Evolving Curricula for an Exponential World

Lyle N. Long Monday, April 4, 2016

Key Takeaways

- The **disparity between human learning** rates and the rates at which **technology changes can create problems,** particularly when it comes to higher education.
- If **U.S. students are to remain competitive** on a global level, dramatic **changes in curriculum** and **faculty education** are required to begin providing **computing and software education for all students**.
- In response to this challenge, **Penn State** has begun offering a variety of **minors and certificates in computational science** and **information science** for both graduate and

undergraduate students.

- To encourage such efforts elsewhere, a **nation-wide discussion is needed** on how to **evolve curricula** to benefit our students and society — as well as to reflect the fact that **we live in the information age**.

Although technology has been improving at an exponential rate for hundreds of years,[1] the structure of the human brain has remained essentially unchanged for millennia. Humans learn linearly: we retain a certain amount of information in a year, and learning twice that amount takes roughly two years. In short, we are not wired to deal with exponential change. This disconnect between human learning rates and the rates at which technology changes can create problems, particularly when it comes to education. This is dramatically illustrated in figure 1. Often the oldest and highest paid people at an organization have the least understanding of the information age and what students need.

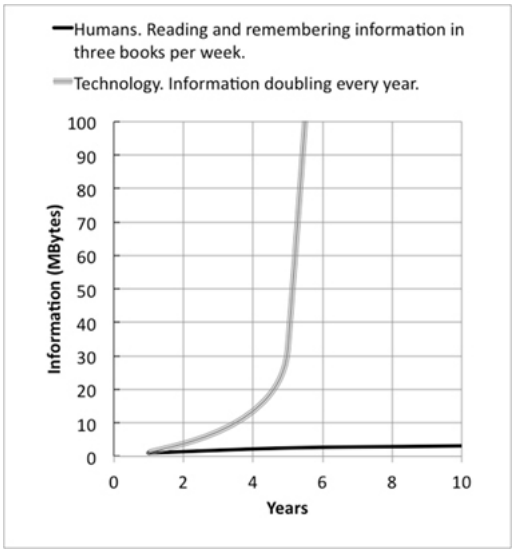

Figure 1. Human learning vs.

In higher education, considerable research and debate focus on *how* to teach and the personal skills students should learn, but *what* to teach is seldom discussed. This disregard of academic curriculum leads to a big problem: students

growth in technology

are not graduating with the knowledge they need to succeed in the long run. For example, a **National Academy of Engineering** report said that US citizens — including college graduates — are " …not equipped to make well-considered decisions or to think critically about technology."[2] Other research institutions, including the **National Academy Press**,[3] also focus on deficiencies — particularly in math and science — among college graduates, emphasizing the importance of " … foundational cognitive skills in math, reading, and writing."[4] But what of computing?

Fifty years ago, people worked in the same field throughout their lives. Today, careers typically have half-lives of 10 years or less. Entire industries — from newspaper and magazine publishing to companies built on analog technology (such as Kodak) — are being completely reimagined (if they survive at all) due to the rate of technology change. Likewise, education is in a state of potentially radical flux, as options such as online courses, MOOCs, Khan Academy, and Internet learning are slowly moving the education domain out of the university.

Faculty are also feeling the impact. Academics used to get a PhD in a field, then research and teach in that field for a lifetime. Now, the rate of technology change can render a research focus irrelevant, particularly in science or engineering. Lifelong learning is essential for everyone, including faculty, but many find changes in research and teaching directions difficult to navigate.

The glacial speed of curriculum change does not work in an exponentially changing world. We cannot

continue to teach material taught 40 years ago; to do so neglects the monumental changes society, and every discipline has experienced as a result of the advent of computers, networking, and data storage, which has fundamentally changed how people work and live. It is time for higher education to move beyond the industrial age and into the information age.

The Case for Radical Change

How the educational system will adjust to these changes is not yet clear, but, as in other fields, we must find a way. Simply opting for incremental change is not enough; if the U.S. higher education system fails to respond to 21st century needs, other countries will develop new ways to educate students to make them more productive in society.

