Date

Intro To Software Engin. Day

study & application of methodologies to develop quality software that fulfill customer needs "

Objectives:

↳ On time : is delivered at the decided date

↳ Reliable : doesn't crash

↳ Complete : fulfill customer needs

- Diff b/w software & system engineering
  - ↳ System engineering is concerned with all aspects of computer-based systems development including hardware, software & process engin.
  - ↳ Software engineering is part of this process concerned with developing the software infrastructure control, applications & databases in the system

- Attributes of Good Software
  - Maintainable
  - Dependable (Reliable, safe secure)
  - Efficient (memory, process cycles)
  - Acceptable

Date _____ Day _____

- Stages for Software Development

Req Analysis → Software Design → Implementation → Testing → Maintenance

- Software Application Domains
  - System Software
  - Application Software
  - Engineering / Scientific Software
  - Embedded Software
  - Product-line Software
  - Web Apps
  - AI Software

"Process of productive use of scientific knowledge is called engineering"

WEEK #02

"Process is generally a set of phases, each phase performs a well defined task and generally produces an output."

→ Project fails due to [Managerial] problems
→ Framework means; set of rules to be followed

Date _____ Day _____

"A structured set of activities req to develop a software system is called [software process]"

"A software process model is an abstract representation of a process"

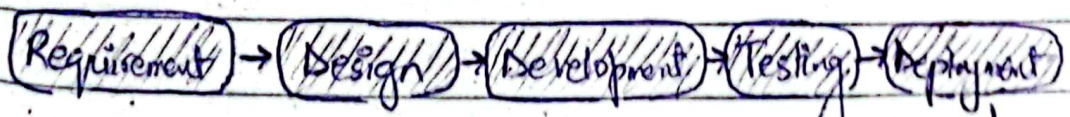
→ Plan Driven Process:

Those process where all of the process activities are planned in advance and progress is measured against this plan –

→ In agile processes, planning is incremental and it is easier to change the process to reflect changing req.

① Waterfall Model:

→ Used in context of large, complex projects.
→ Useful when proj req are well defined
→ Projects having long timelines, little room for error
→ Project stakeholders need to have high level of confidence in the outcome.



↳ Each new phase begins when workproduct of previous has been completed

Date _____ Day _____

→ Problems / Disadvantages:

- Difficult to respond to changing customer require
- Can only be used for large-scale projects
- Unable to work in high uncertainity
- Req^s needs to be regid
- Doesnot handle concurrent events
-

→ Advantages:

- Easy to understand
- Each phase is processed individually
- Each stage in the model clearly defined
- Properly Documented

→ Appications:

- Safety Critical Systems
- Govt & defense projects
- Proj with stable req

(2) Iterative Development Process:

→ Starts with a simple implementation of a subset of the software req and iteratively enhances the evolving versions until the full system is implemented

→ At each iteration design modifications are made

Advantages:

→ The cost of accomodating changing customer req is reduced.
→ Easier to get customer feedbacks
→ More rapid delivery & deployment

Disadvantages:

→ The process is not visible.
→ System structure tends to degrade as new incr. are added
→ More management attention is required.
→ End of project may not be known which is a risk.

(3) Spiral Model:

→ Spiral is primary Risk driven approach
→ Consist of diff cycles. In each cycle try to address some risk elements

i- Planning: Determines objectives, alternatives and constraints

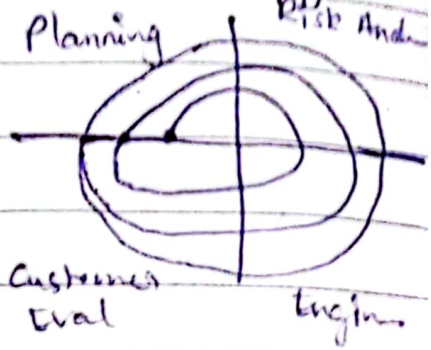
ii- Risk Analysis: Analysis of alternatives as well as an identification and/or resolution of risk

iii - Engineering: Development of next level of product

iv - Customer Evaluation: Assessment of the results of engineering

Date _____ Day _____

→ At each iteration more progressive versions of software are built



▷ SOFTWARE SPECIFICATION:
The process of establishing what services are required and the constraints on the system's operation and development.

