[6m]

Q2) a) A large construction company engaged in real estate construction business decided to develop ERP through Astha softech. The output of system will be cost sheets detailing the relevant information for contracting, budgeting, progress monitoring and bill payment. Astha softech team has no domain knowledge. As a project manager, you have been asked to suggest risk management strategy after identifying the risk.

Answer-->

Lack of domain knowledge:

Risk: Astha Softech's team may struggle to understand the intricacies of the construction and real estate industry, leading to inaccurate or incomplete implementation of the ERP system.

Risk Management Strategy: Conduct thorough knowledge transfer sessions where the construction company's domain experts provide detailed information and insights about their business processes. Regularly review and validate the work done by Astha Softech to ensure accuracy and relevance.

Inadequate project planning:

Risk: Insufficient planning may lead to delays, budget overruns, or unsatisfactory system performance.

Risk Management Strategy: Develop a comprehensive project plan that includes all the necessary activities, milestones, timelines, and resource allocations. Regularly review and update

the plan as the project progresses, and ensure effective communication and coordination between the construction company and Astha Softech.

Integration challenges:
Risk: The ERP system may face difficulties in integrating with existing systems or data sources within the construction company's infrastructure.

Risk Management Strategy:
Conduct a thorough analysis of existing systems and data sources to identify potential integration challenges early on. Engage technical experts from both Astha Softech and the construction company to design and implement effective integration solutions. Perform rigorous testing and validation to ensure seamless data flow and system interoperability.

Data security and privacy:
Risk: The ERP system may be vulnerable to data breaches or unauthorized access, compromising sensitive information such as financial data or customer details.

User adoption and training:

Risk: Users within the construction company may struggle to adapt to the new ERP system, leading to low user adoption rates and reduced productivity.

Risk Management Strategy:

Develop a comprehensive user adoption and training plan. Conduct user training sessions to familiarize employees with the new system, its features, and benefits. Provide ongoing support and assistance to address any user concerns or issues. Encourage user feedback and incorporate necessary improvements based on user experiences.

Vendor reliability:
Risk: Astha Softech may face financial or operational challenges during the project, affecting their ability to deliver the ERP system as planned.

Risk Management Strategy:
Conduct due diligence on Astha Softech's financial stability and track record. Define clear contractual terms and milestones, including penalties for non-compliance. Maintain regular communication and project oversight to identify any warning signs early on. Have contingency plans in place to mitigate the impact of any potential vendor-related issues.

b) Explain the benefits of Agile project management in brief.

Answer-->The 9 Key Benefits of Using the Agile Methodology [4m]

Agile today stands as one of the most popular approaches to project management because of its flexibility and evolutionary nature. It started in 2001 with the Agile manifesto and was originally made for software development.

Over time, agile project management evolved and became a popular choice for many project managers, irrespective of the industry.

Here are some top reasons and benefits of Agile and why it is adopted by top companies for managing their projects:

1. Superior quality product

In Agile project management, testing is an integrated part of the project execution phase which means that the overall quality of the final product is greater. The client remains involved in the development process and can ask for changes depending on the market realities. Since Agile is an iterative process, self-organizing teams keep on learning and growing with time and continue improving.

2. Customer satisfaction

In the Agile, the customer is always involved in the decision-making process which leads to greater customer retention. In the traditional framework, the customer is only involved in the planning phase and does not influence execution which affects the flexibility and adaptability. By keeping the customer in the loop and making changes according to their feedback, you deliver value to the customer and ensure that the final product is truly according to their requirements.

.

3. Better control

Agile allows managers to have better control over the project due to its transparency, feedback integration, and quality-control features. Quality is ensured throughout the implementation phase of the project and all stakeholders are involved in the process with daily progress reports through advanced reporting tools and techniques.

4. Improved project predictability

With increased visibility, predicting risks, and coming up with effective mitigation plans becomes easier. Within the Agile framework, there are greater ways to identify and predict risks and plan to ensure that the project runs smoothly.

Scrum methodology, for example, uses sprint backlogs and burndown charts to increase the visibility of the project which allows managers to predict performances and plan accordingly.

5. Reduced risks

In theory, any project using an Agile methodology will never fail. Agile works in small sprints that focus on continuous delivery. There is always a small part that can be salvaged and used in the future even if a particular approach doesn't go as planned.

6. Increased flexibility

When Agile is truly implemented in a project team, it empowers them with unparalleled flexibility. Teams work in smaller bursts and are supplemented by the constant feedback and involvement of the product owner. In other project management methodologies, changes usually are time-consuming and costly.

