

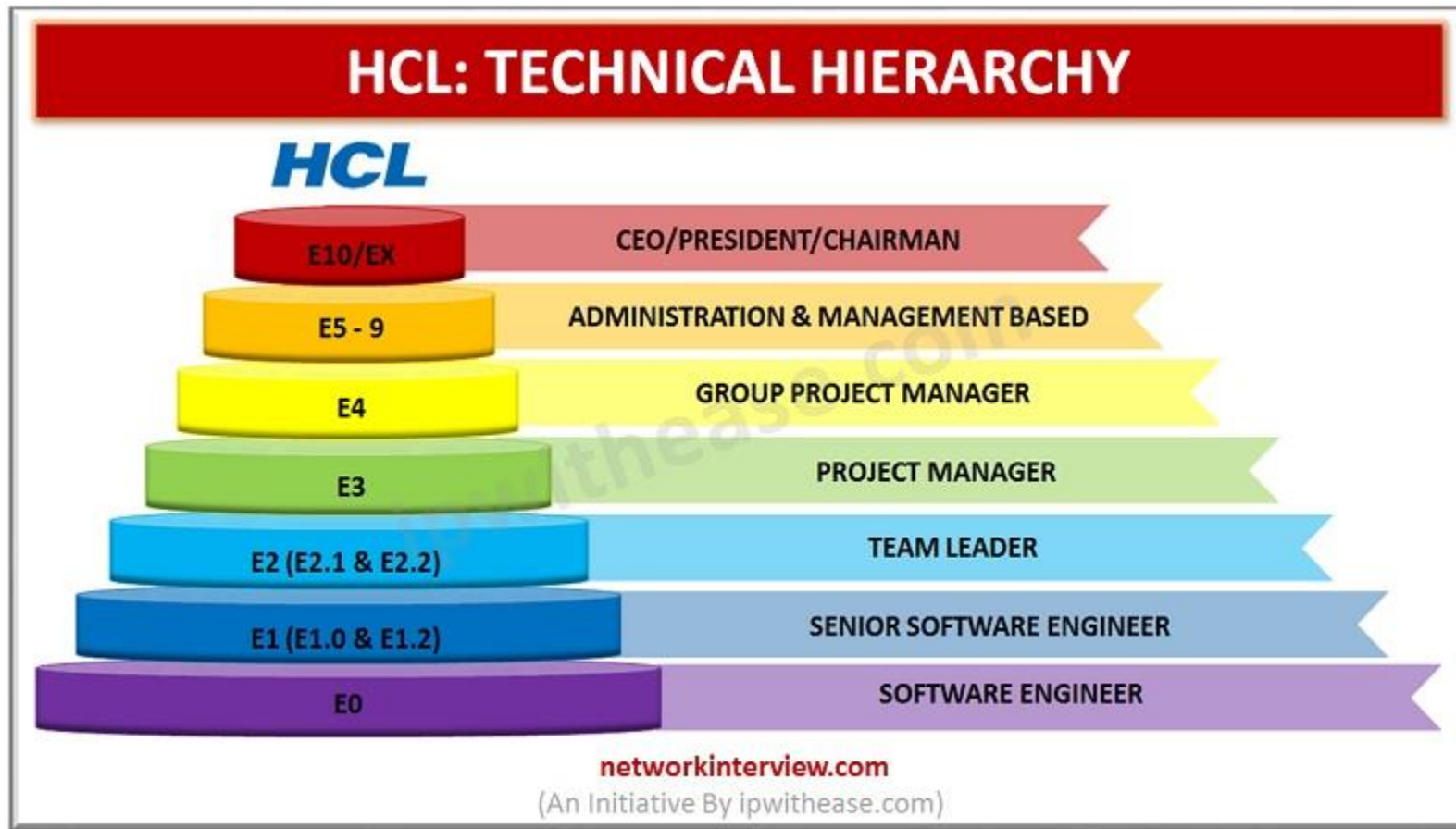
Software Engineering

Srno	Topics to be covered
1	Introduction to Software, Software Engineering Process Models, framework, SDLC
2	Requirement Engineering - Phases of RE , SRS, Analysis Model
3	Design & Architecture , UML Diagrams
4	Agile SDLC & Introduction to JIRA Tool
5	Assignments + SE MCQ Test & Submission(marks)

Objective of this subject

- Learn to develop Professional Software
- Ethics
- Lifecycle of Software Development
- Process & Methods of Software Engineering Discipline
- Quality Product Development

IT Industry Organisation structure



WIPRO: TECHNICAL HIERARCHY



networkinterview.com

(An Initiative By ipwithease.com)

Where are you ?

