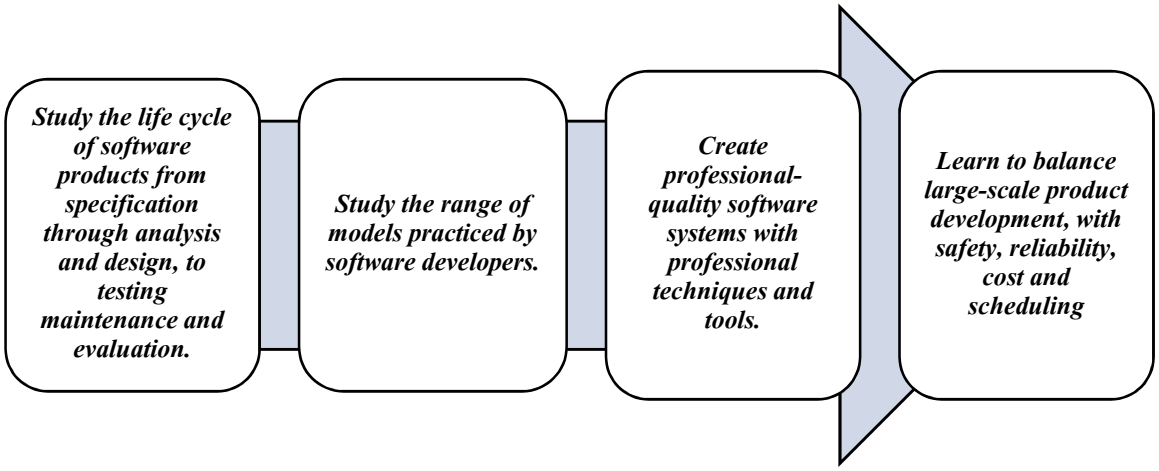
Software Engineering

:Lecture 1

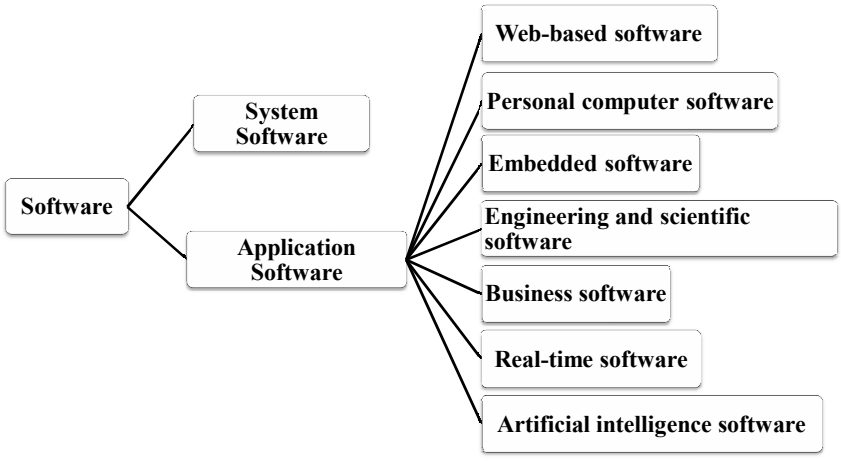
List Of Reference Material:

1. Roger S. Pressman, "**Software Engineering - A Parctitioner's Approach"**, Seventh Edition, Mc Graw-Hill International Edition, 2010.
2. Ian Sommerville, "**Software Engineering**", 9th Edition, Pearson Education Asia, 2011.
3. Rajib Mall, "**Fundamentals of Software Engineering**", Third Edition, PH1 Learning Private Limited, 2009.

Course Objectives



Categories Of Software



UNIVERSITY OF DIYALA

Categories Of Software

- ***Computer Software**: is the product that software professionals build and then support over the long term.*

1. ***System software:** is the software that acts as tools to help construct or support applications software. **Example: Compilers, Operating Systems, Databases and Networking Software.***
2. ***Applications software:** is software that helps perform some directly useful or enjoyable task. **Example : Games Software, ATMs and Airplane Software.***

Software Applications

1. ***Real-time software:** Software that monitors/ analyzes /controls real world events as they occur is called real time.*
2. ***Business software:** Business information processing is the largest single software application area.*
3. ***Engineering and scientific software:** modern applications within the engineering/scientific area are moving away from conventional numerical algorithms.*

