Introduction to Software Engineering

- Softwares
- Importance of SWE
- Basic SWE Concepts

What is a Software?

- Software is a computer program with its documentation such as requirements, design models and user manuals.
- Software products may be
  - Generic - developed to be sold to a range of different customers e.g. general PC software such as Excel or Word.
  - Bespoke (custom, tailored) - developed for a single customer according to their specification.

Attributes of Good Software

- A good software should
  - deliver the required functionality and performance to the user
  - be maintainable: can be evolved to meet changing needs;
  - be dependable: reliable and trustworthy
  - be efficient : should not waste the system resources
  - be acceptable by end-user, i.e., usable, understandable and compatible with other systems.

Software Crisis

The notion of software engineering was first proposed in 1968 at a conference to discuss what was then called ‘software crisis’:

- Informal (ad-hoc) software development
- Major projects were sometimes years late and over budget,
- Softwares were unreliable, difficult to maintain and performed poorly.

Some Facts

- Failed software projects in USA costs $81 bn annually.
- Failure is the exception not the rule, but it can be severe and expensive.
- Problems may happen during any stage of the Software development life-cycle.
- Purpose of SWE is to avoid problems and hence failure

Importance of SWE

- The economies of ALL developed nations depends (somehow) on software.
- More and more systems are software controlled
- Software cost often dominate computer system costs. The costs of software on a PC are often greater than the hardware cost.
- SWE develop theories, methods and tools that help to build cost-effective and high-quality software.

What is Software Engineering?

- SWE is an engineering discipline that is concerned with all aspects of software production.
- Software engineers should
  - adopt a systematic and organised approach to their work,
  - use appropriate tools and techniques

  depending on the problem to be solved (the development constraints + the resources available.)

The Big Picture

- Computer Science
- Computer Engineering
- System Engineering
- Software Engineering

- Science is Theory
- Engineering is more practical
- Software $\subseteq$ Computer $\subseteq$ System

SWE vs. Computer Science

- Computer science is concerned with theory and fundamentals
- SWE is concerned with the practicalities of developing and delivering useful software.
- Computer science theories are still insufficient to act as a complete foundation for software engineering.

SWE vs. System Engineering

- System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering.
- SWE is part of the process concerned with developing the software infrastructure, control, applications and databases in the system.
- System engineers are involved in system specification, architectural design, integration and deployment.

What is a Software Process (SP)?

- SP is a set of activities whose goal is the development or evolution of software.
- General activities in all SPs are:
  - Specification: what should the system do and what are its development constraints?
  - Development: production of the software
  - Validation: checking that the software is what the customer wants
  - Evolution: changing the software in response to changing demands.

What is a Software Process Model (SPM)?

- SPM is a simplified representation of a software process, presented from a specific perspective such as
  - Workflow perspective: sequence of activities;
  - Data-flow perspective: information flow;
  - Role/action perspective: who does what.
- Examples of generic SPM
  - Waterfall
  - Iterative development
  - Component-based software engineering

What is CASE?

- CASE (Computer-Aided Software Engineering) are software systems that are intended to provide automated support for software process activities.
- CASE are often used for method support.
  - Upper-CASE: support the early process activities of requirements and design;
  - Lower-CASE: support later activities such as programming, debugging and testing.

Key Challenges Facing Software Engineering

- Heterogeneity: developing techniques for building software that can cope with heterogeneous (different) platforms and execution environments
- Delivery: developing techniques that lead to faster delivery of software
- Trust: developing techniques that demonstrate that software can be trusted by its users.

Summary

- SWE is an engineering discipline that is concerned with all aspects of software production.
- Software products consist of developed programs and associated documentation. Essential product attributes are maintainability, dependability, efficiency and acceptability.
- SP consists of activities that are involved in developing softwares. Basic activities are software specification, development, validation and evolution.
- CASE tools are software systems which are designed to support routine activities in the SP