed December, 2019.
- <sup>6</sup> Vatyam, Nirupa. (April 25, 2019). Telangana: Software flaws messed up Inter results, concludes probe report. *The Times of India*. [Online]. Available: <https://timesofindia.indiatimes.com/city/hyderabad/telangana-software-flaws-messed-up-inter-results-concludes-probe-report/articleshow/69050995.cms>, accessed December, 2019.
- <sup>7</sup> Burke, Dennis. (November, 1995). All Circuits are Busy Now: The 1990 AT&T Long Distance Network Collapse. *California Polytechnic State University*. [Online]. Available: [http://users.csc.calpoly.edu/~jdalbey/SWE/Papers/att\\_collapse](http://users.csc.calpoly.edu/~jdalbey/SWE/Papers/att_collapse), accessed December, 2019.
- <sup>8</sup> Hartford stadium collapse: why software is just a tool and should be used wisely. (January 23, 2018). *Engineers Journal*. [Online]. Available: <http://www.engineersjournal.ie/2018/01/23/hartford-stadium-collapse-software/>, accessed December, 2019.
- <sup>9</sup> Zhou, Michael. (November 2, 2018). Disasters an Ecommerce Store Can Encounter: How to Prevent and Deal With These Nightmares. *SmallBizDaily*. [Online]. Available: <https://www.smallbizdaily.com/disasters-ecommerce-store-can-encounter-prevent-deal-these-nightmares/>, accessed December, 2019.
- <sup>10</sup> KS, Ashutosh [sic]. (June 19, 2017). Top 10 Sites to Ask All Your Programming Questions. *Hongkiat*. [Online]. Available: <https://www.hongkiat.com/blog/programming-questions-websites/>, accessed December, 2019.