→ Requirement Engineering Process
↳ Req elicitation and analysis
↳ What do the system stakeholders req or expect from the system?
↳ Req Specification
↳ Defining the req in detail
↳ Req validation
↳ Checking the validity of req

- Softwar Design: Designing a software structure the realises the specification
- Implementation: Translation of this structure into an executable program.

System Prototyping, where a version of the system or part of sys is devel quickly to check customer's req

Date _____ Day _____

- Change anticipation, where the software process includes activities that can anticipate possible changes before significant re work is required.
  ↳ is toha such ke software develop kerna ke agr aage koi change aye to easily accomodate ho sake.

- Change Tolerance, where the process is designed so that changes can be accomodate at relatively low cost.
  ↳ incremental delivery, where system increments are delivered to the customer for comment and experimentation.

▷ Software Prototyping:

↳ A prototype is an initial version of a system used to demonstrate concepts and try out design options

→ Can be used in req/ engin. process, in UI design, in testing process.

→ Prototypes should be discarded after development as:
  ↳ they donot meet non-functional req/
  ↳ they are undocumented
  ↳ does not meet quality std

Date _____ Day _____

→ Prototype should focus on areas of product that are not well understood.

→ Error checking & recovery may not be included

→ focus on functional rather than non-functional req

Benefits:

- Improved system usability
- A closer match to user's real needs
- Improved design quality
- Reduced development effort

▷ Design Activities:

(1) Architectural Design: Where you identify the overall structure of the system, the components & their relationship

(2) Database Design: Where you design the system data structures and how these are to be represented in DB.

(3) Interface Design: Where you define the interfaces between system components.

(4) Component Selection & Design: Where you search for reusable components. If unavailable you design how it will operate.

Date _____ Day _____

▷ Software Validation:

→ Verification & Validation (V&V) is intended to show that a system conforms to its specification and meets the req of customer

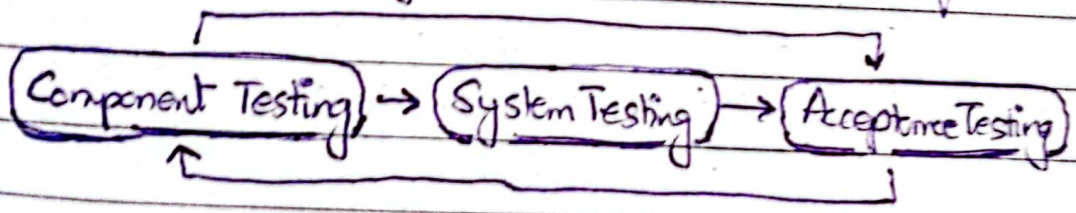
→ System testing involves executing the system with testcases that are derived from the specification
↳ most common activity in V&V.

▷ Testing Stages:

① Component Testing: Individual components are tested individually. they may be functions or objects.

② System Testing: Testing of a system as a whole.

③ Customer Testing: Testing with customer data to check that the system meets customer's req



▷ Incremental Delivery:

→ Incremental Development:

- Develop the system in incs and evaluate each incs. before proceeding to the dev of next incs.
- Used in agile methods
- Evaluation done by user/customer

Date Day

① Initial :

↳ Processes followed are immature & not well defined
↳ Unstable enviornment
↳ No basis for predicting product quality, time for completion
↳ Highly dependent on individual skills rather than standardized process
↳ High risk of project failure.

② Repeatable

↳ Focuses on establishing basic project management policies
↳ It includes project planing – defining resources req, goals, constraints etc.
↳ Focus is on mantaining the performance of product
↳ It involves maragment of customer reviews which results in some change in req set. And also accomodate them.
↳ It involves software quality assurance

③ Defined :

↳ Documentation of std guidelines & procedure takes place.
↳ It involves peer reviews-

Date _____ Day _____

↳ Involves planned interactions b/w diff. develop. teams.
↳ Focuses on activities & practices that should be followed to improve the process.
↳ It also focuses on training programs.