Software Applications

4. ***Embedded software:*** *Intelligent products have become commonplace in nearly every consumer and industrial market e.g., keypad control for a microwave oven.*
5. ***Personal computer software:*** *Such as(Word processing, spreadsheets)*
6. ***Web-based software:*** *The Web pages retrieved by a browser are software that incorporates executable instructions*
7. ***Artificial intelligence software:*** *It makes use of non-numerical algorithms to solve complex problems that are not amenable to computation or straightforward analysis.*

Attributes of Good Software

- *Software products have a number of attributes which reflect the* ***quality of that software****.*

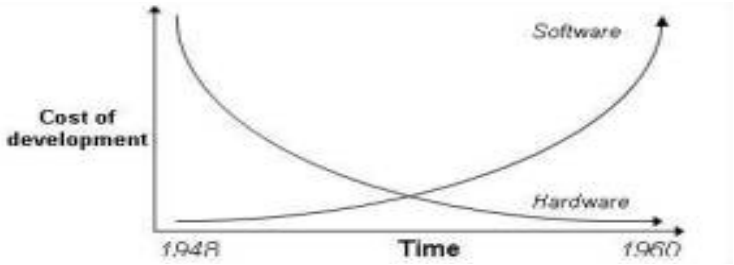
1. ***Maintainability:*** *Software should be written in such a way so that it can evolve to meet the changing needs of customers.*
2. ***Dependability:*** *software should not cause physical or economic damage in the event of system failure.*
3. ***Security:*** *Malicious users should not be able to access or damage the system.*

Attributes of Good Software

4. ***Efficiency:*** *Software should not make wasteful use of system resources such as memory and processor cycles.*
5. ***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.*

Software Crisis

- *Software engineering was first proposed in 1968 at a conference held to discuss what was then called the '**software crisis**'. In 1969 became clear that individual approaches to program development did not scale up to large and complex software systems. These were unreliable, cost more than expected, and were delivered late.*



Software Crisis

- ***The crisis encompasses problems associated with.***

1. *Increasing need for high reliability software.*
2. *Software is difficult to maintain "aging software".*
3. *Projects running over-budget and over-time.*
4. *Software was very inefficient with low quality.*
5. *Software often did not meet requirements.*
6. *Projects were unmanageable and code difficult to maintain.*
7. *Software was never delivered.*

Software Engineering Definitions

- *To define the software engineering term first we should understand what software engineering stands for. The term is made of two words,* ***software*** *and* ***engineering****.*

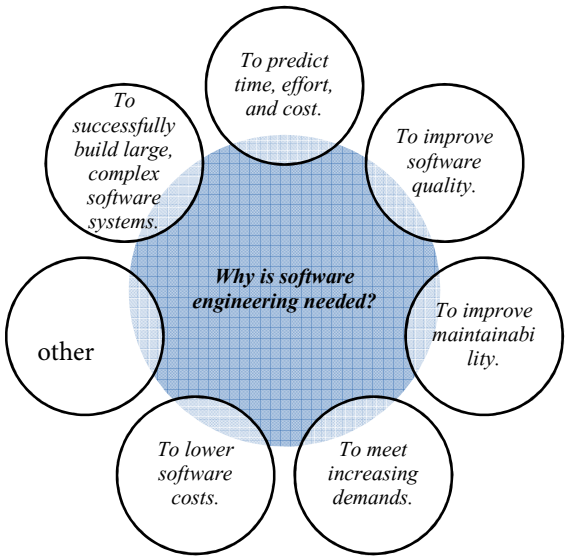
1. ***Software*** *is more than just a program code. A program is an executable code, which serves some computational purpose. Software is considered to be collection of executable programming code, associated libraries and documentations. Software, when made for a specific requirement is called software product.*
2. ***Engineering*** *on the other hand, is all about developing products, using well-defined, scientific principles and methods.*

Software Engineering Definitions

- ***Software engineering*** *is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and work efficiently on real machines.*
- ***Software engineering is important for two reasons:***

