CSCI 4050/6050
Software Engineering

Introduction

Dr. Eman Saleh eman.saleh@uga.edu

Software Engineering

- The term Software Engineering was introduced in the late 60s, during a conference held to discuss the "software crisis"
- Software crisis has not been resolved, yet

Goals of Software Engineering

Produce what is sometimes called "Quality Software", or "Well-Engineered Software"

This is not Fireworks!

In 1996, Arian-5 space rocket, developed at the cost of $7000 million over a period of 10 years was destroyed within less than a minute after its launch. The crash occurred because there was a software bug in the rocket guidance system.

https://www.youtube.com/watch?v=gp_D8r-2hwk

- The Northeast blackout in 2003 has been one of the major power system failures in the history of North America. This blackout involved failure of 100 power plants due to which almost 50 million customers faced power loss that resulted in financial loss of approximately $6 billion. Later, it was determined that the major reason behind the failure was a software bug in the power monitoring and management system.

- Year 2000 (Y2K) problem refers to the widespread snags in processing dates after the year 2000. The roots ofY2K problem can be traced back to 1960-80 when developers shortened the 4-digit date format like 1972 to a 2-digit format like 72 because of limited memory. At that time they did not realize that year 2000 will be shortened to 00 which is less than 72. In the 1990s, experts began to realize this major shortcoming in the computer application and then millions were spent to handle this problem.

In 1996, one of the largest banks of US credited accounts of nearly 800 customers with approximately $9241acs. Later, it was detected that the problem occurred due to a programming bug in the banking software.

During the Gulf War in 1991, the United States of America used Patriot missiles as a defense against Iraqi Scud missiles. However, the Patriot failed to hit the Scud many times. As a result, 28 US soldiers were killed in Dhahran, Saudi Arabia. An inquiry into the incident concluded that a small bug had resulted in the miscalculation of missile path.

Software Engineering

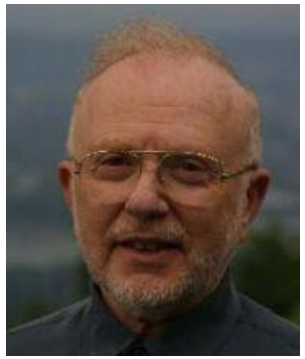
Characteristics of "Quality Software", or "Well-Engineered Software":

- performs precisely as required, under all circumstances
- is entirely fault-free (reliable)
- is maintainable (easy to modify/evolve)
- offers appropriate user interface
- is efficient
- is delivered on time
- is delivered within budget

Why is software development difficult

- The problem domain (also called application domain) is difficult
- The solution domain is difficult
- The development process is difficult to manage
- Software offers extreme flexibility
- Software is a discrete system
  - Continuous systems have no hidden surprises
  - Discrete systems can have hidden surprises! (Parnas)

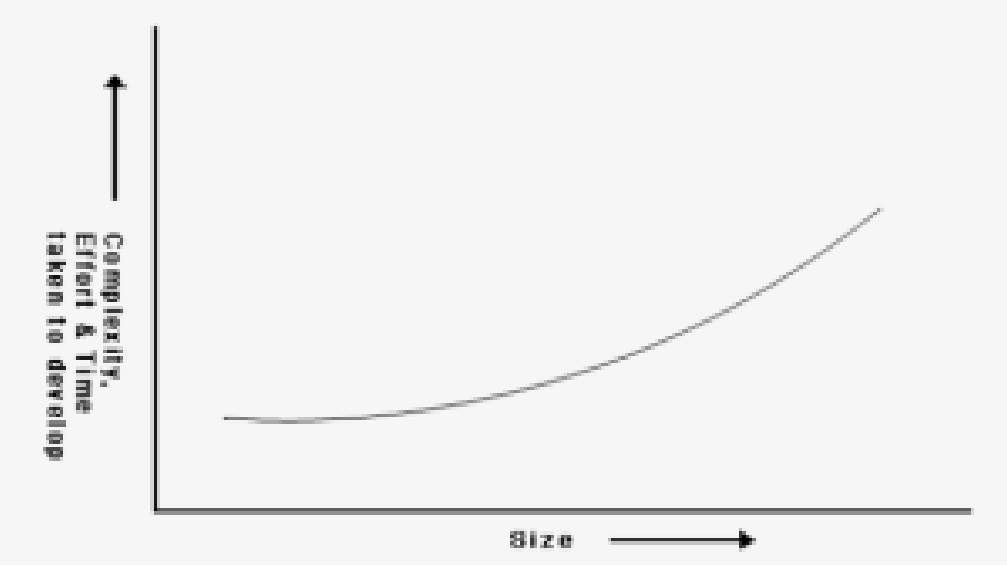
David Parnas is an early pioneer in software engineering who developed the concepts of **modularity** and **information hiding** in systems, which are the foundations of Object-Oriented methodologies.



