Page No.
Date

Q1. What is GitHub? Explain advantages & disadvantages

- GitHub:
- GitHub is a developer platform that allows developers to create, store, manage & share their code.
- GitHub is a web-based platform used for version control & collaboration on software development projects.
- It's widely used in the software development community for managing code, collaborating on projects & contributing on open-source software.

- Advantages of GitHub:

i) Easy Collaboration:

- GitHub makes it easy for multiple people to work on the same project at the same time, using features such as branching & pull request to manage changes & merge them into the main codebase.

ii) Version Control:

- GitHub allows users to keep track of changes to their code over time, making it easy to revert to previous versions if necessary.

iii) Widely used: GitHub is one of the most popular 'code hosting platform'

- Which means it is well-supported by a large community of developers who can provides help & guidance.

iv) Ease of use:

- GitHub has a user-friendly interface & provides a range of tools & features that make it easy to use, even for people who are new to Git & version control.

Page No.
Date

10. Disadvantages of GitHub:

i) Limited Storage & bandwith:
- GitHub offers free accounts with unlimited public repositories, but these repositories are limited to a maximum of 100MB in size.
- If you need more ~~so~~ storage or bandwidth, you'll have to pay for a premium plan.

ii) Lack of privacy:
- All public repositories on GitHub are, well, public. This means that anyone can see our code, which may not be ideal if we are working on a sensitive project.
- If we want to keep our code private, then we have to pay for premium plan.

iii) Collaboration limitations:
- While GitHub makes it easy for multiple people to collaborate on a project, it can difficult to manage large terms & complex workflow.

iv) Learning curve:
- GitHub can be challenging for individuals new to version control system & collaboration platforms. Concepts such as branching, merging, pull requests.

Page No.
Date

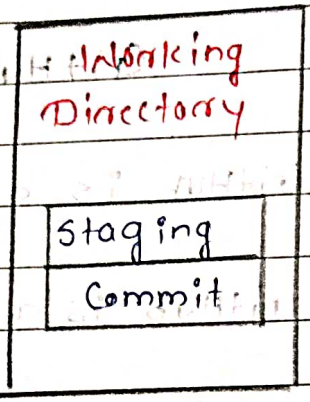
Difference between Git & GitHub.

Git	GitHub.
Git is a software	GitHub is a service
Git is a command-line tool	GitHub is a graphical user interface.
Git is installed locally on the system	GitHub is hosted on the Web
Git is maintained by linux	GitHub is maintained by microsoft.
Git is focused on version control & code sharing.	GitHub is focused on centralized source code hosting.
Git is a VCS to manage source code history	GitHub is a hosting service for Git repositories.
Git was first released in 2005	GitHub was launched in 2008
Git has no user management feature.	GitHub has a built-in user management feature.
Git is open-source licensed	GitHub includes a free-tier & pay-for-use tier.

Date

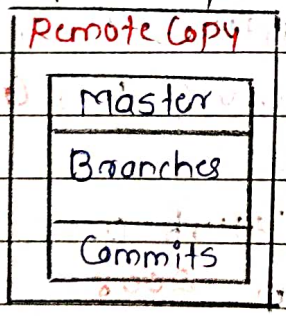
Q4 What local & remote repository?

- Local Repository (e.g our Computer)



- A local repository is essentially a database on our local machine that stores all the Files, Commit history, & other information related to our project.
- It resides directly on our Computer, typically within the directory of our project.
- We interact with local repository using version Cntrl Commands such as 'git add', 'git Commit', git Checkout

- Remote Repository (E.g GitHub)



- A remote repository is a repository that is hosted on a remote server, usually on platforms like GitHub, GitLab
- It serves central location where we can collaborate with others & back up our work.
- We can push our changes form our local repository to remote repository to share our work with others or keep it safely
- We can also pull ~~our~~ changes made by others From the remote repository to our local repository to incorporates their updates into our project.

Page No
Date

Q: What do u mean by staging area & how can you add file to working repository?

- staging area:
- The staging area, also known as the index, is a crucial Component of the Git version control system.
- It serves as an intermediate step bet^n the Working Directory (where we make changes to our files) & the Commit History (where we permanently record those changes)
- The staging area allows us to selectively choose which changes we want to include in our next Commit.

- To add file to working repository follow this step

i) Create or modify the file:
- First, we can create a new file or make modification to an existing file within project directory

ii) check the status:
- After creating or modifying the file, we can check the status of our repository to see which files have bee modified & which are untracked
- Using this Command: 'git status'.
- also this Command provides current state of Directory

iii) Add changes to staging area
- use the 'git add' Command to add specific changes or files to the staging area.

git add file1

iv) Commit changes to Repository:
- Once we have added the desired changes to the staging area, we can Commit them to the repository using this Command. 'git Commit'.

Ex: git Commit -m "Added new features".

Page No.
Date

Q6. Define the use of these Commands:

i) init:-
- Initializes a new Git repository in current directory.
- This Commands creates a hidden directory named '.git' which Contains all the neccessary files for version Control.
- 'git init'
- once the repository is initialized, the process of creating other files begins.

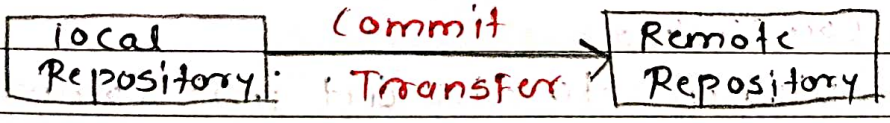
ii) Status:
- Shows the current status of the working directory & staging area, indicating which files are modified, staged, Untracked.
- This helps users understand what changes are ready to be Commited.
- This Command provides the current working branch
- If the files are in the staging area, but not Committed, it will be shown by the git status.
- Also, if there are no changes, it will show the message no changes to Commit, working directory clean
- We can see our directory status using 'git status' this Command.