To keep pace, higher education must implement changes in two key areas.

Curriculum

Changing curriculum entails many challenges. Adding material to the curriculum requires removing other material in order to fit within the standard four-year program. At universities, removing content is a challenge, as there's rarely agreement on what to eliminate. Also, any changes to curriculum must endure an elaborate and slow bureaucratic approval process, which typically take years.

Still, change is required. To exemplify the process in one field of higher education, I documented how (and why) aerospace engineering curricula should change.[5] Half a century ago, if someone understood structures,

aerodynamics, control, and propulsion they could design and build aircraft. Today, up to 70 percent of the cost of aircraft and spacecraft and up to 80 percent of their functionality is in computers and software.

The Boeing 777 has more than a thousand onboard processors and millions of lines of software. Eighty percent of the F-22 fighter plane's functionality is achieved through software, which is why software is called the Achilles heel of aerospace systems.[6] **The F-35 has had very serious delays and cost overruns due to software problems.** This is no doubt in part because traditional aerospace engineers (and many faculty) know very little about software. Thus, we continue to have serious aerospace problems due to faulty software, and our educational system is not addressing this. Figure 2 shows the changes in technology and the lack of changes in curricula over a 40-year period in aerospace engineering. While aircraft now have up to 80 percent of their functionality provided by software, the curriculum remains essentially unchanged, and an aerospace engineering student can graduate by taking only one course in computer programming.[7] This is why if you search a job website such as indeed.com you will find 515,000 jobs that mention software and only 140 that mention aerospace engineer, and the students typically don't know this. Based on the data shown in figure 2, it might make more sense for students to get a degree in computing or software, with a minor in aerospace engineering. And it is very important for students to do a job search and curriculum review *before* they choose a major. In addition, students often have the chance to take several electives, and they can (and should) choose computing or software courses to attempt to make up

for these not being required.

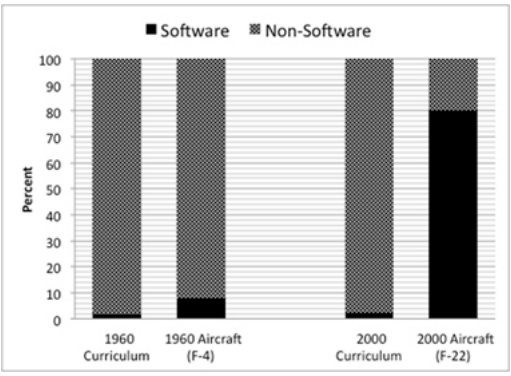

Figure 2. Software in the curriculum and aircraft over time

Faculty

A key challenge in curriculum change is that faculty members often don't know what they don't know. Most were trained before the PC, Internet, or cellphones even existed. As a result, **today's students often spend time learning very difficult material that they don't need, while getting no exposure to critical information that is important to achieving success in the digital age.**

Faculty members are in charge of the curriculum at most universities, yet many cling to their out-of-date research topics and teach the same courses for 20 years or more. Universities need to encourage (and reward) faculty for moving into new technologies and integrating 21st century computing, software, and IT into their courses. Many engineering and science programs have almost no requirements for students (undergraduate or graduate) to learn computing beyond taking a freshman course in programming. This is ridiculous and embarrassing: a course of study that should be essential to the curriculum often isn't even in it. Our students deserve better.

Penn State's Approach

To address these needed changes, **The Pennsylvania State University** has begun offering minor degrees and certificates for both undergraduate and graduate students. We created a **graduate minor program in computational science** and several different **undergraduate minors in information, science, and technology**.