④ Managed:
↳ Quantitative quality goals are setup for the organization for software products as well as software process.
↳ Measurement helps org. predict product/process quality
↳ It includes establishment of plans & strategies
↳ Focuses on controlling project performance

⑤ Optimizing:
↳ Focuses on continuous process improvement in the organization using feedback
↳ New tools, techniques & eval. of software process is done to prevent the recurrence of known defects
↳ Focus on continuous process improvement
↳ Focus to use/adapt new technologies
↳ Focus to improv product quality & decr dev time
↳ Focus to identify the cause of defect & prevents them.

Date _____ Day _____

"WEEK # 03"

– AGILE SOFTWARE DEVELOPMENT –

▷ Difference b/w Incremental & Iterative Development

① Incremental Development:

→ The sys is developed in small fully functional parts (increments) each adding new features until the complete system is built.

→ Each increment is usable and adds functionality.

② Iterative Development:

→ The system is built & improved through multiple cycles (iterations), refining & reworking prev versions based on feedback until final sys meets all req.

→ Focuses on refining & improving the entire sys repeatedly.

▷ Characteristics of Agile Dev.:

① Program specification, design and implementation are inter-leaved

② The sys is developed as a series of increments

③ Frequent delivery of newer versions for evaluation

Date _____ Day _____

④ Extensive tool support to support dev.
③ Minimal documentation

Plan Driven	Agile Development
Follows a fixed plan	① Iterative & flexible
Involves detailed docum-	② Minimal documentation involved, focuses on working project
Req defined before dev	③ Req evolve during dev based on feedback
Delivered at the end after full development	④ Delivered in small, working increments
Change handling is costly & difficult	⑤ Changes are welcomed & incorporated easily
Limited customer involv- ment.	⑥ Continuous customer involvment

▷ Agile Core Values:

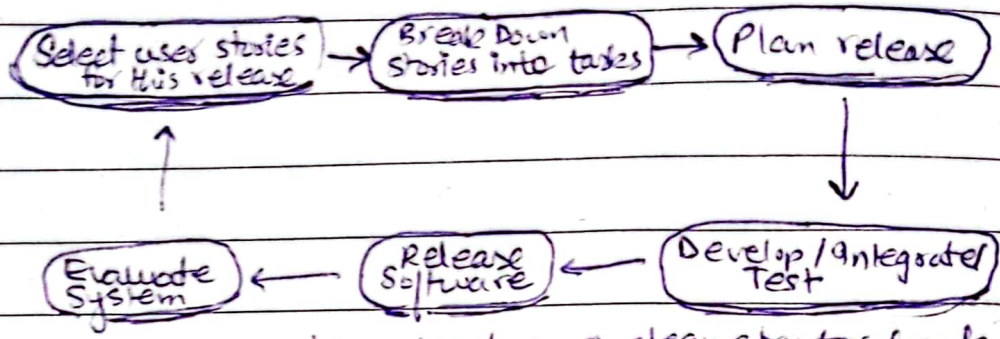
→ Individuals & interaction over processes & tools
→ Working software over comprehensive doc
→ Customer collaboration over contract negotiation
→ Responding to change over following a plan.

" An incremental software process designed to cope with change "

Date _____ Day _____

▷ EXTREME PROGRAMMING (XP)

↳ One of the agile dev techniques.
↳ Like iterative but taken to extreme.



→ Changes are easy to make bcz of clean structured code
→ Expert customer is part of the team
↳ Onsite, available constantly
→ Customer involved active in all stages
↳ Clarifies req
↳ Negotiate
↳ Writes & runs acceptance tests

& architectural refactoring takes time & costs

▷ Extreme Programming Practices

• Incremental Planning:
→ Requirements are recorded on story cards and the stories to be included in a release are determined by the time available & their relative priority
• Small Releases:
→ the minimal useful set of functionality that provides business value is developed first. Releases of the system are frequent and incrementally added.

Date _____ Day _____

- **Simple Design:**

→ Enough design is carried out to meet current req

- **Test-First Development:**

→ An automated unit test framework is used to write tests for a new increment to be implemented.

- **Refactoring:**

→ All developers are expected to refactor the code continuously ~~as~~ as asap code improvements are found.

- **Pair Programming:**

→ Developers work in pair, checking each other's work & providing feedback.

- **Collective Ownership:**

→ The pairs of developers work on all areas of the system, & all developers take responsibility of all code

- **Continuous Integeration:**

→ As soon as work on a task is completed it is soon integerated into the system. And unit test for system is then performed

- **Sustainable Pace:**

→ Large ammounts of overtime not acceptable

- **On site Customer:**

→ A representative of end-user (customer) should be available onsite for full time, he is member of dev team.