iii) add:
- this Command Adds Changes from the working directory to the staging area.
- Before running the Commit Command, 'git add' is used to add any new or modified files.
- We can specify individual files by their names ('git add <Filename>')

or use '.' to add all changes 'git add .'
- This step preparcs changes for the next commit

iv) Commit:
- The Commit Command makes sure that the changes are saved to the local repository.
- The "-m" flag allows us to specify a commit message for describing the changes.
- Commits serve as snapshots of project history
- It is next Command after 'git add' Comimand
- Every Commit Contain index data & Commit message
- A Commit Command used For Fetch the update From staging arca to the repository.
- o syntax: 'git Commit -m "< Commit message >".

v) push:
- The push term refers to upload, a local Repository Content to remote repository.
- pushing is an act of transfer Commits From our local repository, to a remote repository

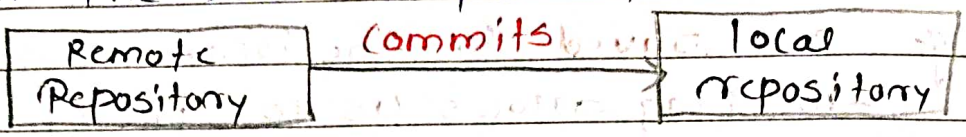


- pushing is capable of overwritting to changes
- This Command is used to share our changes with Collaborators.
- This is used to store the changes permanently into the Git repository.
- This is same as the 'Commit operation' in Subversion.
- Syntax: 'git push < remote 'branch >'.

Page No.
Date

VII) . pull :

- The pull command is used to access the changes (comm from a remote repository to a local repository.
- pulling updates our local repository with changes from the remote repository.

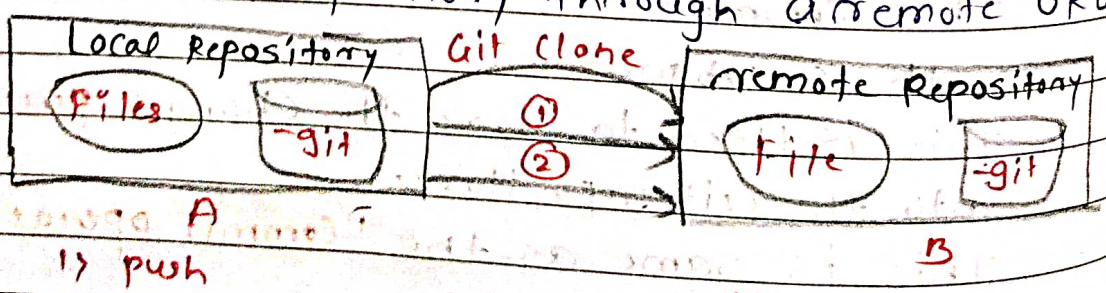


- It's essentially a combination of 'git Fetch' & 'git merge'
- The pull command Fetch & merge the changes remote server to our ~~local~~ working directory.
- It updates the local branches with the remote repository.
- This is same as 'update operation' in subversion

• Syntax: git pull <branch_name> <repository-URL>

VIII) clone :

- Creates a copy of a remote repository on our local machine
- The command downloads the remote repository to the Computer.
- It is equivalent to the Git init command when working with a remote repository.
- It access the repository through a remote URL



• Syntax: git clone <Repository URL>

9) log:
- The advantage of vcs is that it record changes
- These records allow us to retrive data like commits finding the bugs updates.
- so, the git log is a utility tool, which review & read a history of everything that happens to a repository.
- multiple option can be used with a git log to make history more specific.
- Generally the git log is a record of commits.
- syntax: git log
- This commands provides detailed history of changes

9) diff:
- shows the differences btn the working directory & the staging area or btn commits.
- We can specify a file to see changes only for that files. ~~This~~
- This command helps us ~~ref~~ review changes before committing them.
- Git diff is command line utility
- It is a multiuse git command
- It compare the different versions of data source
- The git log command also work as git diff command
- syntax: git diff.

10) branch:
- Creates lits or deletes branches in the repository
- We can create a new branch with a given name or list existing branches.
- The '-d' flag is used to delete a branch
- syntax:

Create a new branch
git branch <branch_name>

list all remote or local branches
git branch -a

delete branch
git branch -d <branch_name>

11) Checkout:

- The git checkout command is used to switch branch whenever the work is to be started on a different branch.
- This command works on 3 seperate entities: Files, Commits, & branches.
- We can specify either a branch name or commit hash.
- This command updates our working directory to reflect the chossen state.
- Syntax: # checkout an existing branch
  git checkout <branch_name>
  # checkout & create a new branch with that name
  git checkout -b <new-branch>

12) merge:

- The git merge command is used to integrate the branches together.
- The command combines the changes from one branch to another branch.
- It is used to merge the changes in the staging ~~area~~ branch to the stable branch
- Git automatically creates a merge commit if necessary
- Syntax: git merge <branch_name>

Credit Test 2.

Page No.
Date

Q1. What does the terms mean: knowledge acquisition refinement, distribution, development?

i) knowledge acquisition:

- knowledge acquisition refers to the process of gathering, collecting, or obtaining new information.
- knowledge acquisition is a process of knowledge management.
- knowledge acquistion may consists of facts, rules (concepts), procedure, heuristics, formulas, relationships, statistics or any other useful info.

Ex. Conducting research, attending training session, reading books or articles.

ii) Refinement:

- Refinement involves processing, organizing, synthesizing, or improving acquired knowledge to make it more useful, relevant or applicable.

OR

- Refinement refers to the process of improving or enhancing something by making small changes or adjustments to it.
- typically to make it more effective, useful.

Ex Analyzing Data, summarizing research findings

iii) Distribution:

- Distribution refers to the process of delivering or spreading goods, services, or information to a wide audience or across various locations channels or platforms.
- Sharing or making refined knowledge accessible to other.

Ex teaching new skills in a workshop or posting information online.

Page No.
Date

iv) Development:
- Applying knowledge to create new soln, products or improvement
- Development refers to process of growth

Ex: Designing a new product based on ~~real~~ research, or best practices in a project, or innovating to solve a problem.

