Software Engineering

Foundations

Concepts

Principles

Lesson objectives

1. Understand software Engineering concepts
2. Understand nature of software engineering
3. Be able to describe software engineering body of knowledge
4. Software engineering as profession

Definition SE IEEE Software engineering is application of systematic discipline quantifiable approach to development, operation and maintenance of software that is application of engineering of software and the study of this approach

- Software engineering is concerned with all technical and managements aspects of software production.

Software engineers:

1. specify, design , validate and implement software systems (technical) and manage software projects , develop methods tools required to support software engineering process
2. They apply theories and tools where appropriate and try to discover solution where non exists.

Engineering disciplines are different from other disciplines based

1. Objects/phenomena of study and practices – software engineering is different from other engineering discipline deals with abstract non tangible software while conventional engineering deals with tangible material artifacts.
2. Body of knowledge
3. Design

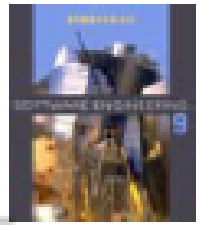
System engineering - It's a discipline concerned with all aspects of computer based systems including hardware, software and process engineering, Software engineering is part that deals will all aspects of software.

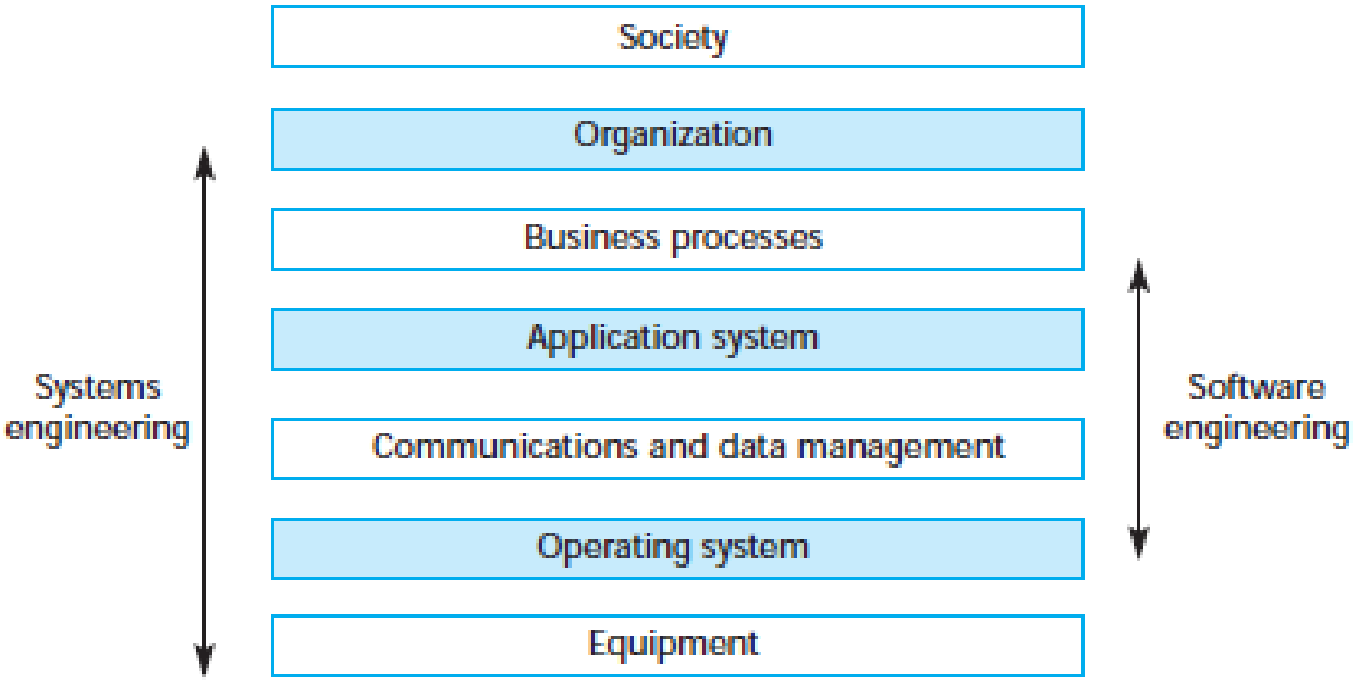
- **Software engineering -** It's a discipline concerned with all aspects of computer based systems including hardware, software and process engineering, Software engineering deals will all aspects of software.