All students in every major can benefit from such a minor, whether they are in liberal arts, science, engineering, medicine, or law. When these minors are created carefully, students can still graduate in four years. And, while curriculum often includes material that should *definitely* be removed, it is not necessary with this approach — making it easier to implement and easier for faculty to embrace. It's also an approach with a powerful impact on students. Indeed, several undergraduate and graduate students have told me that they received job offers because of their minors, with their majors being a secondary consideration. Students who take advantage of our new programs will be better prepared for the rapidly changing future. **In my view, every student, in every major, should complete a minor in IT, computing, programming, or software.** In addition, we should increase our enrollment in majors such as computer science, computational science, data science, software engineering, etc. by a factor of ten or more. This too is very difficult to do at slowly changing universities.

Conclusions

Many of our society's technical failures are due to poorly trained engineers and scientists who do not understand complex hardware and software systems.

Aircraft, spacecraft, power plants, automobiles, information systems, networks, and so on are all extremely vulnerable to software failures — and our workforce is not trained to handle these issues. Only about five percent of our engineering undergraduates can even program a computer. In addition, only about 55 percent of entering engineering students actually graduate. Perhaps if these students were taught more relevant, modern, and interesting material they would stay in the programs. Often, students themselves realize what's important; one school offered a computer science minor, and it was so popular that the university could not meet the demand and it was canceled. Unbelievable.

Clearly, we need a nation-wide discussion on how to evolve curricula for the benefit of our students and society. Academic curricula must reflect the fact that we are in the information age. Every person graduating from college should have a thorough understanding of computing, software, and information science. The computer is the greatest tool ever invented. College graduates — especially those in science and engineering — should know how to use it.

Notes

1. Ray Kurzweil, ***The Singularity Is Near: When Humans Transcend Biology***, Viking Press, 2006.
2. Greg Pearson and A. Thomas Young, editors, ***Technically Speaking: Why All Americans Need to Know More About Technology***, Committee on Technological Literacy, National Academy of Engineering, US National Research

Council, 2002.

3. Shirley Ann Jackson, ***Envisioning A 21st Century Science and Engineering Workforce for the United States: Tasks for University, Industry, and Government***, The National Academies Press, 2012.
4. Judith Anderson Koenig, ***Assessing 21st Century Skills: Summary of a Workshop***, The National Academies Press, 2011.
5. Lyle N. Long, "**The Critical Need for Software Engineering Education**," *CrossTalk: The Journal of Defense Software Engineering*, vol. 21, no. 1, January 2008, pp. 6–10; and Lyle N. Long, "**On the Need for Significant Reform in University Education, Especially in Aerospace Engineering**," *Proceedings IEEE Aerospace Conference*, Big Sky, Montana, March 2015, pp. 1–7.
6. Long, "On the Need for Significant Reform in University Education."
7. Ibid.

Lyle Long, DSc, is a distinguished professor of Aerospace Engineering, Computational Science, Neuroscience, and Mathematics at The Pennsylvania State University. He has a doctor of science degree from George Washington University, a master's of science degree from Stanford University, and a bachelor's of Mechanical Engineering from the University of Minnesota. In 2007–2008 he was a Moore Distinguished

Scholar at the California Institute of Technology. He received the Penn State Engineering Society Outstanding Research Award, the IEEE Computer Society Gordon Bell Prize, and a Lockheed award for excellence in research and development. He is a fellow of the American Physical Society and the American Institute of Aeronautics and Astronautics and an associate editor of *IEEE Transactions on Neural Networks and Learning Systems*. He has written more than 260 journal and conference papers.

Keep up with higher education IT trends

Sign up for free *EDUCAUSE Review* weekly e-mails to hear about new content.

SUBSCRIBE NOW

Explore

- Columns
- Editors' Picks
- Articles
- Blogs
- Multimedia
- EDUCAUSE Review Archives
- About EDUCAUSE Review

Contact

Contact our editors:
editors@educause.edu

For advertising information:
advertising@educause.edu

Stay Connected

- Find us on YouTube
- Follow us on Twitter
- Subscribe to the newsletter

EDUCAUSE

Research & Publications
Conferences & Events
Career Development
Focus Areas & Initiatives
Connect & Contribute
About EDUCAUSE