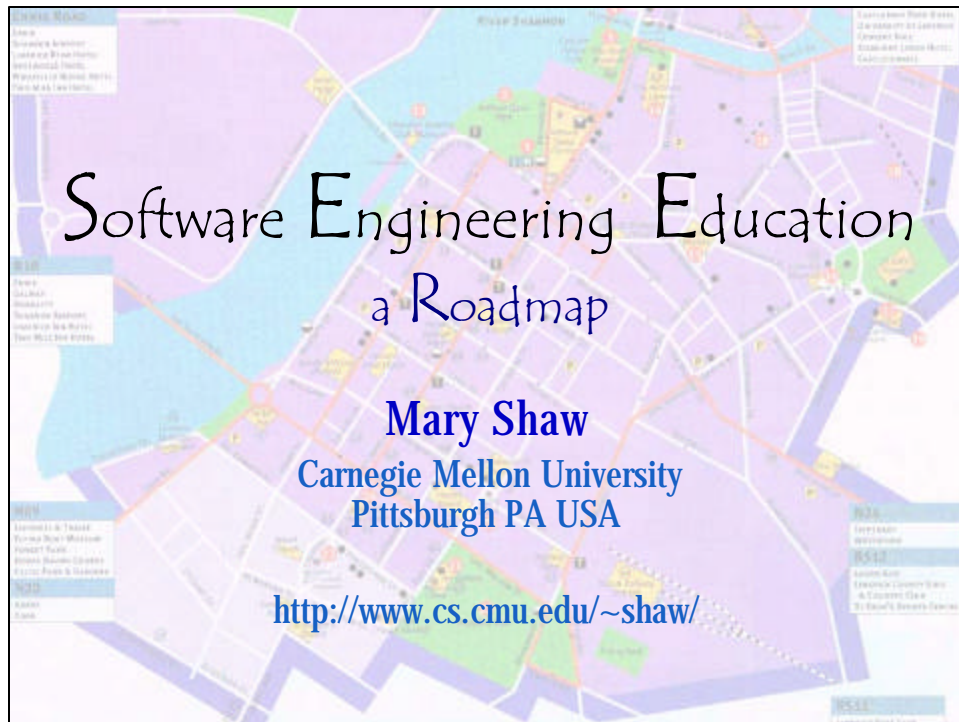


Software Engineering Education Roadmap



Overview

*Software-intensive systems have become an integral part of society
Quality depends on supply of qualified software developers*

/// **Current status:** traditional educational institutions

/// **Forces on software development:** rapid change

- ❖ evolving models for software development and use
- ❖ pressure for professional credentials
- ❖ competition for traditional institutions

/// **Challenges and aspirations:** engineering maturity

- ❖ for structure of the profession
- ❖ for educational institutions
- ❖ for currency in the face of rapid change
- ❖ for professional credentials

Institute for Software Research, International



Software Engineering Education Roadmap

Current Software Education

*Three decades of traditional education
has not produced enough qualified software developers*

⚡ Programs and institutions

- ❖ Undergraduate and graduate programs in colleges and universities
- ❖ Vocational courses in proprietary schools
- ❖ In-house and outsourced training sponsored by companies
- ❖ Personal initiative

⚡ Credentials

- ❖ Academic degrees
- ❖ Certificates of attendance
- ❖ Certificates of proficiency in specific skills

⚡ Little discrimination between engineering and other roles

Institute for Software Research, International



3

Forces on Software Development

*The prevailing **closed-shop** model of software development is
increasingly at odds with actual practice.*

- ⚡ System requirements emerge as clients understand potential
- ⚡ Complex economic and legal constraints affect systems
- ⚡ Much software is now developed by communities of cooperating volunteers
- ⚡ Software increasingly depends on autonomous resources that may be modified without notice to its users
- ⚡ Software development is increasingly disintermediated -- tailored or composed by end user, not professionals

Institute for Software Research, International



4

Software Engineering Education Roadmap

Forces on Technical Credentials

Software is of increasing public importance. The public wants and deserves assurances about how much to trust the software.

Assurances about software can come from

- ❖ direct validation of software product
- ❖ certified capability of organization that produces the software
- ❖ certified capability of person who is responsible for the software

Credentials

- ❖ much current attention to licensing software engineers
 - ❖ much actual skill certification for specific products (not engineering)
 - ❖ scant credentials for engineering in focused areas
- ♦ this is a significant opportunity

Institute for Software Research, International



5

Forces on Educational Institutions

Universities must balance enduring principles against training in current technology.

Neither extreme is appropriate, but different schools can reasonably choose different balances

Tension is increased by

- ❖ Increasingly, we shift from lecture to project courses -- the direct involvement is good, but implementation can overwhelm the principles
- ❖ Shortfall of developers is so severe that students start working before finishing degrees.
- ❖ Traditional institutions are challenged by
 - ♦ for-profit schools that teach skills that may be short-lived
 - ♦ external critics calling for accountability and efficiency
 - ♦ on-line training and education

Institute for Software Research, International



6

Software Engineering Education Roadmap

Software Engineering vs Software Developing

Definitions of “engineering” abound. They have in common:

Creating cost-effective solutions ...
... to practical problems ...
... by applying scientific knowledge ...
... building things ...
... in the service of mankind

Engineering entails making decisions under often-conflicting constraints of limited time, knowledge, and resources.

