The Education of a Software Engineer

Mehdi Jazayeri
Distributed Systems Group
Technische Universität Wien

Intended Audience

- Instructors of software engineering
- Instructors of computer science
- Students of computer science
- People in computer science
- …

What to teach a software engineering student?

- Variety of courses and textbooks
- Maturing of the discipline (?)
- Accelerating developments in technology

▪Process, abstraction, testing, requirements, module design, corba, client, provider,
▪specification, safety, quality metrics, uml 2.0, statecharts, .NET, Java, modeling
▪exceptions, cvs, bugzilla, open source, extreme programming, xml, muttation testing
▪reliability, aspects, team structure, reviews, code reading, hierarchical design, objects
▪ layered, peer to peer, security, configuration, reconfiguraton, risk, concern
▪Rigor, evolution, reengineering, GUI, message, …etc. etc. ???

Tradeoffs in software engineering education

- Theory versus practice or
  SE “as it should be” versus “as-is”
- Development or management
- Product or process
- Formal versus empirical or
  mathematics versus engineering
- Abstract concepts or concrete
  methodologies

New challenges due to technological trends

- Distribution (end of mainframes?)
- Computing platform (middleware)
- Pervasive computing (interface with the environment)
- The Internet and open source
- Proliferation of tools
- Software evolution
- Really “pervasive” computing (interdisciplinary computing)

SE Challenge: Distribution

- Of the software
  - SE+distributed systems (concurrency, caching, fault-tolerance, synchronization, …)
- Of the people
  - Outsourcing, organizational boundaries, …
- Of the process
  - 24/7 development, concurrent engineering, …

SE Challenge: Platforms

- Client-server, middleware, events …
- .NET, EJB, J2EE, CCM, …
- Portable devices, cell-phones
- Do standards matter?
- Is it all about marketing?

SE Challenge: Pervasive computing

- Scale (lots of nodes)
- Heterogeneity (sensors, kitchen appliances…)
- Dynamic configurations (ad hoc networks)
- Embedded systems (hw/sw, resource constraints)
- Interfacing to the physical environment (different interfaces, including UI)

SE Challenge: Internet

- Internet as a resource in practice (outsourcing, beta-testing)
- Internet as a resource for education (OCW)
- Internet as a delivery platform (Web informatics)
- Internet as a development platform
- Internet technologies (XML …php…asp…)
- Internet time versus quality?

SE Challenge: Software Evolution

- Challenge to industry
- Can it be separated from “SE”?
- Resists simplifications
  - Mixture of technical, organizational, social
  - Product families
  - Architectures

Proliferation of tools

- No longer just line-oriented tools
- Environments that enforce methodologies, processes, tools, …
- Learning curve?
- Commercial forces?
- Start with market-neutral environment?

SE Challenge: Open source

- New process (just like science!)
- New support tools (sourceforge, …)
- New organizational rules
- Business model?
- Advocacy or criticism?

The real SE Challenge: *Really* "pervasive" computing

- Software is critical to every field today: science, commerce, business, education, government, …
- A software engineer must be able to work with different domains
- Interdisciplinary software engineering or *-informatics
- Who are we educating?

What does a software engineer need at work?

- Knowledge of theory and techniques
- Experience with technologies and tools
- Ability to work in a team
  - Just as software binds most systems, the software engineer often binds the team
- Ability to communicate with colleagues and clients
- Experience and judgment

Non-technical skills

- Communication
  - Technical and non-technical
  - Written and oral
- Work in a team
  - With computer scientists
  - With non-computer scientists

Ingredients of a curriculum: Università Svizzera Italiana

- Theory
- Technology
- Systems approach
- Interdisciplinary applications
- Teamworking and communication
- How: project-based learning

Semester structure

Contents of software atelier

- Tools and technologies
- Professionalism (competence and ethics)
- Problem solving and project management
- Communication skills and team work
- Integrative projects
- Interdisciplinary projects
- The real world (as it should be!)

Levels of professionalism

- Self (basic engineering skills)
- Team (interpersonal and cooperation skills)
- Society (greater responsibilities)

Software Atelier: support

- New building architecture to encourage teamwork
  - Faculty and students in close proximity
  - Laptops as primary computing platform for students
  - Labs organized as modular group areas
- Lectures on project management, problem solving, technical documentation, …

Project sequences

- Basic tools: productivity, unix, html, configuration management, bug reporting, tex
- Visual environments, Matlab, …
- Req., Spec., testing tools. Robot programming
- Web-based, DB, GUI, scripting
- Network-oriented programming, COTS, application servers
- Business plan (financing, scheduling, marketing)

Interdisciplinary studies

- Mathematical models
- Life sciences models
- Economics and business models
- Modeling techniques

A digression: SE for others

- SE principles are basic skills for users too
- SE essentials for non-computer scientists
  - Dealing with complexity
  - Abstraction and modeling
  - Problem solving (decomposition and modularity)
  - Quality assurance

Conclusions?

- Good software engineering skills are necessary for all computer scientists
- Software engineering education must combine theory, practices, and application experience
- Software engineering education must be woven into the computer science curriculum
- On-the-job training is also necessary

Further reading

Computing curricula 2001, Computer Science, http://www.computer.org/education/cc2001/final/

Computing curricula 2003, Software Engineering, http://sites.computer.org/ccse/

- SE Education Track: Global Software Engineering
- Organized by Paola Inverardi and Mehdi Jazayeri


USI
Università
della Svizzera
Italiana
www.unisi.ch

USI in cifre – 2003-04

1600 Studenti di 30 nazionalità diverse

200 Studenti post-graduate

150 Docenti

4 Facoltà

15 Istituti di ricerca

3 Scuole dottorali

7 Executive Master

Facoltà

Architettura
- progettazione continua
- con insegnamenti culturali, umanistici e tecnici

Scienze della comunicazione
- comunicazione di massa e nuovi media
- comunicazione d'impresa
- comunicazione istituzionale
- comunicazione nei contesti di formazione
- tecnologie della comunicazione

Scienze economiche
- profilo economico
- profilo aziendale
- profilo finanziario

Scienze informatiche
- programma interdisciplinare in collaborazione con i Politecnici e la SUPSI, specializzazioni nuove, nuovi metodi didattici, sinergie con le facoltà esistenti