Date _____ Day _____

▷ Code Refactoring :

① Reorganization of class heirarchy to remove duplicate
② Tiding up & renaming attributes to make them easier to understand
③ The replacement of inline code with function calls

→ Don't refactor in middle of implementation.

▷ User Stories :

→ Each story is broken into tasks

Story : customer-centric description
Task : developer-centric description

→ Example :

Story : "I can create named accounts"
Task : "Ask the username of account"
"check to see if already exist"
"create any empty account"

▷ Why Writing Tests First?

① Testing first clarifies the task at hand
② Forces you to think in concrete terms
③ Helps identify edge cases
④ Forces simplicity
⑤ Act as useful document.

Date _____ Day _____

▷ Pair Programming Advantages:

→ Knowledge is shared between the persons in pair — knowledge & skill migration
→ Develops common ownership of code
→ Serves as an informal review process
→ Encourages refactoring
→ Reduces risks & results in better code

▷ When NOT TO use XP:

(1) Req truly known & fixed
(2) Cost of late changes is very high
(3) Your customer is not available

▷ SCRUM:

"Agile method that focuses on managing iterative development rather than specific agile practices"

• Roles in Scrum
• Development Team:

→ A self organizing group of software developer with no more than 7 people.
→ Consist of
  ↳ Testers, UI/UX designer, Ops Engin., & developers
→ Work collaboratively to ensure successful sprint completion

Date _____ Accept/Reject work of each iteration Day _____

- Product Owner :
  - → An individual (or small group) whose job is to identify product feature or req/s
  - → Understand & prioritize the changing needs of end users
  - → Give team clear guidance on which feature to deliver next
  - → Bridge gap b/w what the business wants & what the team understands
  - → Continuously review product backlog to ensure that project meets req/
  - → Can be a customer or product manager.

- Scrum Master: → protect dev team from external distraction
  - → He is responsible for ensuring that the Scrum process is followed and guides the team.
  - → Shedules the resources needed for each Sprint
  - → Facilitate other Sprint events & team meetings
  - → Solve external changes faced by team
  - → Tracks backlog work to be done

▷ Scrum Events :

- Sprint Planning:
  - → The team estimates the work to be completed in the next sprint.

Date _____ Day _____

→ Members define Sprint goals
→ At end of meeting each member knows how each increment can be delivered in Sprint.

- Sprint:

→ A Sprint is the actual time period (2-4 weeks) when the Scrum Team works together to finish an increment.
→ The more complex the work and the more uncertain, the shorter the sprint should be.

- Daily Scrum or Stand-up:

→ A standup is a short meeting in which team members check in & plan for the day.
→ they report on work completed & any challenges in meeting sprint goals.

- Sprint Review:

→ At the end of the Sprint, the team gets together for an informal session to review the work completed & show it to stakeholders.
→ The Product owner might also rework the Product Backlog based on the current Sprint.

Date _____ Day _____

- Sprint Retrospective:
  → The team comes together to document and discuss what worked & what didn't, during the Sprint.
  → Ideas generated are used to improve future Sprint

Scrum Artifacts:

- Product Backlog:
  → It is a dynamic list of features, req, enhancements and fixes that must be completed for proj success.
  → It is essentially a team's to-do list.
  → The product owner maintains & updates the list.