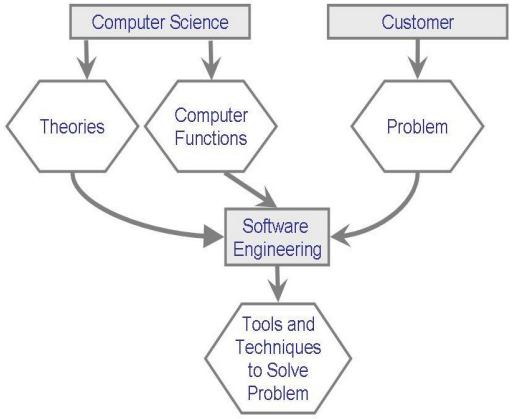
1. *More and more, individuals and society rely on advanced software systems. We need to be able to produce reliable and trustworthy systems economically and quickly.*
2. *It is usually cheaper, in the long run, to use software engineering methods and techniques for software systems rather than just write the programs as if it was a personal programming project. For most types of systems, the majority of costs are the costs of changing the software after it has gone into use.*

- .



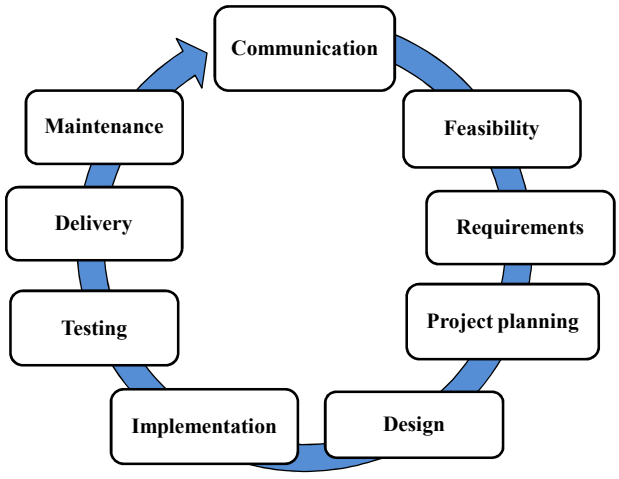
Computer Science VS Software Engineering

- ***Computer science:*** *Focusing on computer hardware, compilers, operating systems, and programming languages.*
- ***Software Engineering:*** *a discipline that uses computer and software technologies as a problem-solving tools.*



15

Software Development Life Cycle, SDLC



Software Development Life Cycle, SDLC

- *The software life cycle is the sequence of different activities that take place during software development. These activities are describe as follow:*

1. ***Communication:*** *Before any technical work can commence, it is critically important to communicate and collaborate with the customer. The goal is to understand customer objectives for the project and to gather requirements that help define software features and functions.*
2. ***Feasibility:*** *Determining if the proposed development is worthwhile.*

   Market analysis: *Determining if there is a potential market for this product.*

Software Development Life Cycle, SDLC

3. ***Requirements:*** *Determining what functionality the software should contain.*
   - ***Requirement elicitation:*** *Obtaining the requirements from the user.*
   - ***Domain analysis:*** *Determining what tasks and structures are common to this problem.*
4. ***Project planning:*** *Determining how to develop the software.*
   - ***Cost analysis:*** *Determining cost estimates.*
   - ***Scheduling:*** *Building a schedule for the development.*
   - ***Software quality assurance:*** *Determining activities that will help ensure quality of the product.*
   - ***Work-breakdown structure:*** *Determining the subtasks necessary to develop the product.*

Software Development Life Cycle, SDLC

5. ***Design:*** *Determining how the software should provide the functionality.*
   - ***Architectural design:*** *Designing the structure of the system.*
   - ***Interface design:*** *Specifying the interfaces between the parts of the system.*
   - ***Detailed design:*** *Designing the algorithms for the individual parts.*
6. ***Implementation:*** *Building the software.*

Software Development Life Cycle, SDLC

7. ***Testing:*** *Executing the software with data to help ensure that the software works correctly.*
   - ***Unit testing:*** *Testing by the original developer.*
   - ***Integration testing:*** *Testing during the integration of the software.*
   - ***System testing:*** *Testing the software in an environment that matches the operational environment.*
   - ***Acceptance testing:*** *Testing to satisfy the purchaser.*
   - ***Regression testing:*** *Saving tests from the previous version to ensure that the new version retains the previous capabilities.*

Software Development Life Cycle, SDLC

8. ***Delivery:*** *Providing the customer with an effective software solution.*
   - ***Installation:*** *Making the software available at the customer's operational site.*
   - ***Training:*** *Teaching the users to use the software.*
   - ***Help desk:*** *Answering questions of the user.*
9. ***Maintenance:*** *Updating and improving the software to ensure continued usefulness.*

editt by Sir. Ali Alani