Q2. What do you mean by knowledge sharing?
- knowledge sharing refers to the act of exchanging information, experiences among individuals or group within an organization or community.
- It involves transferring knowledge from one person to another, typically with the goal of improving understanding, collaboration, decision making

i) Transfer of Information:
- sharing knowledge involves communicating ideas, best practices, lesson learned

ii) Collaboration:
- It promotes collaboration & teamwork by enabling individuals to work together, & collectively solve problem

iii) learning & Development:
- knowledge sharing facilitates continuous learning & development by providing opportunities for individuals to acquire new skills, learn from other's experience & stay update on relevent info

iv) Organizational Culture:
- promoting a culture of knowledge sharing within an organization can enhance its overall effectiveness productivity & adaptability to change

Page No.
Date

Q3 Explain SCRUM framework.

- Scrum is an Agile framework used primarily in software development to manage complex project highlights iterative & incremental development, flexibility & collaboration
- Scrum concentrates specially on how to manage tasks within a team based development environment
- Scrum is derived from activity that occur during a ~~ruby~~ rugby match.
- Scrum teams work in short iteration or a time-boxed period called Sprint (typically 2-4 weeks).
- each Sprint has its goal.

i) Roles:

i) product Owner:
- Represents the stakeholders & is responsible for maximizing the value of the product by managing the product backlog.
- ensuring the team understand the requirements.

ii) Scrum master:
- Facilitates the scrum process, & ensure that the team follows to scrum practices & principles

iii) product Backlog:
- A prioritized list of all desired work on the project, maintained by the product owner.
- It represents requirements, enhancement & fixes needed to deliver a successful product.

iv) Sprint Backlog:
- It Contain the work that the devdopment team plans to complete during the sprint.

v) Sprint planning:
- A meeting held at the beginning of each Sprint where the team plans the work to be completed during the sprint.

vii) Daily scrum:
- A short (15-minute) daily meeting where the team synchronizes their activities discussed progress.

viii) Sprint Review:
- A meeting held at the end of the sprint where the development team demonstrates the completed work to stakeholder & receive Feedback.

viii) Sprint Retrospective:
- A meeting held at the end of the sprint where team reflects on their process for improvement & creates a plan for implementing changes in the next sprint.

Q4. Explain the roles of product owner & scrum master

i) Product owner:
- Represents the stakeholders, customers & users of the product.
- Defines the product vision & conveys it to the development team.
- Manages the product backlog, prioritizing items based on values & ensuring it is constantly refined & updated.
- Work closely with stakeholders to gather requirements, provide feedback & ensure alignment with business goal.
- make decision regarding the product feature, releases & overall direction based on input from stakeholders & market analysis.
- The product owner focuses on maximizing the value of the product & ensuring that it meets the needs & expectations of its users & stakeholders.

Scrum master:

11) Facilitates the Scrum process & ensure follows to Scrum principle, practices & rules.

- Acts as a servant leader to the development team, helping them understand & implement scrum practices effectively.
- Facilitates meeting such as sprint planning, daily scrum, sprint review & sprint retros-pective to ensure that they are focused & effective
- Coaches & mentors the team members to improve their understanding of scrum & Agile principle & practices.
- The Scrum Master focuses on supporting the development team creating an environment where they can work efficiently & effictively
- ultimately helping them to deliver high-quality increments of the project.

45. What is Daily Stand up, sprint, product backlog, sprint planning, sprint review, & retrospective:

- Daily stand up [illegible] / Daily scrum

  9 3 →

Page No.
Date

Q6. What is Requirement engineering:

- Requirement engineering is the process of defining, documenting & maintaining the Requirements.
- It is a process of gathering & defining services provide by the system.

i) Requirements elicitation:

- Gathering requirements from stakeholders through various techniques such as interviews, workshops., surveys & observation.
- The goal is to identify & understand the need & expectations of all stakeholders involved in the project.

ii) Analysis

- Analyzing & prioritizing requirements to ensure clarity, Consistency.
- This involves: Unclearity, inconsistency in the requirements & resolving them through negotiation & Collaboration with ~~state~~ stakeholder.

iii) Documentation:

- Documenting requirements in a clear & structured manner.
- Documentation serves as a Communication tool bet^n stakeholder & provides a basis for validation & verification of the Final solution.

iv) Validation & verification:

- Validating requirements to ensure that ~~they~~ they accurately represent the need & expectation of stakeholders & that they can be satisfied by proposed solution.
- In validation Conducting review from stakeholder as per requirement & then verifying that proposed sol^n aligns with requirements.

v) Management:
- Managing requirements throughout the project lifecycle, including tracking changes
- Effective requirement management ensures that project stay on track & delivers the desired outcomes.

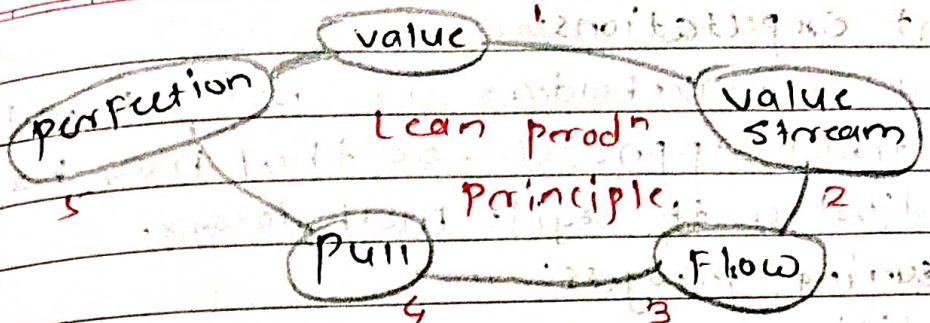
vi) Requirement Specification:
- Requirement specification in requirement engineering refers to the detailed documentation of the system Functional & non Functional requirements.
- specification serve as a blueprint for the development team, guiding them in building the system or product according to the stakeholder's expectation.

q8. Explain the term requirement prioritization
- once Requirement are identified, prioritization involves determining the importance or urgency of each requirements.
- This is crucial because it helps in allocating resource efficiently & ensure that the most critical needs are addressed first.
- prioritization based on factors such as bwines value, ~~Saketol~~ stakeholder input.