- **Engineering disciplines** including software consist of science knowledge with mathematical foundation and Design knowledge. Several other disciplines have science knowledge including social science but what differentiates engineering disciplines from other disciplines is design knowledge.

SE systems are social technical systems (Sommerville 2011)

The sociotechnical systems stack





Equipment - Hardware devices, some of which may be computers. Most devices will include an embedded system of some kind.

Operating system - Provides a set of common facilities for higher levels in the system.

Communications and data management - Middleware that provides access to remote systems and databases.

Application systems - Specific functionality to meet some organization requirements.

Business processes - A set of processes involving people and computer systems that support the activities of the business.

Organizations - Higher level strategic business activities that affect the operation of the system.

Society - Laws, regulation and culture that affect the operation of the system.

Software is important because it affects all aspects of our lives, computers have become pervasive and many devices including everyday devices like identity cards , ATM cards are embedded computer and contain software written by software engineers

Software success to become one of most widely technologies by business and individuals is result of law of unintended consequences. Fifty years ago nobody would have thought software would become more important than hardware, even thirty years ago some companies did a way with their software business to concentrate on hardware.

- More recently some companies thought PCs software would be most dominant just to have tablet and mobile software become an important segment.
- This indicates a degree of unpredictability of software innovation and fast rate of change
- Software is both a product and a vehicle for delivery of product. Applications software is software that provides useful services, while as vehicle software is used to deliver product by controlling hardware in system software, networks and in creation and control of other programs (environments). Software delivers most important product of our time information.

Some difference between hardware and software

1. Software is engineering not manufactured
2. Software doesn't wear out
3. Software industry is moving towards component based software development although most companies still built custom built software
4. Cloud computing services are going to become a significant sector of software industry with utility computing where software services are provide like utilities like power and water

Some methodologies such as total quality management and CMMI have organization focus on quality and culture that promotes continues improvement. Continuous improvement is technology trend and what has driven software industry.

software engineering layers

Software process holds methods, tools together. A software process is collection of activities actions and tasks performed when a work product is created.

An activity aims to achieved a broad objective – detailed design

Action – consists of set of tasks for developing a work product

Task – focuses on small well defined task e.g. testing

Software engineering methods provide the know how to build software. Software tools automate the process for process and methods

A generic framework for software engineering encompasses the following activities:

Communication and collaboration– between stakeholders to get requirements and ensure that it meets them and communication between computing professionals developing the software.

Planning – Any complex activity needs to be planed and a software plan is like a map to be followed to develop a software artifact.

Modeling

Construction

Deployment

Umbrella activities – these complement framework activities and are carried throughout a software project:

1. Software project tracking
2. Risk management
3. Software quality assurance
4. Measurement
5. Technical reviews
6. Configuration management
7. Work product preparation and production
8. Reusability management

The essence of engineering practices

1. Understand the problem – communication and analysis
2. Plan the solution – design and modeling
3. Carry out the plan – code generation
4. Examine plan for accuracy – testing and quality assurance

Understanding problem

- Stakeholders and their competing interests
- What is known and unknown about how the problem can be solved – data , functions and features
- Can divide and conquer strategy be used – by dividing problem into smaller solvable parts
- Can visual modeling being used to help visualize elements of problem and their interactions

Plan solution

- Have similar problems been solved and can their solutions be reused
- Are their solutions to sub problems that can be reused
- Can solution be represent in a way that lead to effective implementation e.g. design – Solutions such as requirements specification are not an end in themselves they must lead to effective implementation
- Code the solution
- Test the code

1. Professionalism

1.1. Accreditation, Certification, and Licensing- Software engineers should conduct them professionally by adhering code ethics, standards and practices,

1.2. Codes of Ethics and Professional Conduct

1.3. Nature and Role of Professional Societies - professional societies consists of practitioners and academics

1. Professionalism …

1.4. Nature and Role of Software Engineering Standards

1.5. Economic Impact of Software

1.6. Employment Contracts

1.7. Legal Issues

1.8. Documentation

1.9. Tradeoff Analysis

2. Group Dynamics and Psychology

2.1. Dynamics of Working in Teams/Groups

2.2. Individual Cognition

2.3 Dealing with Problem Complexity

2.4. Interacting with Stakeholders

2.5. Dealing with Uncertainty and Ambiguity

2.6. Dealing with Multicultural Environments

3. Communication skills

3.1. Reading, Understanding, and Summarizing

3.2. Writing

3.3. Team and Group Communication

3.4. Presentation skills

Revision questions

1. Discuss why software engineering is important
2. Discuss factors that have influence software engineering from its beginning to today