Intro to SE

LECTURE 1

Tentative Grade Distribution

Quizzes (Announced + Unannounced) – 10 %

Assignments + Project + Class Exercises – 20 %

Mid Term Exams – 30 %

Final Exam – 40 %

Grading Criterion: Absolute

Passing Criteria

- Minimum Attendance Requirements Met
- Minimum Marks Requirements Met

Text Book

Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli, Fundamentals of Software Engineering, 2nd Edition. Pearson

Dick Hamlet, Joe Maybee, The Engineering of Software: Technical Foundations for the Individuals. Addison Wesley

Roger S. Pressman, Software Engineering A Practitioner’s Approach, 8th Edition. McGrawHill

What is Software?

The product that software professionals build and then support over the long term.

Software encompasses: (1) instructions (computer programs) that when executed provide desired features, function, and performance; (2) data structures that enable the programs to organize, adequately store and manipulate data and information and (3) documentation that describes the operation and use of the programs.

Software products

Generic products

- Stand-alone systems that are marketed and sold to **any customer** who wishes to buy them.
- Examples – PC software such as editing, graphics programs, project management tools; CAD software; software for specific markets such as appointments systems for dentists.

Customized products

- Software that is commissioned (custom-build) by **a specific customer** to meet their own needs.
- Examples –air traffic control software, traffic monitoring systems.

Why Software is Important?

The economies of ALL developed nations are dependent on software.

More and more systems are software controlled (transportation, medical, telecommunications, military, industrial, entertainment,)

Software engineering is concerned with theories, methods and tools for professional software development.

Software costs

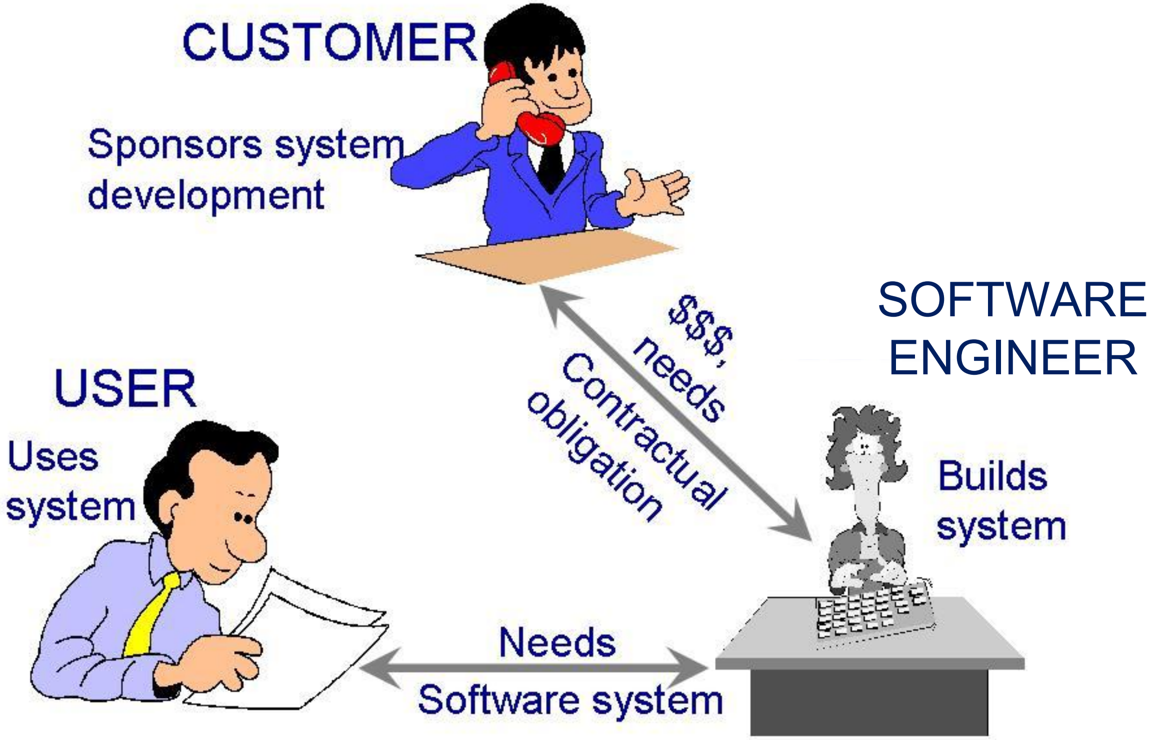
Software costs often dominate computer system costs. The costs of software on a PC are often greater than the hardware cost.

Software costs **more to maintain** than it does to develop. For systems with a long life, maintenance costs may be several times development costs.

Software engineering is concerned with cost-effective software development.

Who Does Software Engineering?

Participants (stakeholders) in a software development project



Engineering?

Engineers' Job?

- Make things work
- Provide solutions in absence of applicable theories and methods
- Realize financial and organizational constraints

Software Engineers?

Adopt a systematic and organized approach, effectively, to produce high quality software

May have to use Ad hoc approaches to develop software

- Some real complex problems may not be solved using elegant theories of CS
- new algorithms?

What is a Good Software Product?

Good software engineering must always include a strategy for producing quality software

Product Quality?

- Multiple facets…

What is a Good Software Product?

Users judge external characteristics (e.g., correct functionality, number of failures, type of failures)

Designers and maintainers judge internal characteristics (e.g., ease of modification)

Thus different stakeholders may have different criteria

Need quality models to relate the user’s external view to developer’s internal view

Aspects of Software Production?

Size of the software system to be built

Complexity of the software system to be built

Need and involvement of teams

Technical process of developing software

Activities such as management of project and teams

Development of tools, theories, methods to support production of software

Software Application Domains

System software. A collection of programs written to service other programs e.g., compilers, editors, file management utilities, operating system components, drivers, networking software, telecommunications processors

Application software. Stand-alone programs that solve a specific business need. Applications in this area process business or technical data in a way that facilitates business operations or management/technical decision making.

Software Application Domains

Engineering/scientific software. A broad array of "number-crunching" or data science programs that range from astronomy to volcanology, from automotive stress analysis to orbital dynamics, from computer-aided design to consumer spending habits, and from genetic analysis to meteorology.

Embedded software. Resides within a product or system and is used to implement and control features and functions for the end user and for the system itself.

Embedded software can perform limited and esoteric functions (e.g., key pad control for a microwave oven)

Fuel control

Dashboard display

Software Application Domains

Product-line software. Composed of reusable components and designed to provide specific capabilities for use by many different customers. It may focus

on a limited and esoteric marketplace (e.g., inventory control products) or attempt to address the mass consumer market.

Web/mobile applications. This network-centric software category spans a wide array of applications and encompasses browser-based apps, cloud computing, service-based computing, and software that resides on mobile devices.

Artificial intelligence software. Makes use of heuristics to solve complex problems that are not amenable to regular computation or straightforward analysis. Applications within this area include robotics, decision-making systems, pattern rec ognition (image and voice), machine learning, theorem proving, and game playing.