CS342 Software Engineering

Dr. Ismail Hababeh

German-Jordanian University

Lecture 1

Course Objectives

- Extract Software project requirements
- Create the Software Specification Document
- Model the classical system analysis
- Model the dynamic systems analysis
- Build the classical System Design
- Build the Object-Oriented Design
- Build the software Implementation phase
- Build the software testing model

Course Objectives

- Measure the efficiency of the project functionality
- Build the Business Model

Course Outline

- The software process
- Software life-cycle models
- The workflows of the software life cycle:
    - Requirements
    - Classical analysis
    - Object-oriented analysis
    - Design
    - Implementation
    - Testing
    - Maintenance

Course Outline (cont)

- Software engineering tools
- Unified modeling language
- Software testing
- Reusability, portability, and interoperability (*ability of computer systems or software to exchange and make use of information*).
- Planning and estimation

Textbooks

1. Software engineering 9th Edition.
   Sommerville, Ian.
   ISBN-10 137035152 (2011).
2. Object-oriented and classical software engineering.
   Schach, S.R.
   New York. McGraw-Hill.

Software Development Life Cycle

Series of software development steps, from concept exploration through final retirement:

1. **Requirements phase** (concept explored, includes rapid prototyping)
2. **Specification/Analysis phase** (contract)
3. **Design phase**
   a) high-level (architectural design => modules)
   b) detailed (design of each module)
4. **Implementation phase** (coding)
5. **Testing phase** (coding/testing)
   a) Unit testing
   b) Integration of sub-systems
6. **Maintenance phase** (any changes after acceptance)
7. **Retirement**

Need of Software Development Life Cycle

- Allows a systematic and well-organized way to develop a software.
  - Team members know to do what and when
- Allows development of large software projects.
- Allows smooth interfacing between different development sectors:
  - Helps in identifying inconsistencies, redundancies, and omissions in the development process.
  - Helps in modifying a process model for specific projects.

Need of Software Development Life Cycle

- A life cycle model defines entry and exit criteria for every phase
- A software development phase begins only if its phase-entry criteria have been satisfied.
- A phase is completed when all its exit criteria are satisfied.
- Example: The phase exit criteria for the software requirements specification phase:
  - Document is complete
  - Reviewed
  - Approved by the customer

Software Maintenance

Software maintenance:

- Good software is maintained where bad software is discarded.
- Different types of maintenance:
  - Corrective maintenance [about 20%]
  - Enhancement maintenance [about 80%]
    - Adaptive maintenance [about 20%]

      changes made in response to changes in the environment where the product operates (ex: new government regulations)
    - Perfective maintenance [about 60%]

      changes that the clients think will improve the effectiveness of the product.

Scope of Software Engineering

Economic Aspects:

- Let us assume a new software system is 10% faster than currently used system.
- Should we use it instead of the current system?

Answers:

- Ordinary: Yes
- From Software Engineering point view and before saying yes:
  - Consider the cost of training
  - Consider the impact of introducing a new technology
  - Consider the effect of new coding method on maintenance