However, Agile divides the project in short sprints that are both manageable and flexible enough to allow the team to implement changes on short notice. This unmatched flexibility is one of the top reasons why dynamic organizations prefer to use Agile in their project.

7. Continuous improvement

Working on self-reflection and striving for continuous improvement is one of the 12 core principles of the Agile manifesto. The methodology works in iterations which means that each sprint will be better than the last one and previous mistakes will not be repeated. Agile methodologies foster an open culture of idea exchange and collaboration which allows team members to learn from shared experiences and improve together.

b) Explain sprint retrospective in detail.[4 marks]

-->As described in the Scrum Guide, the purpose of the Sprint Retrospective is to plan ways to increase quality and effectiveness.

The Scrum Team inspects how the last Sprint went with regards to individuals, interactions, processes, tools, and their Definition of Done. Inspected elements often vary with the domain of work. Assumptions that led them astray are identified and their origins explored. The Scrum Team

discusses what went well during the Sprint, what problems it encountered, and how those problems were (or were not) solved.

The Scrum Team identifies the most helpful changes to improve its effectiveness. The most impactful improvements are addressed as soon as possible. They may even be added to the Sprint Backlog for the next Sprint.

The Sprint Retrospective concludes the Sprint. It is timeboxed to a maximum of three hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

During the Sprint Retrospective, the team discusses:

- What went well in the Sprint
- What could be improved
- What will we commit to improve in the next Sprint

The Scrum Master encourages the rest of the Scrum Team to improve its process and practices to make it more effective and enjoyable for the next Sprint. During each Sprint Retrospective, the Scrum Team plans ways to increase product quality by improving work processes or adapting the definition of "Done" if appropriate and not in conflict with product or organizational standards.

By the end of the Sprint Retrospective, the Scrum Team should have identified improvements that it will implement in the next Sprint. Implementing these improvements in the next Sprint is the adaptation to the inspection of the Scrum Team itself. Although improvements may be implemented at any time, the Sprint Retrospective provides a formal opportunity to focus on inspection and adaptation.

Search all Resources related to Sprint Retrospectives.

The Sprint Retrospective

The Sprint Retrospective is the last event in the Sprint. Unlike other Scrum Events where the focus is on inspecting and adapting ways to improve the product, the Sprint Retrospective is a place for the Scrum Team to inspect and adapt their working practices.

Q3) a) A project of 200 KLOC is to be developed. Software development team has average experience on similar type of projects. The project schedule is not very right. Calculate the effort, development time, average staff size of the project by using semi-detached mode of cocomo model.[6]

-->Example2: A project size of 200 KLOC is to be developed. Software development team has average experience on similar type of projects. The project schedule is not very tight. Calculate the Effort, development time, average staff size, and productivity of the project.

Solution: The semidetached mode is the most appropriate mode, keeping in view the size, schedule and experience of development time.

Hence E=3.0(200)1.12=1133.12PM

D=2.5(1133.12)0.35=29.3PM

$$\text{Average Staff Size (SS)} = \frac{E}{D} \text{ Persons}$$

$$= \frac{1133.12}{29.3} = 38.67 \text{ Persons}$$

$$\text{Productivity} = \frac{KLOC}{E} = \frac{200}{1133.12} = 0.1765 \text{ KLOC/PM}$$

P = 176 LOC/PM

b) Differentiate : Agile project management v/s Traditional project management [4 mark]

-->The table down below shows the major differences between the traditional and agile project methodology.

Characteristics	**Agile approach**	**Traditional approach**
Organizational structure	Iterative	Linear
Scale of projects	Small and medium scale	Large-scale
User requirements	Interactive input	Clearly defined before implementation
Involvement of clients	High	Low
Development model	Evolutionary delivery	Life cycle

Customer involvement	Customers are involved from the time work is being performed	Customers get involved early in the project but not once the execution has started
Escalation management	When problems occur, the entire team works together to resolve it	Escalation to managers when problem arise
Model preference	Agile model favors adaption	Traditional model favors anticipation
Product or process	Less focus on formal and directive processes	More serious about processes than the product
Test documentation	Tests are planned one sprint at a time	Comprehensive test planning
Effort estimation	Scrum master facilitates and the team does the estimation	Project manager provides estimates and gets approval from PO for the entire project
Reviews and approvals	Reviews are done after each iteration	Excessive reviews and approvals by leaders

b) Explain Agile project management life cycle.[4 mark]

-->What is Agile project management (APM)? Agile project management (APM) is an iterative approach to planning and guiding project processes. It breaks project processes down into smaller cycles called sprints, or iterations

Phases of Agile Model:

1. Requirements Gathering: The customer's requirements for the software are gathered and prioritized.

2. Planning: The development team creates a plan for delivering the software, including the features that will be delivered in each iteration.