- @ HOME with Zoom Meeting



Where are you going to?



Ans: Create Develop Software

- Project Sponsor
 - Senior Management
 - Business Unit(HR, Sales, Finance)
 - Customer
- External Stakeholders
 - IT
 - HR
 - Admin
- Project Team
 - Business Analysts
 - Program / Project / Product Manager
 - Quality / Architecture / Analysts
 - Developers
 - Tech Lead, Senior S/W Engg.,
Associate Engg., Junior engineer
 - Tester

Chapter 1- Introduction

Day 1 - Points Covered

- Software
- Difference between Software and Hardware
- Software Engineering, need , Importance
- Types of Software Products, Costs
- Software Engineering Principles/Ethics
- Process /Product/Project
- Software Process Framework
- Software Process Model & Case Studies

What is Software?

- Set of programs
- Data Structures
- Documentation :

What are the characteristics of Software?

1. Logical rather than Physical
2. Software is developed or engineered, it is not manufactured
3. Software does not wear out
4. Moving towards component based construction, however most of the software is custom built

Hardware vs. Software

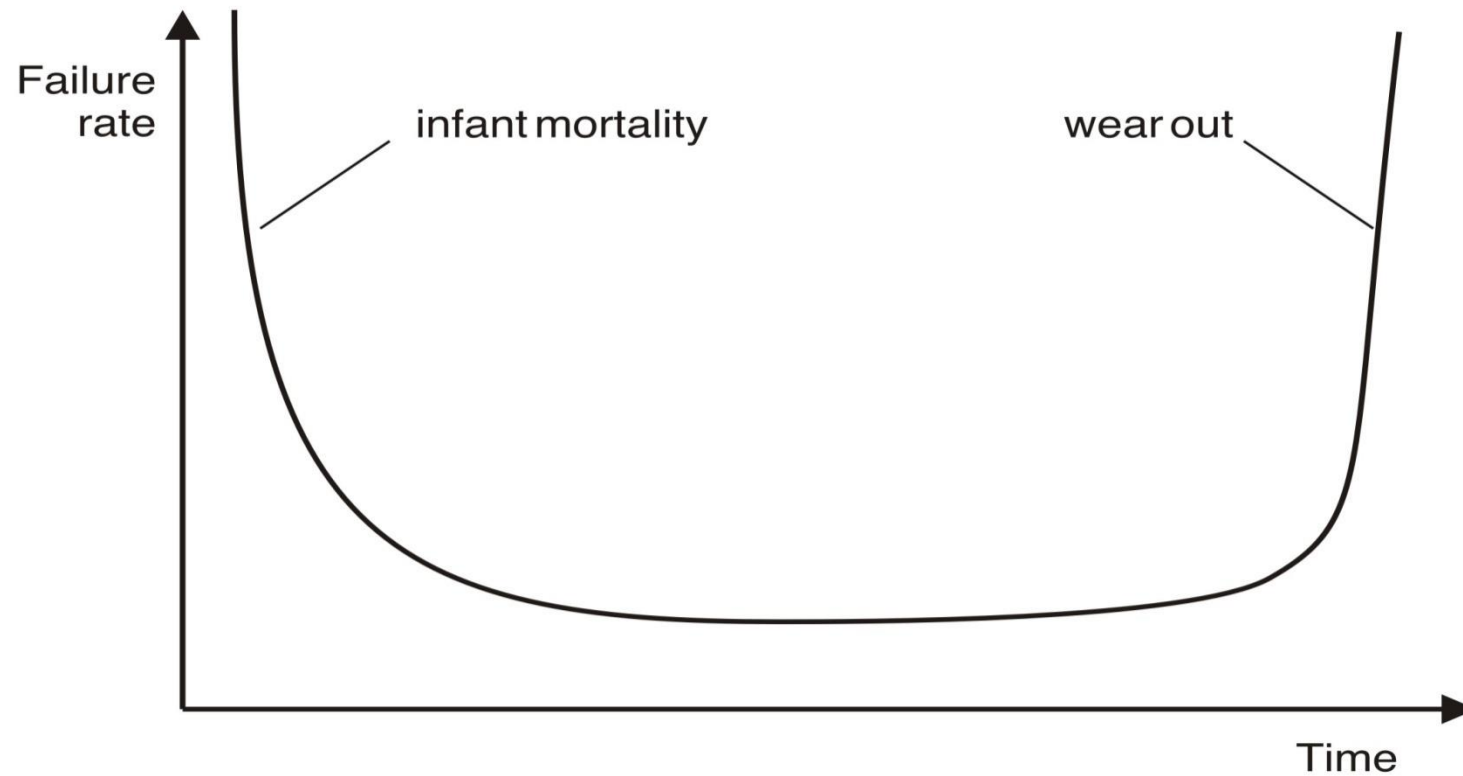
Hardware	Software
<ul style="list-style-type: none">■ Manufactured■ Wears out■ Built using components■ Relatively simple <p>Example : Desk, Car</p>	<ul style="list-style-type: none">■ Developed/Engineered■ Deteriorates■ Custom built■ Com■ Example : Attendance Management System,