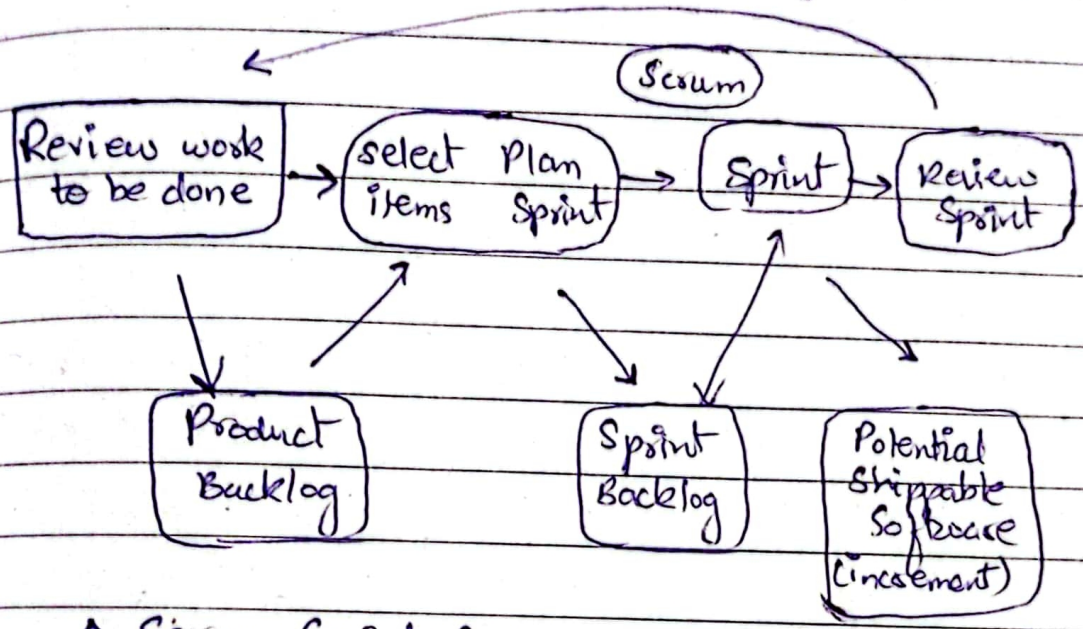
- Sprint Backlog:
  → List of items to be completed by the development team in the current Sprint cycle.
  → The items for sprint backlog is choosen from Product Backlog before each sprint.
  → Sprint backlog is flexible

- Increment:
  → It is a step towards goal or vision
  → It is the usable end product from sprint

Date _____ Day _____

• Velocity :

→ An estimate of how much product backlog effort that a team can cover in a single sprint.



▲ Scrum Sprint Cycle

▷ Agile Disadvantages :

→ Continuous changes and lack of clear boundaries can lead to uncontrolled scope expansion

→ Frequent meetings, reviews and feedback loops can consume a lot of time

→ Sucess depends on expertise and collab of team, making it challenging without skilled members

→ Less emphasis on documentation which can lead to misunderstanding in req.

→ Difficult to implement in large teams or across complex projects

Date _____ Day _____

-> Read last 20 slides of week #02

"WEEK # 04"

Types Of Requirements :

① Business req ② User Req ③ System Req

- **User Requirements:**
  → Statements in natural language plus diagrams of the services the system provides and its operational constraints.

- **System Requirements :**
  → A structured document setting out detailed descriptions of the systems functions, services and operational constraints.

Functional Requirements :

→ Describe functionality or system service
→ Func. req should describe the system services in detail.
→ Func. req must be:

Complete – Should include descriptions of all facilities req.

Consistent – Should be no conflicting descriptions

Date _____ Day _____

Non Functional Requirements:

→ These defines system properties & constraints
→ Eg: response time, scalability, performance
→ It may include mandating a particular IDE, programing langs. or dev method.
→ More imp than func. req.
→ These req may affect overall architecture of a system.

- Classification of Non-Func. Requirement:
  - Product Requirement: Req which specify that the delivered product must behave in a particular way. Eg: execution speed, reliability etc
  - Organisational Requirement: Req which are consequence of org. policies & procedures. Eg: process std used
  - External Requirements: Req which arise from factors which are external to the system & its dev process. Eg: legislative req.

The sys shall process 2000 transaction/sec with 99.97% uptime

- Verifiable NFR: one that can be measured, tested
  Eg: Performance, Security, Availability, Realibility
- Non-Verifiable NFR: that cant be measure
  Eg: A system should be fast
  System should be secure

Date _____ Day _____

▷ Metrics for specifying & verifying NFR:

① Speed → Processed transactions/sec, User responsetime
② Size → MBytes, No. of ROM chips, No. of CPUs/GPU
③ Ease of Use → Training time
④ Reliability → Mean time to failure, Probability of unavailability, Rate of failure occurence
⑤ Robustness → Time to restart after failure, % of events causing failure..
⑥ Portability → No. of target system (Android, IOS etc)

▷ User Story Format:

" As a waiter, I will be notified when an order has been complete "
the person who performs it — task

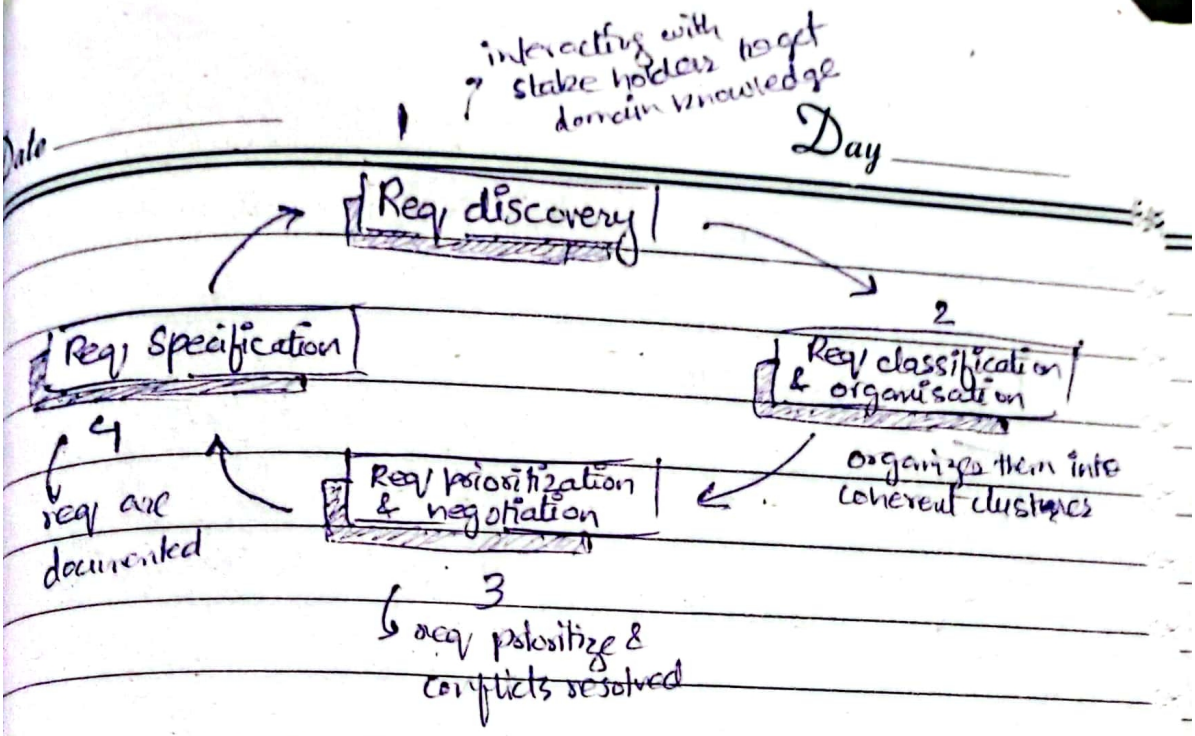
▷ Requirement Engineering:

" The process of finding out, analyzing, documenting and checking these services and constraints is called requirements engineering (RE)

• Requirement Elicitation / Discovery:

↳ Software engin. work with a range of system stake holder to find out about the application domain, the services that the sys should provide, the req sys performance etc.

Date _____ Day _____



- Types of Interviews for Req. Gathering:
  1. Closed Interview: based on pre-determined set of questions
  2. Open Interview: where various issues are explored with stakeholders

  ↳ Normally a mix of both is conducted

→ Interviews not good for understanding domain req.

- Ethnography:

→ It is a qualitative research method used to study ~~observe~~ people & culture in their natural enviornment.

See page # 48 - 51

Date _____ Day _____

- Ways Of Writing System Requirements :

(1) Natural Language :
↳ Req are written as natural lang sentences supplemented by diagrams & tables.
↳ Lack of clarity, FR & NFR tends to mix up

(2) Structured Specification :
↳ Req are written in std way limiting the freedom of writer

(3) Form based Specifications
↳ Definition of func entity
↳ Description of input/output
↳ Info about information needed for computation
↳ Descr. of the action to be taken
↳ Pre & Post conditions.
↳ Side effects (if any) of the func.
See page # 61 and further.