Much (most?) of software development is not engineering

Software engineering is the aspect of software development that requires technical design and decision-making and direct responsibility for success of product.

Institute for Software Research, International



7

Aspirations for the Structure of the Profession

Software development requires many skills. Despite occasional attempts to identify specific roles, the distinctions are still unclear, and “software engineer” is used loosely.

/// **Aspiration 1:** Discriminate among different software development roles.

❖ Engineering vs programming, validation, analysis, user studies, ...

/// **Aspiration 2:** Make undergraduate software education a valuable long-term investment.

❖ Undergraduate education should have a 30-year lifetime

❖ Curriculum space is a scarce resource

/// **Aspiration 3:** Provide for specialization through training and graduate education.

Institute for Software Research, International



8

Software Engineering Education Roadmap

Aspirations for Educational Institutions

*Regular complaints say CS programs don't educate engineers well.
Fixing this for SE would also help all of CS*

/// **Aspiration 4: Integrate an engineering point of view in undergraduate computer science and other information technology curricula.**

Shortcoming

Programming from scratch
Programming before reasoning
Implementing the first design
Designing for the implementer
Failing to understand scale
Writing throwaway exercises
Ignoring system and nontechnical requirements

Improvement

Study good examples
Theory and models in context of practice
Require analysis of at least two designs
Require consultation with end users
Teach back-of-the-envelope estimation
Modify systems as well as creating them
Make assignments with those requirements

Institute for Software Research, International



9

Aspirations for Dealing with Rapid Change

Software technology changes rapidly. Without sacrificing the enduring principles, we must update the content, even the structure of the curriculum

/// **Aspiration 5: Make curricula flexible & responsive to change.**

❖ How many current curricula include system-level abstractions? techniques and models for internet-based systems?

/// **Aspiration 6: Exploit our technology in support of education.**

❖ We're better than most fields about hands-on assignments, but we could make better use of our technology for presenting the material.

/// **Aspiration 7: Provide effective means for software engineers to keep their skills current.**

❖ We should establish life-long educational relations with students.

Institute for Software Research, International



10

Software Engineering Education Roadmap

Aspirations for Professional Credentials

Demand for professional credentials is high. Achievable overall practice does not yet justify engineering licensing but specific selected areas

✦ **Aspiration 8:** Establish distinct and appropriate credentials for distinct engineering roles.

❖ For example, start with safety-critical, or embedded, or real-time skills

✦ **Aspiration 9:** Establish credentials that accurately reflect achievable practice.

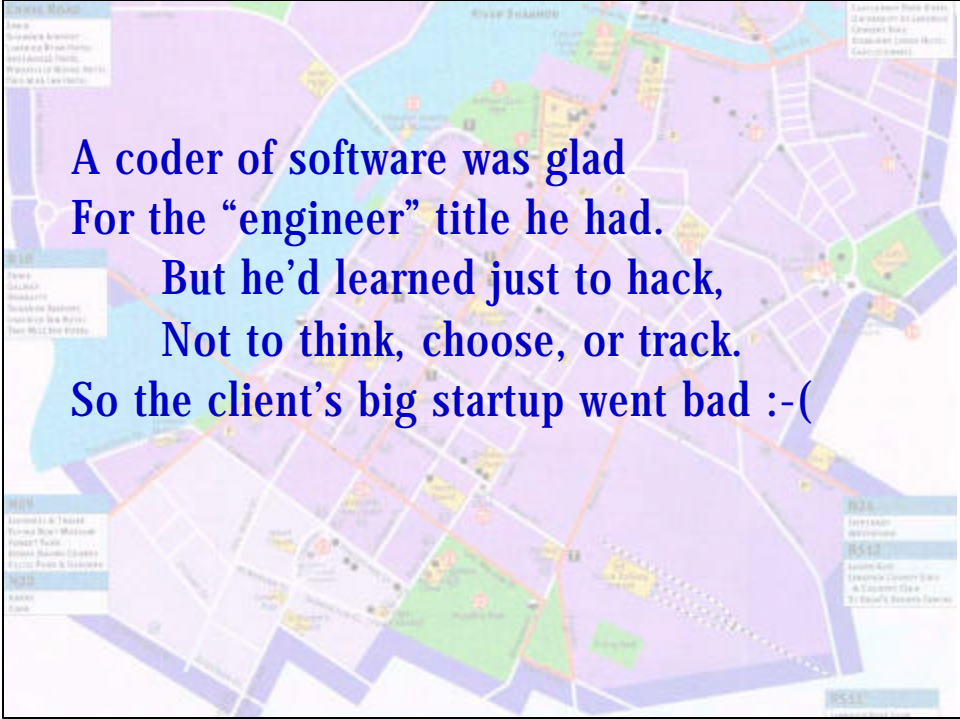
❖ Public credentialing for individuals practicing engineering requires:

- ❖ achievable level of practice that ensures quality consistent with public need
- ❖ assessment method that predicts whether an person will practice at this level
- ❖ in an evolving field, a means of ensuring update of skills

Institute for Software Research, International



11



A coder of software was glad
For the “engineer” title he had.
But he’d learned just to hack,
Not to think, choose, or track.
So the client’s big startup went bad :-)