Manufacturing vs. Development

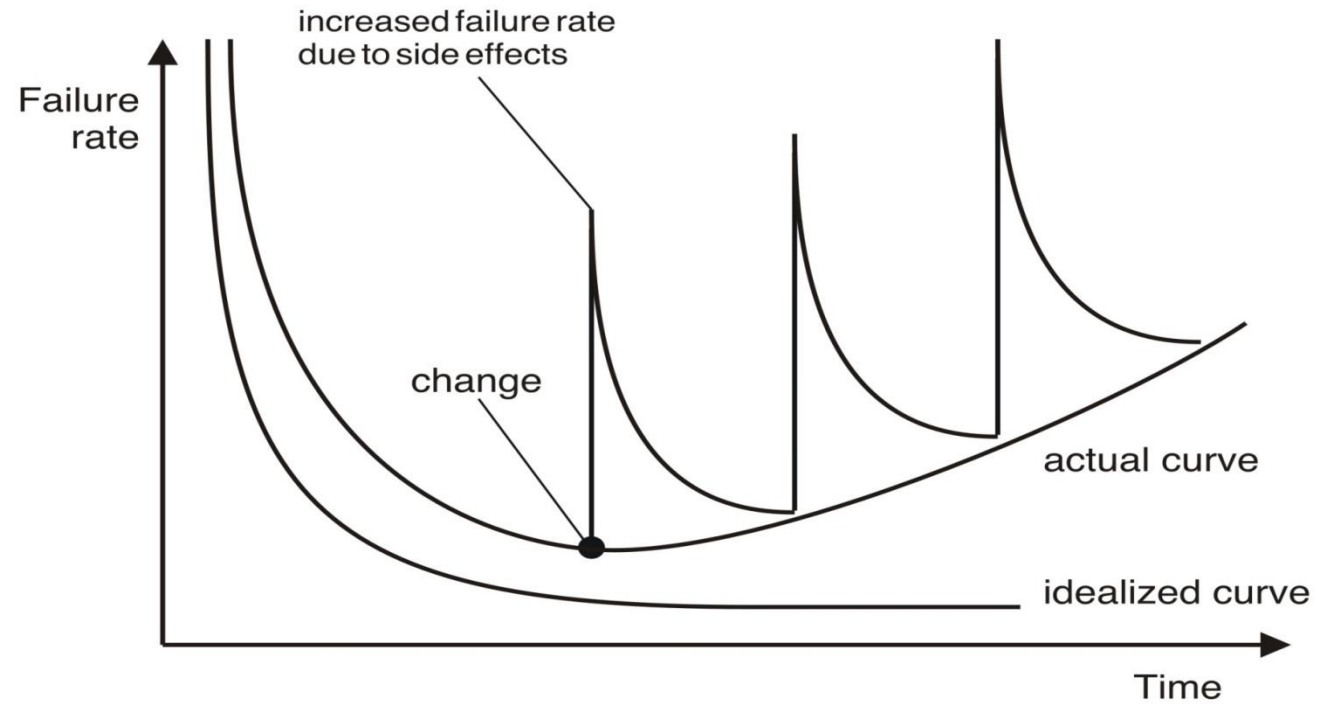
- Once a hardware product has been manufactured, it is difficult or impossible to modify.
- In contrast, software products are **routinely modified and upgraded**.
- In hardware, hiring more people allows you to accomplish more work, but the same does not necessarily hold true in software engineering.
- Unlike hardware, software costs are concentrated in design rather than production.