Q 9. What is lean production

- Lean production is an approach to management that focuses on cutting out waste & ensuring quality
- This approach can be applied to all the aspects of a business - from design, through production, to distribution
- The lean production aims to cut cost by making the business more efficient & ~~rep~~ responsive to market needs.
- Lean production is a part of agile methodology which helps to minimize waste & maximize the efficiency & continuous improvement.
- It refers to delivering values to the customer through iterative development cycles.
- Lean principle are used to enhance the productivity & streamline workflow.
- unnecessary things are cut out in lean product that is also called as cut out a waste.
- The lean production is originated in the manifacturing plants of Japan, but now they has been adopted well for large manifacturing activities.
- Lean production simply means:
  - (i) Doing simple thing
  - (ii) Doing thing better
  - (iii) To involve empolyee continuously in the improvement process
  - (iv) eliminate the unwanted waste
- Key aspects of lean production are:

i) Time based management
ii) Just in time production
iii) quality improvement & management
iv) Cell production.



Q6 List out the challenges faced while migrating to agile methodologies ? Explain

- Agile methodology is an iterative approach to software development.
- It provides flexibility, collaboration & customer satisfaction.
- Agile follows iterative lifecycles which is known as sprints.
- Some challenges are faced while migrating to agile methodology:

i) Resistance to change:
- Teams & stakeholders may resist to adopt agile because of its extra features & they have to leave traditional methods behind.

ii) Lack of understanding:
- Sometimes agile principles may create confusion which can cause misunderstanding among team mates

iii) Organization culture:
- Agile method acquires a culture shift towards collaboration, transperancy & adaptability & which don't be lie in organizations culture.

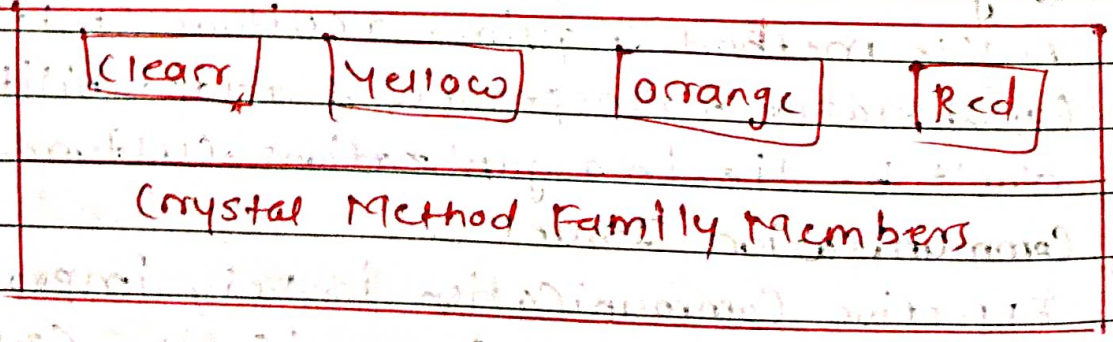
iv) Communication Issue:
- Effective communication is very important in agile teams & inadequate communication cause issue.

vi) Client expectations:
- Client & stakeholders are being familier with traditional approach, so they may have difficulty to adjust with agile methodology.

vii) Measuring progess:
- Traditional metrics may not align with agile values. It makes difficult to measure a progress accurately.

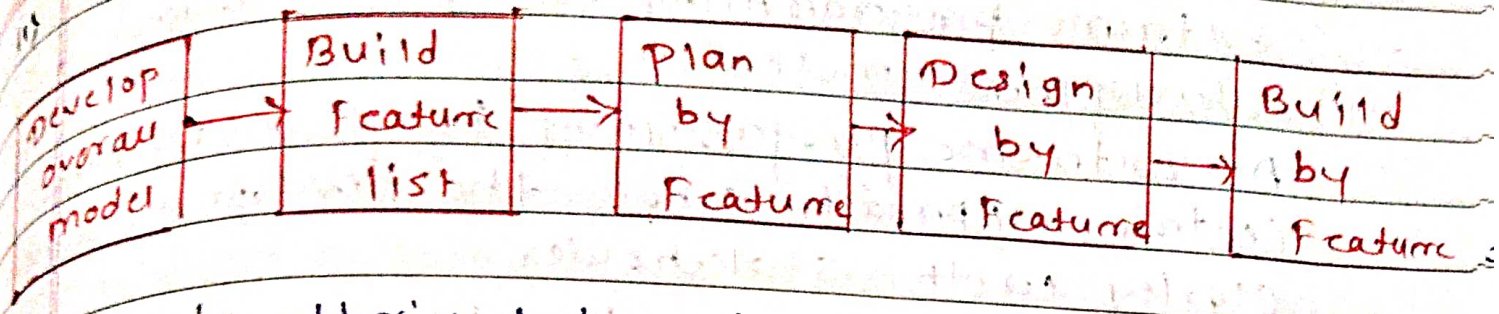
Q10. Explain the Crystal Feature, Driven development Adaptive Software development & extreem programming
- All these are software development methodologies
- These are used in agile methodology & agile project management.

i) Crystal methodologies:
+ Crystal methodology are a family of agile methodologies.
- Crystal methodology was developed by Alistair Cockburn
- They focus on people, interaction, Community skill talents & Communication.
- All these are considered as a key component of Software development to make it more successful
- Crystal methodologies have various Flavours
eg. Crystal clear, Crystal orange, Crystal Yellow for diff crend, project & environment.



