18-Feb-2024

SOFTWARE ENGINEERING;

What is Software Engineering:-

It is concerned with theories, method and tools for professional software developement. It also concerned with cost-effectiveness of software developement as software cost often dominate computer hardware and sometimes maintenance cost is quite higher than a cost of developing

Software Project failure; (Why software project fails?)

i) Increasing system complexity:- SE helps us to built large, complex system. A system should made feasible to adapt changes and demand.

ii) Donot use SE methods:- A proper guidelines can help in cost effictive and increase adaptation but not using SE result in system expensive.

Software Product and its Specifcation:-

i) Generic Products:- Stand-alone systems that are marketed and sold to any customer. Example:- Management tools, CAD softwares.
It Specification is what software do is owned by developer and changes made to software resides to developer.

ii) Customized Products:- Software that is commissioned by a specific customer to meet their own needs. Example:- Embedded control systems, Air traffic control. It specifcation is what software do is owned by the customer and they make decision to changes if required.

Essential attributes of good software:-

i) Maintainability:- Software should be easily adaptable to changes.

ii) Security & Dependable:- software should not cause physical or economic damage in the event of failure

Importance of SE:- i) Society rely on advanced system. SE helps to produce system which are easily adaptable and economic friendly.
ii) SE methods are cheaper to use which helps in long run of systems.

iii) Efficiency:- Software should not waste resources like memory, CPU cycl
iv) Acceptability:- Compatible with other system and acceptable to all type of user

Software process Activities:-
i) Specification:- Customer & engineers define the software and its constraints
ii) Development:- Software is designed and programmed.
iii) Validation:- Software is checked to ensure that it meets customer requires
iv) Evolution:- Software is modified to customer and market requirements

General issues that affect software:-
i) Heterogeneity:- Systems are operated on distributed system across networ that includes different types of computer & mobile devices.
ii) Business and social change:- Emerging technologies in market result in out-dated system hence new development required according to marke
iii) Security and trust:- A software should be trustworthy and covers all aspect of lives, it is essential a user should feel secured and trusted.
iv) Scale:- Software developed across a very wide range of scales, from embedded system to wearable devices through cloud-based systems

Application types:-
i) Stand-alone Application:- App run on local computer such as PC. Includes all necessary functionality and donot need an internet connection.
ii) Interactive transaction-based Applications:- App execute on remote computer & accessed by user from their own PCs; E.g:- Web-App.
iii) Embedded control systems:- Software controls & manage hardware device. These are embedded system which performs specific task efficiently.

ii) Responsibilities of Engineers:- Responsibility to act ethically and uphold professionals standards to ensure Safety, Privacy, and well-being of users and stake-holder.

---

iv) Batch Processing Systems:- Business systems that are designed to process data in large batch. Process large individual inputs to create outputs.

v) Entertainment systems:- System that are primarily for personal use and which are intended to entertain the user.

vi) Systems for modeling and Simulation:- Developed by scientists and engineers to model physical processes or simulations.

vii) Data collection systems:- Systems which collects data from their environment using set of sensor and send data for processing.

viii) Systems of systems:- Systems composed of number of other system

Software Engineering Ethics:-

Software Engineers must behave in an honest and ethical responsible way. Ethical behaviour is more than simply upholding the law but involves following set of principles:-

i) Confidentiality:- Engineers should respect the employee's or client's confidentiality irrespective a formal agreement has been signed.

ii) Competence:- Engineers should not misrepresent their level of competence. They should not accept work which outwith their ability.

iii) Computer Misuse:- Engineer should not use technology of client's device for their own use. Sometimes misuse can be game playing.

iv) Intellectual property rights:- Engineers should aware of local laws governing the use of property such as patents, copyrights, etc.

So; Upper 4 points are how an Engineer can show ethics in work, ↳ principle & guidlines

Now; how ethics is establish:-

i) Ethical considerations:- Software Engineers must consider the potential impact of their work on individual, society, and the environment.

Software Process and Description:-

A structured set of activities required to develop a software system.

There 4 necessary processes includes Specification, Design, Validation, Evolution

⇒ A software process model is an abstract representation which presents a description of a process from particular perspective.