Wear vs. Deterioration

Hardware wears out over time



Wear vs. Deterioration



Software deteriorates over time

- Hardware can replace parts but not with software
- Software design can be reason for failure, which cannot be mended

Software costs vs Hardware Cost

- Software costs often **dominate** computer system costs.
- Costs of software on a PC are often **greater** than the hardware cost.
- Software costs more to **maintain** than it does to develop.
- For systems with a long life, maintenance costs may be several times development costs.
- Software engineering is concerned with cost-effective software development.

- **Software Myths**
 - Management Myths
 - Customer Myths
 - Practitioners Myth

Management Myth

- We are already having a book **that is full of Standards & Procedures for building** software.
- It will provide my people everything they need to know.
- If we get behind schedule, we can **add more** developers & catch up.
- We can **outsource** the software project to a third party.
- I can just relax & let that firm build it.

Customer Myths

A general statement of objectives is sufficient to begin writing programs.
We can fill in the details later

Project requirements continually change, but the change can be easily accommodated because software is flexible

Practitioners Myth

Once we write the program and get it to work our job is done

Until I get the program running, I have no way of accessing its quality

The only deliverable work product for a successful project is the working program

Software Engineering will make us create voluminous & unnecessary documentation & invariably slow us down

- **Nature of Software**
 - Systems Software
 - Applications Software
 - Engineering / Scientific software
 - Embedded Software
 - Product Line Software
 - Web Applications
 - Artificial Intelligence Software etc

Demand in Change

- **Increasing system complexity – Changes , Demand, Time Constraint**

New software engineering techniques help us to build larger, more complex systems, the **demands change**.

- Systems have to be built and delivered more quickly;
- larger, even more complex systems are required;
- systems have to have **new capabilities** that were previously thought to be impossible.

- **Failure to use software engineering methods- later face issues,change ,cost**

- It is fairly easy to write computer programs without using software engineering methods and techniques.
- Many companies have drifted into software development as their products and services have evolved.
- They do not use software engineering methods in their everyday work. Consequently, their software is often more expensive and less reliable than it should be.

Application Types

- Stand-alone applications
- Interactive transaction-based applications
- Embedded control systems
- Batch processing systems
- Entertainment systems
- Systems for modeling and simulation
- Data collection systems
- Systems of systems

Application brief

- **Stand-alone applications**
 - Run on a local computer, such as a PC, not need to be connected to a network.
- **Interactive transaction-based applications**
 - E-commerce
 - Applications that execute on a remote computer and are accessed by users from their own PCs or terminals.
- **Embedded control systems**
 - Software that control and manage hardware devices.
- **Batch processing systems**
 - Process large Business data
- **Entertainment systems**
 - For Personal entertainment games, netflix
- **Systems for modeling and simulation**
 - Developed by Scientist , Engineers
- **Data collection systems**
 - These are systems that collect data from their environment using a set of sensors and send that data to other systems for processing. Weather forecast

Software products & Specification

- **Generic products – purchased by any customer**
 - 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, Dentist, Jewellery Shop, MSOffice
 - 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- made on customer demand**
 - Software that is commissioned by a specific customer to meet their own needs.
 - Examples – embedded control systems, air traffic control software, traffic monitoring systems, government based, insurance company
 - 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.

Diversity in Software

IT Industry can be seen 2 dimensions

- Product Based / Service Based Industry

- **What is a Product Based Company?**

- A product based company mainly deals with selling a physical or digital product.
- Product development and quality is the core objective of a product based company. Microsoft, Google, Adobe, Dell are some of the examples of a product based company.
- Microsoft's products are Windows OS, Microsoft Office Suite, Microsoft Outlook, Microsoft Edge, etc.
- [Qualcomm](#) is another product based company that sell products like Snapdragon Processors, Snapdragon Modems, etc.
- Apple, one of the product based company giant, sell products such as the iPhone, iMac, iPad, etc.

Service Based

- A service-based company deals with non-physical products. The core business of a service-based company is providing services to clients. Service-based companies are clients of product-based companies. Examples of service-based companies are TCS, Wipro, LTI, Cognizant, Infosys, Capgemini, etc.
- Infosys provides services like database management, network management, blockchain, applied AI, digital marketing, Internet of Things, cybersecurity, testing, etc.
- Persistent is another service-based company that delivers services such as privacy management, identity access management, cloud, infrastructure, intelligent business automation, etc.