Page No.
Date

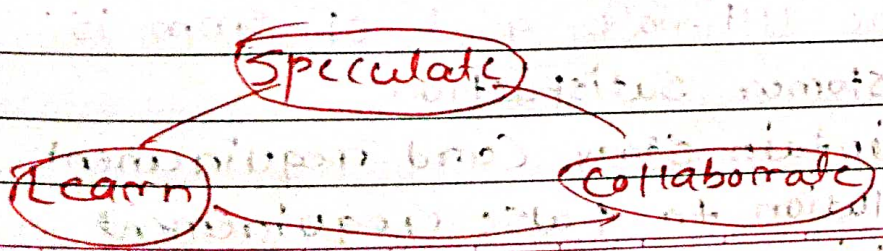
Feature Driven Development:



- FDD is abbrivated as 'Feature driven development'
- It is an iterative & incremental software development methodology.
- FDD focus on building the feature incrementaly
- It involve five basic activities as:-
  (i) Develop overall model (ii) Build Feature list
  (iii) plan by feature (iv) Design by Feature
  (v) Build by Feature.
- FDD provides structured approach to manage development activities.
- It is particularly used for large teams & complex projects.

Adaptive Software Development:

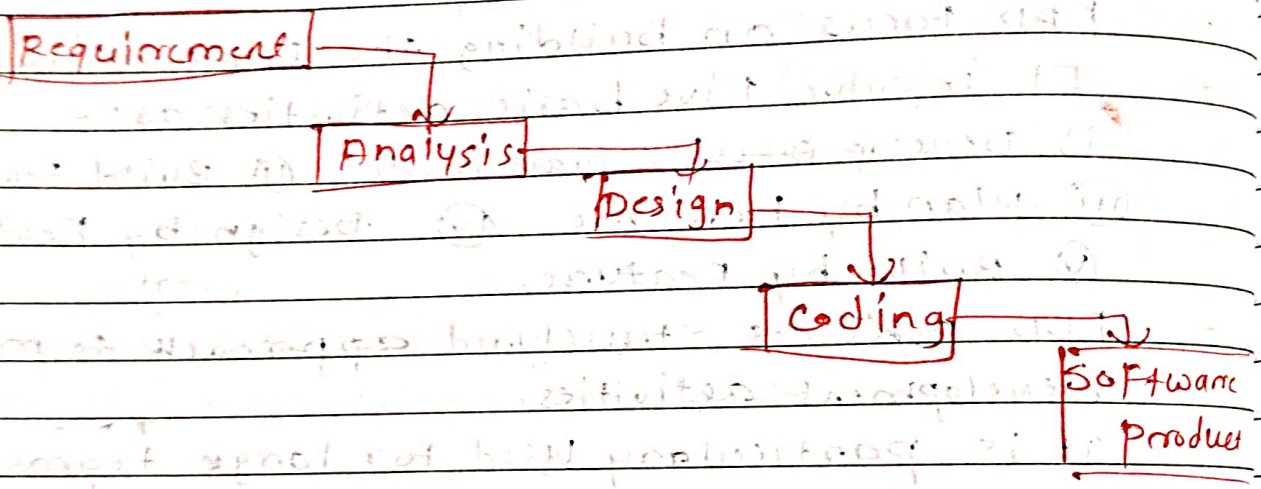
- ASD (Adaptive Software Development) is agile methodology.
- It was developed by Jim Highsmith
- ASD Focus on collaboration self-organization & flexibility which respond to change.
- ASD allows teams for continuously adjust an approach based an feedback changing requirment
- Three phase of ASD
  i) Speculation ii) Collaboration iii) Learning



Date

iv) Extreme programming:

- Extreme programming is agile software development methodology
- An Extreme programming focus on customer centric approach where the developers work closely with stakeholder
- Extreme programming focuses on giving a high quality software delivery



Q 11. What is Story-Card Maturity Model (SCMM)

- "SMM" means a Story-Card maturity model.
- The SMM model designed to improve & enhance the agile SOFT development methodology
- The Smm helps to boost up the agile requirement principle & objective - such as : lower Cost, Customer Satisfaction, requirements quality
- In Story-Card maturity model each level has a pre-defined goal for helping organization to focus on their improvement activities

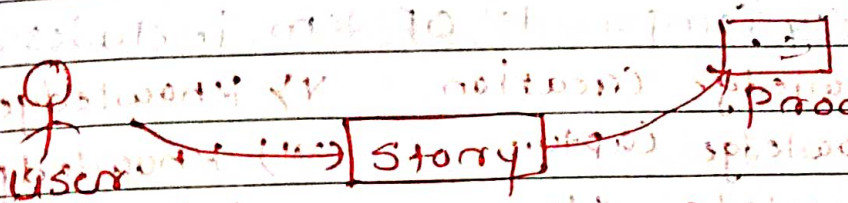
• The Ultimate goal of Smm is:

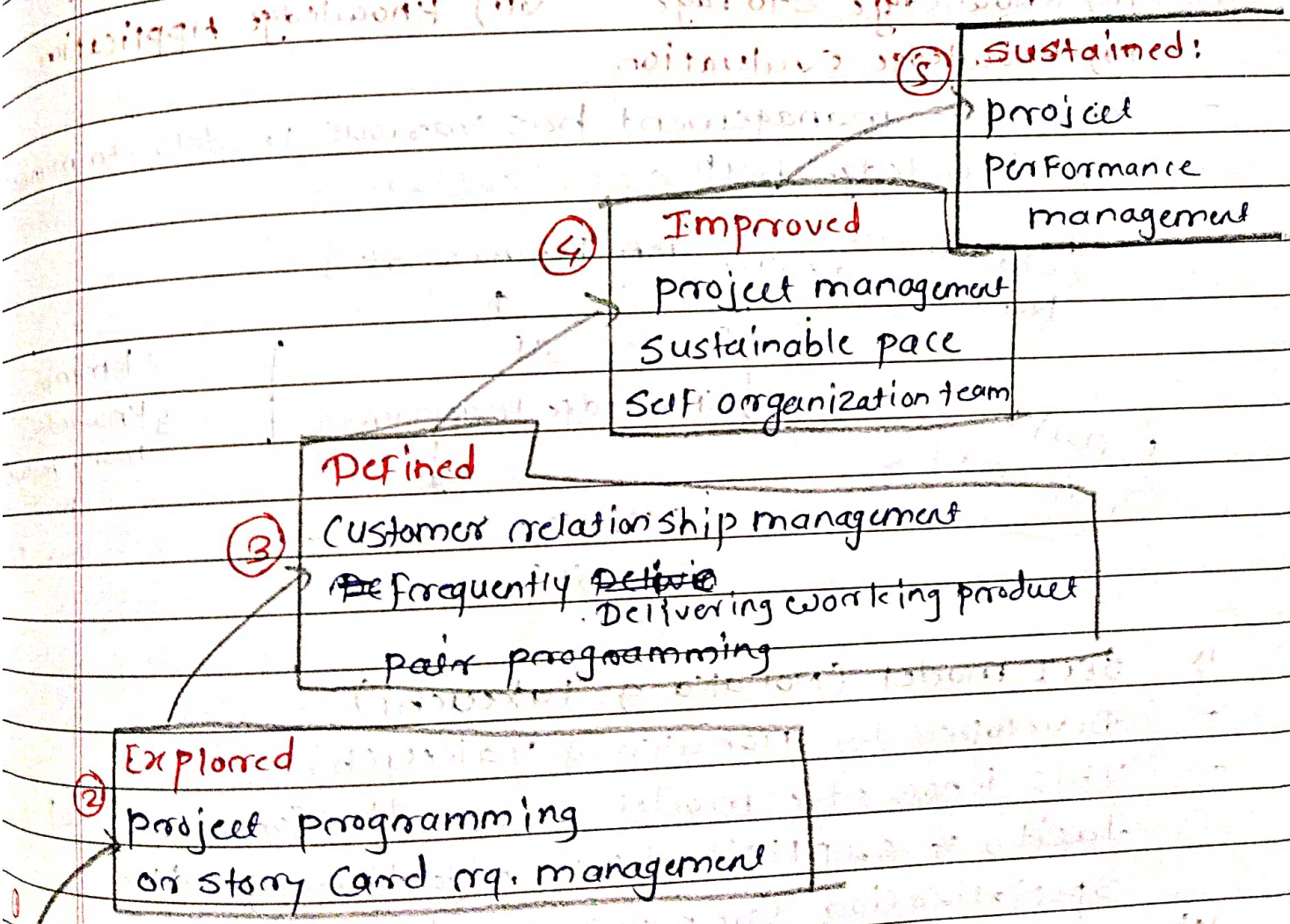
i) Customer Satisfaction
ii) maintain Story Card requirement changes
iii) Solution to False requirement
iv) Obtain understanding from User Story on story Card

Page No.
Date

vi) obtain Commitment to user story
vii) maintain tracability of requirement
viii) Identify the inconsistancy betn project work & user story.
viiii) manage requirement stories.
ix) Develop a project based on the story-cards.

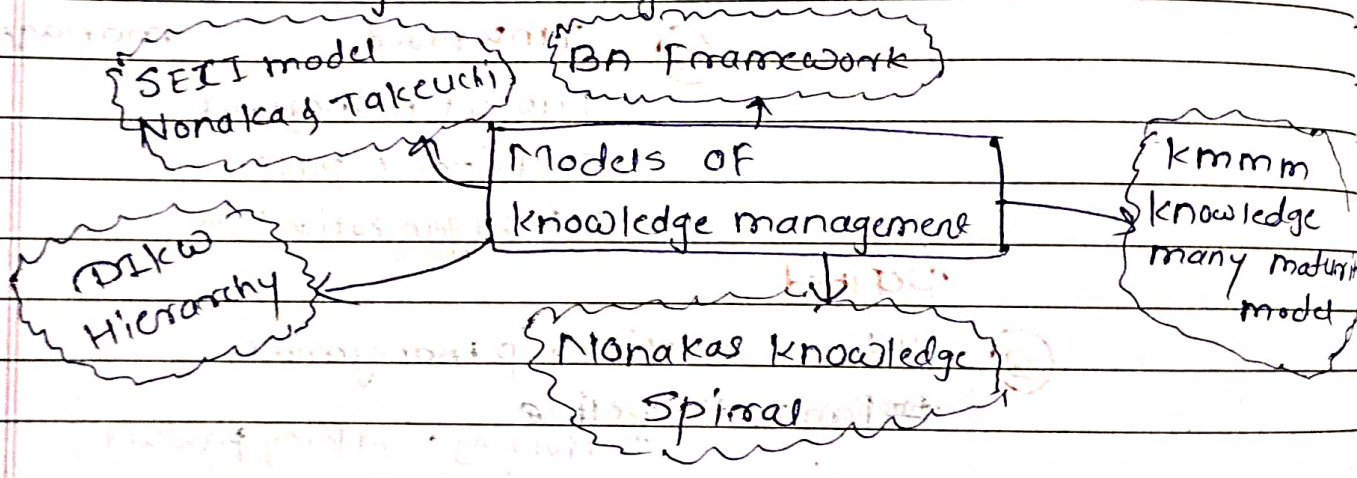
Story Card:





Q12) What are the different modes of knowledge management?

- Knowledge management:
- km is abbriviated as 'knowledge management'
- It is a special approach to identify, capture, store, retriving & sharing of knowledge.
- knowledge management involves systematic approach for managing both explicit & tacit knowledge assets
- The key components of km includes:
  1) knowledge creation
  2) knowledge capture
  3) knowledge storage
  4) knowledge evaluation
  5) knowledge sharing
  6) knowledge transfer
  7) knowledge application
- knowledge management has various models to manage the knowledge bet^{n} organization.



1) SECI model (Nonaka & Takeuchi)
- Developed by Nonaka & Takeuchi
- This knowledge model ensure the conversion of tacit, & explicit knowledge through the Specialization Combination etc

2) DIKW Hierarchy:
- Represents the relationship between data, information knowledge & so on.
- It illustrate how data is transformed

Nonaka's knowledge spiral

(iii) It describe the dynamic process of a knowledge creation transformation & utilization within organization.