SE is more than writing code

False assumption:

"Anybody who has programmed computers can write a small Java program in a few minutes (perhaps hours). Writing a large program will surely take longer, so let's get down to writing code ASAP, not wasting time on some theories."



SE is more than writing code

Some perspective on the "size of the problem"

Lines of code	Comments
300	No problem ☺☺☺

SE is more than writing code

Some perspective on the "size of the problem"

Lines of code	Comments
300	No problem ☺☺☺
3,000	More difficult, but still easy ☺☺

SE is more than writing code

Some perspective on the "size of the problem"

Lines of code	Comments
300	No problem ☺☺☺
3,000	More difficult, but still easy ☺☺
30,000	Still more difficult, but still possible ☺

SE is more than writing code

Some perspective on the "size of the problem"

Lines of code	Comments
300	No problem ☺☺☺
3,000	More difficult, but still easy ☺☺
30,000	Still more difficult, but still possible ☺
300,000	Usually, not for a single person ☹

SE is more than writing code

Some perspective on the "size of the problem"

Lines of code	Comments
300	No problem ☺☺☺
3,000	More difficult, but still easy ☺☺
30,000	Still more difficult, but still possible ☺
300,000	Usually, not for a single person ☹
3,000,000	Really complex ☹ ☹ Example: Space Shuttle software

SE is more than writing code

Facets of Software Engineering

- Problem solving
  - Creating a solution
  - Engineering a system based on the solution
- Modeling
- Knowledge acquisition
  - Knowledge about solution domain
- Rationale management
  - Capture context in which decisions are made
  - Understand the forces involved and the rationale behind which these decisions were made.

Software engineering: Definition (1)

- According to **IEEE**, software engineering is defined as 'the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.’

- In anutshell, software engineering can be defined as a systematic approach to develop software within specified time and budget.

Software engineering: Definition (2)

- Software engineering is an engineering discipline that is concerned with all aspects of software production from the early stages of system specification through to maintaining the system after it has gone into use.
- Engineering discipline
  - Using appropriate theories and methods to solve problems bearing in mind organizational and financial constraints.
- All aspects of software production
  - Not just technical process of development. Also project management and the development of tools, methods etc. to support software production.

SE: A Working Definition

Software Engineering is a collection of techniques, methodologies and tools that help with the production of

A high quality software system developed

- within a given *budget*,
- meeting a given *deadline*,
- while *change* occurs.