- 1. A product based company have high salaries and a high frequency of hikes. In contrast, a service based company provides less salary, and the hike frequency is shallow.
- **Example:** The average salary in a product-based company like Google can range from ₹50,00,000 to ₹1,50,00,000 a year. In a service-based company like Accenture, salaries can range from 3 LPA to 16 LPA.

- 2. The learning curve in a product based company is extraordinary, and decent means are delivered. While in a service-based company, the learning curve is average.
- **Example:** In product-based companies like Microsoft, you can learn new technologies in the computer science space. In a service-based company, you have to struggle for project allocation occasionally.
- 3. Product based companies have an adaptable working time, whereas a service-based company has a strict 8 to 9 working hours. You can even work from home in a product based company.
- **Example:** Some product-based companies like Twitter and Facebook have permanently permitted their staff to work from home.

- 4. A product based company has a more creative, efficient, productive, and complicated job than a service-based company.
- **Example:** Facebook and Google are working on futuristic technologies like holograms, augmented reality, virtual reality, etc.
- 5. The recruitment process in a product-based company is callous, whereas, in a service-based company, it is straightforward to crack an interview.
- **Example:** There are at least six stages in the recruitment process of the product based companies such as online round, on-site round, technical round one, technical round two, technical round three, and technical-cum-HR round. In a service-based company, there are at the most three rounds.
- 6. A product based company hire a very fixed number of candidates, whereas a service-based company carries mass recruitments.
- **Example:** Microsoft hires one or two candidates from a tier-one college on average. Infosys, which is a service-based company, hires 50 to 100 candidates in single on-campus recruitment.
-

Examples

Product Based Companies

- Amazon
- Google
- SAP
- IBM
- Microsoft
- Flipkart

Service Based Companies

- Accenture
- TCS
- Infosys
- Wipro
- Tech Mahindra
- Cognizant

3P's = Product /Process/Project

Professional software development

Background Software engineering

- The economies of ALL developed nations are dependent on software.
- More and more systems are software controlled
- Software engineering is concerned with theories, methods and tools for professional software development.
- Expenditure on software represents a significant fraction of GNI in all developed countries.

Software engineering diversity

- Due to diversity in software there is no fix pattern /principle/policy or method to develop a software
- Selection depends on many factors like user, stake holders, business needs, feasibility , budget, resources etc

Software engineering

- 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.

Frequently asked questions about software engineering

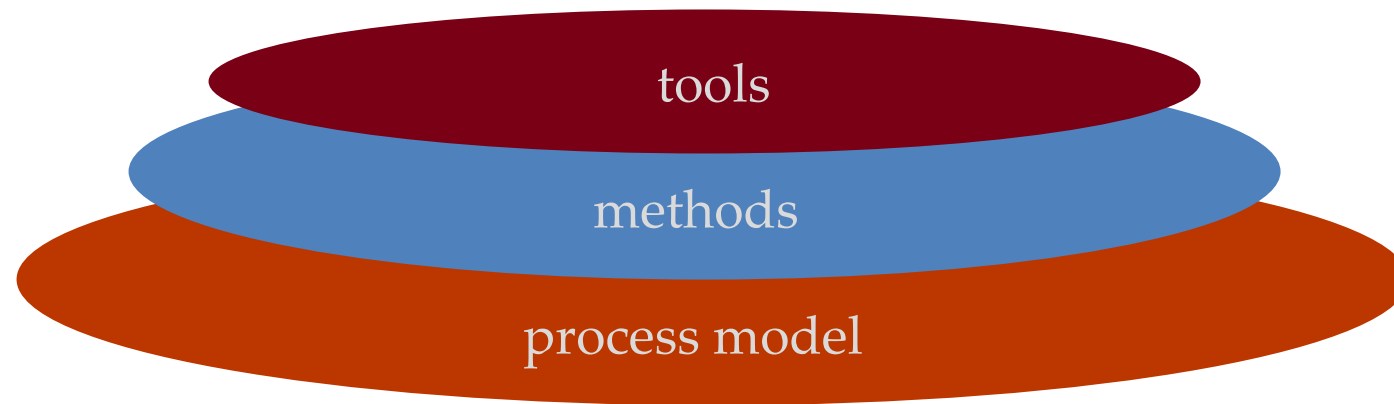
Question	Answer
What is software?	Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.
What are the attributes of good software?	Good software should deliver the required functionality and performance to 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.

