SE 2A04 Fall 2000

01. Software Engineering

Instructor: W. M. Farmer

Revised: 20 September 2000

What is Software Engineering?

- **Software engineering** is the branch of engineering concerned with the development of software products that are well-understood, reliable, maintainable, and cost effective.
- Software engineering is essential in the development of software products which are:
  - Large in size requiring multiple developers and multiple versions
  - Safety, security, or cost critical
  - Part of a physical product

How is Software Engineering Different?

- Software products are composed of and manipulate **digital objects** (instead of **physical objects**)
  - Low tolerance for error
  - Visualization is often difficult
- Today software development is more an art than a science
  - No established "best practice"
  - Quality is difficult to measure
  - The development process is usually not rationally based
- The field of computing is constantly changing
  - Cost and time constraints are often severe

What is Software Engineering Today?

- Software engineering is not well defined nor well established
- The field has largely developed outside of the engineering community
  - The early leaders in the field were mathematicians and scientists, not engineers
  - Many engineers are still unaware of the importance of computers in their profession
  - Most university software engineering courses are taught by nonengineers in computer science departments
- Because badly designed software projects are hard to manage, emphasis has been on project management rather than software design

What Should Software Engineering Be?

- Should be a professional program in engineering
  - Incompetently designed software can be dangerous to the public
  - People who need and work with software are often unable to judge the competence of software designers
  - Software developers need the independence (from their employers) to follow professional obligations
- Should be regulated by a professional engineering society
  - Software engineering programs should be accredited
  - Software engineers should be professionally licensed
  - Software products should be approved by professional software engineers

Professional Responsibilities

A software engineer should have the same professional responsibilities as other engineers:

- Accept individual responsibility
- Solve the real problem
- Be honest about capabilities
- Produce reviewable designs
- Maintainability

See D. Parnas, "Professional responsibilities of software engineers", in: K. Brunnstein and E. Raubold, eds, *13th World Computer Congress 94*, Vol. 2, North-Holland, 1994.

The "Software Crisis"

- In contrast to hardware, software costs tend to increase over time
- Software is normally unreliable
- Software often ages prematurely
- Humanity is increasingly dependent on software
  - Software is a component of most critical systems
  - More and more products have software components

Software Reliability

- Reliable software is unusual and good scientific evidence of reliability is even more unusual
  - Software does not usually come with a warranty
  - Even safety, security, and cost critical software are often unreliable
- Reliability is difficult to guarantee
  - Software systems are often very large and very complex
  - The behavior of supporting hardware, operating systems, programming languages, software tools, etc. is usually not well understood
- Reliability is difficult to determine
  - What is the software required to do?
  - What does the software do?

Software Aging

- Usefulness of software diminishes over time
  - Maintenance tends to become increasingly difficult and costly
  - Range of applicability tends to decrease
- Two types:
  - Failure to keep up with a changing environment
  - Damage due to maintenance
- Causes:
  - Software is not well understood (developers are gone, documentation is poor, system is overgrown)
  - Software is not designed for change

See D. Parnas, "Software Aging", 1995.

A Prediction for the Future

Sometime in the next couple decades software production will undergo a precipitous transformation

- Software products will be well designed, understood, and documented
- Warranties on software will be standard
- Critical software will be extremely reliable
- Software development will be largely performed by professional software engineers