⇒ A process description specifies data model, design a user interface and also includes ⇒ i) Products ii) Roles (responsibility of the people), iii) Pre- and Post conditions.

Plan-Driven Processes:- All process activities are planned in advanced and progress is measured against the plan.

Agile Processes:- Planning is incremental and its easier to change the process to reflect changing customer requirements.

Types of Software Process Models:-

*) Waterfall Model:- A plan driven model. Have separate and distinct phases of specification and development. The phases includes:-

i) Requirements definition; ii) System and software Design.

iii) Implementation & testing iv) Integrate & system test; v) Maintenance.

Pros:- i) Simple and easy to understand ; ii) Clear project milestone

iii) Documentation centric approach result in early fault/error detection.

Cons:- i) Difficulty of accomodating change after process is underway.

ii) less flexible due to distinct stages, so difficult to respond customer's changes

iii) Stages oriented model, As a phase has to complete before moving to next

*) Incremental Development:- Based on specification, development and validation are interleaved. May be plan-driven or agile.

Pros:- i) The cost of accommodating changing customer requirement is reduced

ii) Easier to get customer feedback on development work that has been done

iii) Rapid delivery & development of software can be achieved.

Cons:- i) Process is not ~~possible~~ visible as manager need regular deliverable to justify the progression of software.

ii) System structures tends to degrade as new increments are added.

*) Integration and configuration:- The system is assembled from existing configurable components. May be plan-driven or agile.

Based on "reusability"; Reused elements may be configured to adapt their behaviour according to user's requirements.

Types of reusable software:- i) Stand-alone; ii) Frameworks like .NET

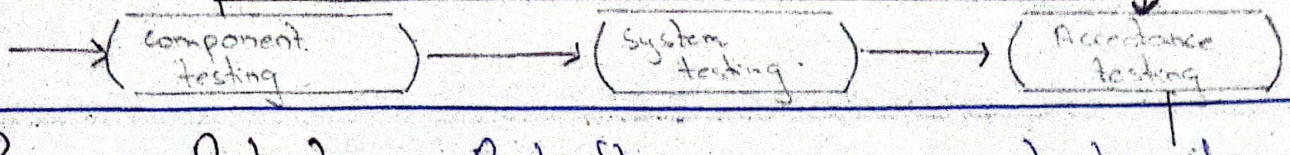
Reuse-oriented Software Engineering:- Pros:- i) Reduced cost and risks as less software is developed from scratch.

ii) Fast delivery and development of system.

Cons:- i) Requirements of users compromises.

ii) Loss control over evolution of reused system software.

ii) System testing:- Testing of the system as a whole.
iii) Customer testing:- Testing with customer data to check the system meets the customer needs.



Process Activities:- Real software processes are inter-leaved sequence of technical activities with goal of 4 main stages/process.
These process are organized differently in different development processes.

i) Software Specification:-
Process of establishing what services are required & constraints on the system's operation and development.
Include Requirements Engineering process which have following:-
1) What does system stakeholder requires ; (2) Defining the requirements in detail ; (3) Check validity of the requirement.

ii) Software Design and implementation:- Process of converting the system specification into an executable system. So Design & implement are closely related and may be inter-leaved.
⇒ Software Design:- Design structure that realises the specification.
includes:- Database design, interface Design and Component Selection.
⇒ System Implement:- Translate structure into executable program.
includes:- Programming and Debugging phases to configure an App.
Key Idea of This phase :- *) Decompose systems into modules, specify interface between modules, Code up the design, Test each module, put the piece/modules together and then test the entire system.

iii) Software Validation:- Verification and validation to show that system conforms to its specification and meets the requirements.
Involves checking and review processes and system testing.
Stages of testing:- i) Component testing:- Individual components are tested independently.

Coping with change requirements;

→ Maintenance.

iv) System Evolution:- Software is inherently flexible and can change. As requirements change through changing business circumstance, thus a softwar should support change and also evolve.

S10:- Coping with Change Requirements;

Change is unavoidable in all large software projects. Changes leads to both re-analyzing requirements as well as cost of new fun

⇒ To Reduce cost of re-work:-

i) Change anticipation; where the software process includes activities that can anticipate possible changes.

ii) Change tolerance, where the process is designed so that changes can be accomodate at relatively low cost.