A Layered Technology

Software Engineering



a “quality” focus

Software Engineering Layers

- Quality Focus : Organizational commitment to quality (TQM, Six Sigma)
- Process : Defines a framework (Foundation for Software Engineering)
- Methods : “How to”s for building software (Tasks)
- Tools : Automated or semi-automated support (Rational Rose, CASE tools)

General issues that affect Software

- Heterogeneity
- Business & Social Change
- Security & trust
- Scale

General issues that affect software

- **Heterogeneity**
 - Increasingly, systems are required to operate as distributed systems across networks that include **different types of computer and mobile devices**.
- **Business and social change**
 - Business and society are changing incredibly quickly as emerging economies develop and new technologies become available
 - They need to be able to change their existing software and to rapidly develop new software.
- **Security and trust**
 - As **software is intertwined with all aspects of our lives**, it is essential that we can trust that software.
- **Scale**
 - Software has to be developed across a very wide range of scales, from very small embedded systems in portable or wearable devices through to Internet-scale, cloud-based systems that serve a **global community**.

Importance of software engineering

- To develop a Reliable & Trustworthy Software
- Economically cheap, can be used for Long run
- Society advancing in use of software
- Demanding Change in needs

Core Principles of Software Engineering

- 1.The Reason it all Exists
- 2.Keep It Simple, Stupid (KISS!)
- 3.Maintain the Vision
- 4.What you Produce, others will Consume
- 5.Be Open to Future
- 6.Plan Ahead for Reuse
- 7.THINK

Modern SE Principles



Ethical principles – A pledge for Software Developer

1. PUBLIC - Software engineers shall **act consistently with the public interest.**
2. CLIENT AND EMPLOYER - Software engineers shall act in a manner that is in the **best interests of their client and employer consistent with the public interest.**
3. PRODUCT - Software engineers shall ensure that their products and related modifications **meet the highest professional standards possible.**
4. JUDGMENT - Software engineers shall **maintain integrity and independence in their professional judgment.**
5. MANAGEMENT - Software engineering managers and leaders shall subscribe to and promote an **ethical approach** to the management of software development and maintenance.
6. PROFESSION - Software engineers shall **advance the integrity and reputation of the profession consistent with the public interest.**
7. COLLEAGUES - Software engineers shall be **fair to and supportive of their colleagues.**
8. SELF - Software engineers shall **participate in lifelong learning regarding** the practice of their profession and shall promote an ethical approach to the practice of the profession.

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.