Challenge: Dealing with complexity and change

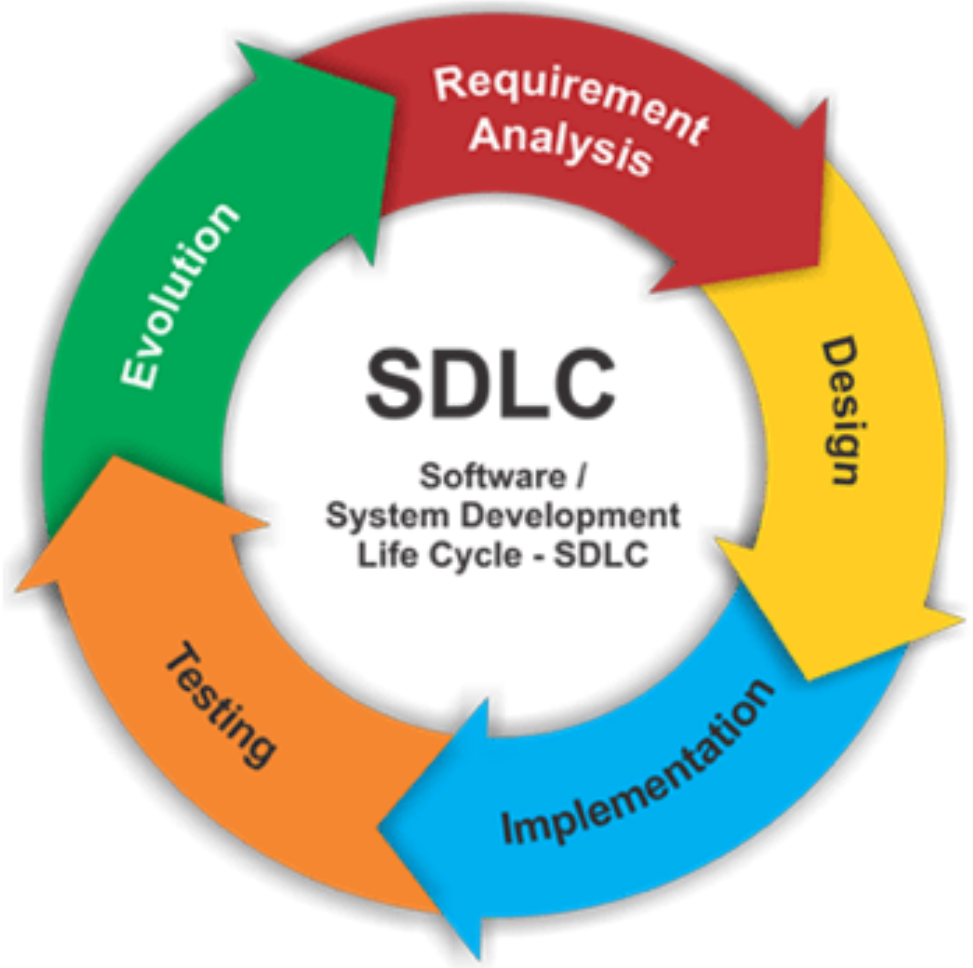
Techniques, Methodologies, Tools

- Techniques:
  - Formal procedures for producing results using some well-defined notation
- Methodologies:
  - Collection of techniques applied across software development and unified by a philosophical approach
- Tools:
  - Instruments or automated systems to accomplish a technique
  - CASE = Computer Aided Software Engineering

Example of quality attributes

- Reliability
- Modifiability
- Maintainability
- Understandability
- Adaptability
- Reusability
- Efficiency
- Portability
- Traceability of requirements
- Fault tolerance
- Backward-compatibility
- Cost-effectiveness
- Robustness
- High-performance
- Good documentation
- Well-defined interfaces
- User-friendliness
- Reuse of components
- Rapid development
- Minimum number of errors
- Readability
- Ease of learning
- Ease of remembering
- Ease of use
- Increased productivity
- Low-cost
- Flexibility

Software Engineering Life Cycle (Phases/ Activities)



Software Development Process

Many different software processes exist, but all involve:

- Specification – defining what the system should do
- Design and implementation – defining the organization of the system and implementing the system
- Validation – checking that it does what the customer wants
- Evolution – changing the system in response to changing customer needs

Computer Science vs. SE

- Computer Scientist
  - Assumes techniques and tools have to be developed
  - Proves theorems about algorithms, designs languages, defines knowledge representation schemes
  - Has infinite time…
- Engineer
  - Develops a solution for a problem formulated by a client
  - Uses computers & languages, techniques and tools
- Software Engineer
  - Works in multiple application domains
  - Has only 3 months...
  - …while changes occurs in the problem formulation (requirements) and also in the available technology

SE: A Problem-Solving Activity

- Analysis:
  - Understand the nature of the problem and break the problem into pieces
- Synthesis:
  - Put the pieces together into a large structure

For problem solving we use techniques, methodologies and tools.

Essential attributes of good software

Product characteristic	Description
Maintainability	Software should be written in such a way so that it can evolve to meet the changing needs of customers. This is a critical attribute because software change is an inevitable requirement of a changing business environment.
Dependability and security	Software dependability includes a range of characteristics including reliability, security and safety. Dependable software should not cause physical or economic damage in the event of system failure. Malicious users should not be able to access or damage the system.
Efficiency	Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, memory utilisation, etc.
Acceptability	Software must be acceptable to the type of users for which it is designed. This means that it must be understandable, usable and compatible with other systems that they use.

Frequently asked questions about software engineering

Question	Answer
What is software?	Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.
What are the attributes of good software?	Good software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable.
What is software engineering?	Software engineering is an engineering discipline that is concerned with all aspects of software production.
What are the fundamental software engineering activities?	Software specification, software development, software validation and software evolution.
What is the difference between software engineering and computer science?	Computer science focuses on theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.
What is the difference between software engineering and system engineering?	System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering. Software engineering is part of this more general process.

Frequently asked questions about software engineering

Question	Answer
What are the key challenges facing software engineering?	Coping with increasing diversity, demands for reduced delivery times and developing trustworthy software.
What are the costs of software engineering?	Roughly 60% of software costs are development costs, 40% are testing costs. For custom software, evolution costs often exceed development costs.
What are the best software engineering techniques and methods?	While all software projects have to be professionally managed and developed, different techniques are appropriate for different types of system. For example, games should always be developed using a series of prototypes whereas safety critical control systems require a complete and analyzable specification to be developed. You can't, therefore, say that one method is better than another.
What differences has the web made to software engineering?	The web has led to the availability of software services and the possibility of developing highly distributed service-based systems. Web-based systems development has led to important advances in programming languages and software reuse.

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 graphics programs, project management tools; CAD software; software for specific markets such as appointments systems for dentists.
- **Customized products**
  - Software that is commissioned by a specific customer to meet their own needs.
  - Examples – embedded control systems, air traffic control software, traffic monitoring systems.

Product specification

- **Generic products**
  - The specification of what the software should do is owned by the software developer and decisions on software change are made by the developer.
- **Customized products**
  - The specification of what the software should do is owned by the customer for the software and they make decisions on software changes that are required.

- Next Lecture: We will discuss the different application types