These two are achieved widely in incremental model.

i) Software Prototyping:- This is an approach which supports change anticipation, As prototyping refers to a vision of the system which is developed quickly to check the customer's requirement.

⇒ A prototype can be used in :- i) Requirements elicitation & validation in requirements engineering process; (ii) Explore option and develop UI design; (iii) Testing process to run back-to-back tests.

⇒ Benefits of Prototypes:-

i) Improved system usability; (ii) Improved design quality; iii) Improved maintainability; (iv) Reduced development efforts.

Prototype Development:- i) A prototype should be focus on areas of the product that are not well understood.
ii) Error checking and recovery.
iii) Focus on functional rather than non-functional requirements such as security and reliability.

ii) Incremental Delivery:- Rather than delivery the system as a single delivery, the development and delivery is broken down into increments.. Highest priority requirements are included in early increments.

Incremental Delivery and Development:-
⇒ Development refers to develop the system in increments and evaluate each increment before processing to next increment. Normal approach used in agile methods. Evaluation done by user/customer.
⇒ Delivery refers to deploy an increment for use by end-users. More realistic evaluation about practical use of software.

Pros of Delivery :- i) System functionality is available earlier.
ii) Early increments acts as prototype to help requirements for later increment
iii) Lower risk of overall project failure.
iv) Highest priority requirements tends to have highest/most testings.
Cons of Delivery:- i) Requirements are not defined in detail until an increment is to be implement; hence hard to identify common facilities.
ii)

Primary characteristics to Agile is :- (i) Rapid Delivery; (ii) Responsiveness to changing customer requirements

---

S10 :- Process Improvement; It means understanding existing processes and changing these processes to increase product quality and reduce costs and developement time.

Approaches to Process Improvement;

i) Process Maturity approach which focuses on improving process and project management and introduce good SE practices and tools/methods.

ii) The Agile Approach, which focuses on iterative development and reduce the overheads in the software process.

Process Improvement cycle;

(i) Process Measurement :- Measures one or two atributes of the software process or product. These measurements forms a baseline that helps you decide if process improvements have been effective.

Ways for Process Measurement;

i) Wherever possible, quantitative process data should be collected.

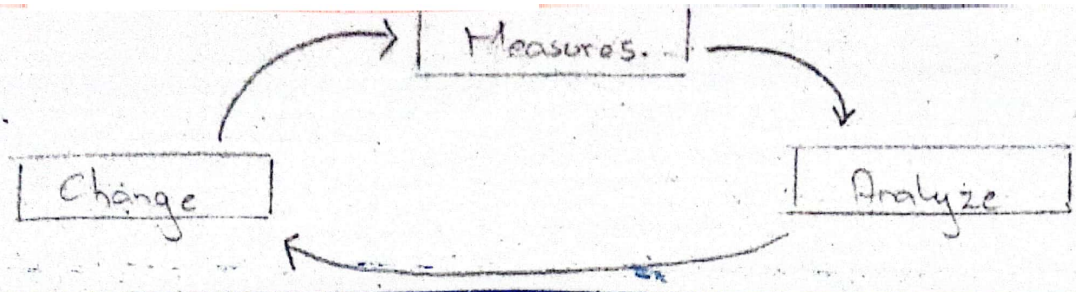
⇒ There are some metrics for process:-

i) Time taken for process activities to be completed. (Calender time)

ii) Resource required for processes or activities. (Efforts in person-day)

iii) Number of occurence of particular events. (Defects discovered).

ii) Process Analysis :- Current process is assessed; identifying weaknesses and bottlenecks. Process models/maps that describe the process may



iii) Process Change:- Addresses some of the identified process weakness. These are introduced and the cycle resumes to collect data about the effectiveness of the changes.

*) The SEI Capability "Maturity Model"

Level-1= Initial = Essentially uncontrolled

Level-2= Repeatable= Product Management procedures defined and used

Level-3= Defined= " " " and strategies defined

Level-4= Managed= Quality Measurement strategies defined and used

Level-5= Optimizing= Process Improvement strategies defined and used.