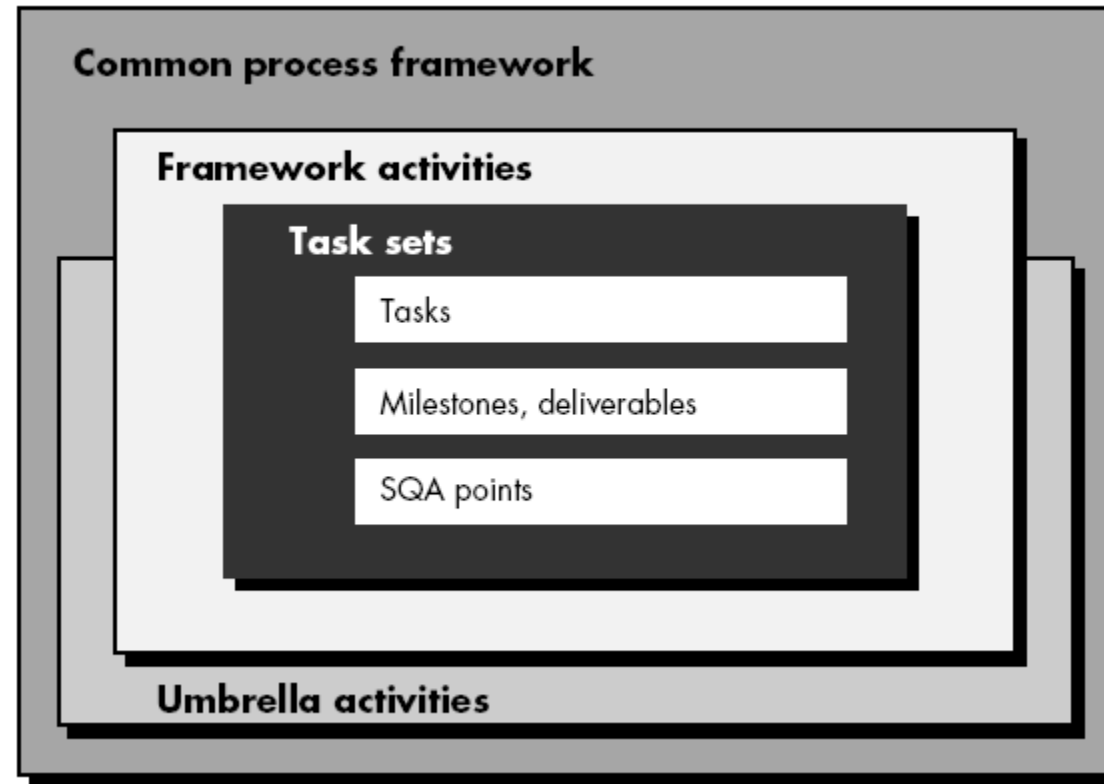
Product/Process/Project

- Product is Software (Outcome)
- Process (Technique steps , method)
- Project is (A certain goal to be accomplished in a period of time)

Software process activities

- Software **specification**, where customers and engineers define the software that is to be produced and the constraints on its operation.
- Software **development**, where the software is designed and programmed.
- Software **validation**, where the software is checked to ensure that it is what the customer requires.
- Software **evolution**, where the software is modified to reflect changing customer and market requirements.

Software Process Framework



- CMMI – Capability Maturity Model Integration
- Level 0 to level 4

- The process of framework defines a small set of activities that are applicable to all types of projects.
- The software process framework is a collection of task sets.
- Task sets consist of a collection of small work tasks, project milestones, work productivity and software quality assurance points.

Common Framework activities

- CPMCD
- Communication – project initiation, requirement gathering
- Planning- estimating , scheduling ,tracking
- Modeling – Analysis, Design
- Construction – Code , test
- Deployment –Delivery , Support , Feedback

Umbrella Activity

- Project Tracking & Control
- Risk management
- SQA- Software Quality Assurance
- SCM- Software Change Management
- Technical Reviews

- **Typical umbrella activities are:**

- **1. Software project tracking and control**In this activity, the developing team accesses project plan and compares it with the predefined schedule.
 - If these project plans do not match with the predefined schedule, then the required actions are taken to maintain the schedule.
- **2. Risk management**Risk is an event that may or may not occur.
 - If the event occurs, then it causes some unwanted outcome. Hence, proper risk management is required.
- **3. Software Quality Assurance (SQA)**SQA is the planned and systematic pattern of activities which are required to give a guarantee of software quality.
For example, during the software development meetings are conducted at every stage of development to find out the defects and suggest improvements to produce good quality software.
- **4. Formal Technical Reviews (FTR)**FTR is a meeting conducted by the technical staff.
 - The motive of the meeting is to detect quality problems and suggest improvements.
 - The technical person focuses on the quality of the software from the customer point of view.
- **5. Measurement**Measurement consists of the effort required to measure the software.
 - The software cannot be measured directly. It is measured by direct and indirect measures.
 - Direct measures like cost, lines of code, size of software etc.
 - Indirect measures such as quality of software which is measured by some other factor. Hence, it is an indirect measure of software.
- **6. Software Configuration Management (SCM)**It manages the effect of change throughout the software process.
- **7. Reusability management**It defines the criteria for reuse the product.
 - The quality of software is good when the components of the software are developed for certain application and are useful for developing other applications.
- **8. Work product preparation and production**It consists of the activities that are needed to create the documents, forms, lists, logs and user manuals for developing a software.

- Think For Domain!



- Each Breakout will be labelled and assigned
- Health & Research Management
- Education (e-Learning & exam)
- E-commerce
- Banking Finance /Insurance
- Defence /Army/Military
- Geological &Weather Forecast
- Travel & Tourism
- Sports & Entertainment
- Government Services&License
- Software Industry tools

- Bio –Medical /Diagnosis
- Mobile Apps- UTS Railway Travel Booking Reservation
- Shipping Services
- Government Services
- Courier Delivery System
- AI /Robotics
- Natural Calamities – Weather Rain Forecast
- Social Media App – Instagram , FB
- Sports Software
- Entertainment – Netflix/Hotstar/Amazon Prime
- Aerospace, Satellite Monitoring
- Education
- Defence & Army , navy