(iv) knowledge mathagement maturity model

- kmmm means knowledge management maturity model.
- kmmm provides a framework for maturity level across various dimension.

v) BA Framework:

- BA Framework introduced by Nonaka & konna.
- It describe the conceptual space where knowledge sharing & creation occur.

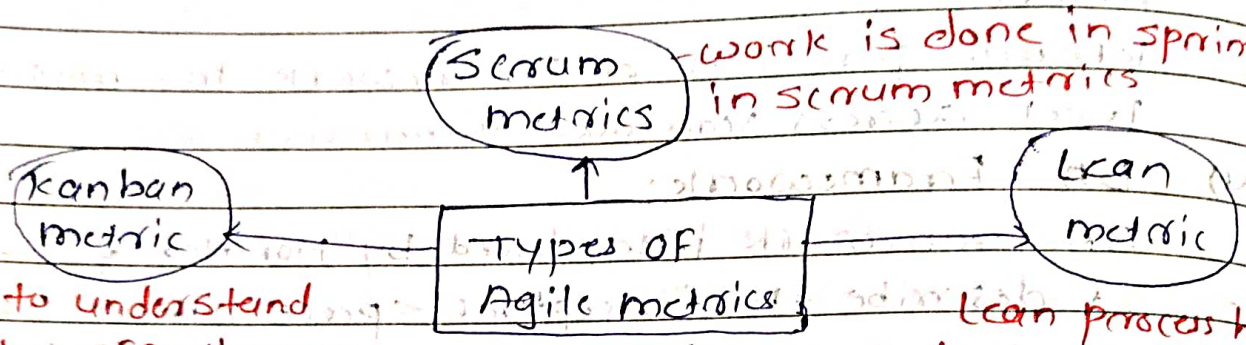
Q13. Explain the agile metrics in details with diagram.

- Agile metrics is a tool that helps marketing teams to measure the progress & productivity of marketing activities.
- It track the activities of marketing teams
- Agile metrics are most effective when they are used for the specific needs of individual projects.
- We can use agile metrics at both team level & individual level.
- At team level they help to access the overall market activities
- At individual level they help to identify areas of improvement for each team member.

• Importance of Agile metrics:

- Agile metrics are important because they helps to track progress & identify areas for improvemat
- Increase productivity by providing visuals of project.

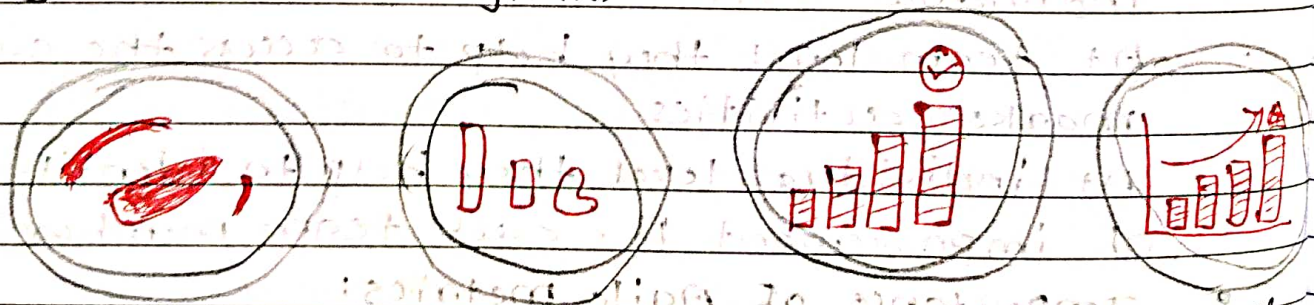
- Build transperancy btn stakeholders because everyone knowns what their team & stakeholder expect
- Improve Communication btn team members
- It helps to manager & leaders to identify risks & potential from data.



- Agile metrics should be

1) Clearly defined
2) Evaluated together
3) Role-specific
4) Actionable
5) Easy to measure
6) Workflow aligned

- Agile metrics Diagram:

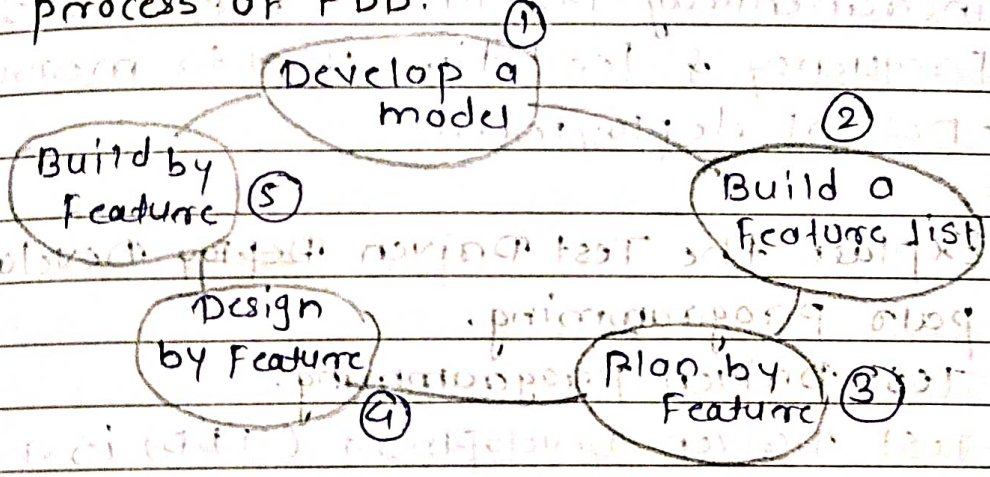


Productivity metrics | progress metrics | Quality metrics | overall performance of teams

Page No.
Date

Q14. Explain the feature driven development (FDD) with financial & production metrics in FDD

- FDD is abbrivated as Feature driven development
- FDD is an iterative & incremental software development methodology.
- FDD focus on delivering the feature incrementally
- It focuses on building a software around specific features or functionalities.
- The feature driven development works on design & domain modeling.
- Feature driven development (FDD)
- For process of FDD.



- Now let's know how FDD is applied with a financial & production metric

i) Feature Identification & prioritization:

- In FDD the feature are prioritized based on their importance to stakeholders. Financial metrics used to prioritorize feature.

ii) Feature Teams & work breakdown:

- FDD works on a specific feature. each feature team is responsible for designing, implementing & testing assigned feature production metrics can be used to identify productivity.

iii) Iterative Development & Feedback:
- FDD promotes iterative development life cycle where feature are developed incrementally & time to time feedback is taken from stakeholder Production metric used to identify improvements.

iv) Quality Assurance & Testing:
- FDD provides early & Continuous testing throughout development process Financial metrics can be used for quality assurance.

v) Deployment:
- FDD focus on delivering working software incrementally production metrics such as Frequency & lead time used to measure effective-ness of deployment.

Q15. Explain the Test Driven ~~Deploy~~ Development & pair programming.

1) Test Driven Programming:
- Test Driven Development (TDD) is a software development approach, where tasks are written before the code
- The development process consists of three steps;
  (i) Writing a Failing test
  (ii) Writing a code to pass the test
  (iii) Refactoring the code.
- This process cycle is repeated iteratively
- The aim of TDD is to create a well tested & reliable code.
- The TDD process follows following step:
  (1) Write a test: In this initial phase developers write, the ~~minimum amount of code.~~ automated test that define the desired behaviour of a piece of code.

Page No.
Date

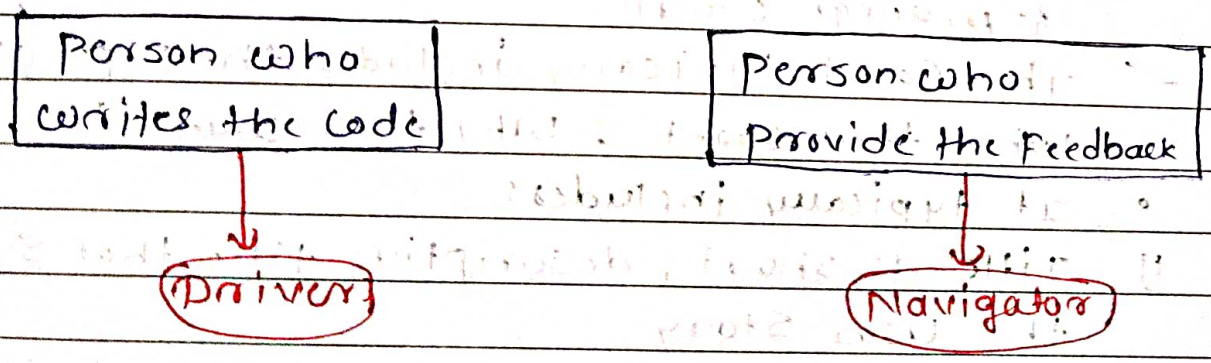
(1) Write a code:
- After writing failing test developers write the minimum amount of code.

(11) Refactor the code:
- once the test processes developers refactor the code to improve its structure performance & readability

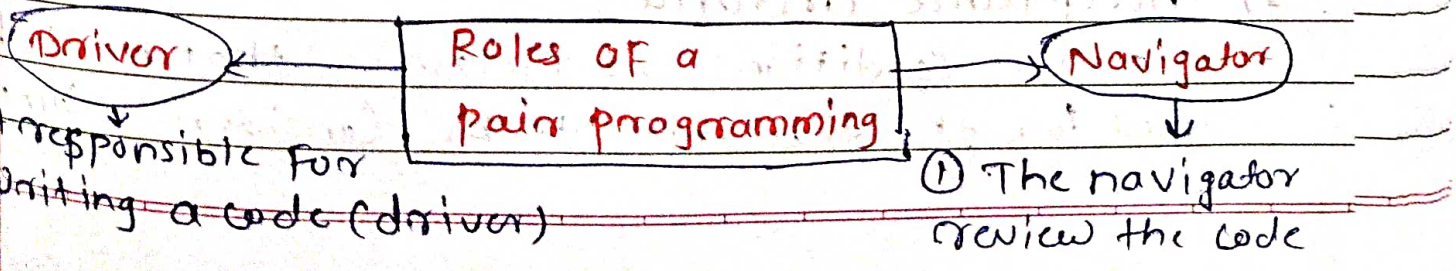
(111) Repeat:
- Developers iterate through this cycle continuously

- Benifit of TDD:
① Improve Code Quality ③ Increased Confidence
② Early Detection of Bugs

2) pair Programming:
- pair programming is a collaborative technique
- In pair programming two programmers work together at one Computer.
- one person Write the code while other person review each line & provide the Feedback.



- In pair programming Concept roles can switch frequently & higher quality of code is there
- The two main roles of a pair programming are:



2) Driver responsible for typing a keyboard & implementing the soln to the problem

2) The navigator perform tasks like - providing Feedback suggesting improvements & Considering alternative approach

(3) Navigator Focus on big picture as project goal.

- Benefits of pair programming:

(1) Improve Code Quality (2) knowledge sharing
(3) Reduced risk (4) Increased Focus

Q What is the use of Story Card? Template of User Story Card.

- A Story card, often used in Agile development methodologies like ~~Scrum~~ scrum, is a Concise ~~doc~~ document used to capture the requirements & details of a user ~~satory~~ story
- ~~Aa~~ is a tool used in agile marketing to plan & manage work.
- The Card typically includes the type of User What they want & Why they want it.
- It typically includes:

1) Title: A short, descriptive title that summarizes the User Story
2) Story: A brief narrative that describe the desired Functionality or Feature From the User's perspective
3) Acceptance criteria:
- specific Conditions or criteria that must be met for the User Story to be Considered Complete

iv) priority:
- The relative importance or urgency of the user story compared to others.

v) Estimate:
- An estimate of the effort required to implement the user story, often measured in story point or another unit of estimation.

- The template for a user story card can vary depending on the team's pre