3. Development: The development team works to build the software, using frequent and rapid iterations.

4. Testing: The software is thoroughly tested to ensure that it meets the customer's requirements and is of high quality.

5. Deployment: The software is deployed and put into use.

6. Maintenance: The software is maintained to ensure that it continues to meet the customer's needs and expectations.

Example: Let’s go through an example to understand clearly how agile actually works. A Software company named ABC wants to make a new web browser for the latest release of its operating system. The deadline for the task is 10 months. The company’s head assigned two teams named Team A and Team B for this task. In order to motivate the teams, the company head says that the first team to develop the browser would be given a salary hike and a one-week full-sponsored travel plan. With the dreams of their wild travel fantasies, the two teams set out on the journey of the web browser. Team A decided to play by the book and decided to choose the Waterfall model for the development. Team B after a heavy discussion decided to take a leap of faith and choose Agile as their development model.

The Development plan of the Team A is as follows: Requirement analysis and Gathering – 1.5 Months Design of System – 2 Months Coding phase – 4 Months System Integration and Testing – 2 Months User Acceptance Testing – 5 Weeks

The Development plan for the Team B is as follows:

 Since this was an Agile, the project was broken up into several iterations. The iterations are all of the same time duration. At the end of each iteration, a working product with a new feature has to be delivered. Instead of Spending 1.5 months on requirements gathering, They will decide the core features that are required in the product and decide which of these features can be developed in the

first iteration. Any remaining features that cannot be delivered in the first iteration will be delivered in the next subsequent iteration, based on the priority At the end of the first iterations, the team will deliver working software with the core basic features.

Q4) a) Demonstrate value-driven development with suitable example

-->[6 mqrks]

Explain the roles of scrum master, product owner and development team.[6 marks]

scrum master:

The scrum master's role in supporting product owners in the following aspects:

- Find methods to effectively manage the product backlog.
- Help communicate the owner’s wishlist to the project team.
- Arrange and optimize product backlog.
- Organize scrum events as necessary.

product owner:

One of the main roles of a Product Owner is to manage the product backlog. This may include the following activities:

- The product backlog must be clearly defined, and all the items need to be mentioned elaborately.
- Prioritize and order the product backlog in the right manner so that the important tasks are given topmost priority.
- Prioritize work items and product backlog, this must be in line with customer vision and goals.
- Evaluate the work done by the development team and provide constant feedback.
- The Product Owner must ensure that the product backlog is communicated clearly to all team members.
- The Scrum Team must have clarity on the product requirements and user expectations.

development team:

The development team leader role is also known as the technical team leader. Development team leaders are very experienced developers with extensive subject matter expertise. You need tech leads with skills relevant to your project.

Development team leaders have the following responsibilities:

- Analyzing the project requirements;
- Offering inputs to architects on the technical solutions;
- Explaining the business and technical requirements to other team members;
- Guiding other developers and providing technical support to them;
- Coding;
- Reviewing;
- Unit testing;

- Collaborating with UI designers, quality assurance engineers, and DevOps engineers;
- Troubleshooting technical and other issues that impact the team;
- Helping the PM and architect to mitigate and reduce project risks;
- Providing the project status to the PM;
- Communicating with the key stakeholders;
- Leading continuous improvement efforts in the development team.

b) Write short note (any one). [4] i) Agile tools. ii) GitHub.

i) Agile tools.:

In agile development, leading as project management is not the easiest job. Jumping between your daily scrums to your next sprint, it causes hard to focus on the work. The agile development tools fulfill your needs, and does it for you.

Why do teams need agile tools?

Agile is the most pervasive method used by software development teams the world over. Ask many development teams what it means and they'll probably attempt to pin it down in terms of the tools used and the different ways of implementing it, such as kanban boards, daily standups and scrum meetings. While it's true that these tools and techniques are used in agile, no single one can define the methodology as a whole.

Agile is really a way of thinking and working. It's one way out of many that a team can organize itself when approaching certain tasks and projects. Agile is based on harnessing adaptability,

flexibility and clear communication. This helps teams face the chaotic and exciting world of product development to deliver cost-effective, user-friendly products.

To achieve this, it forces teams to make room for regular client and end-user feedback in habitual testing and unlimited iteration. Instead of setting everything in stone in a rigid timeline, the work is instead planned out in sprints that tackle a work backlog. It often deploys the fail fast philosophy and allows for pivoting and changes in direction.

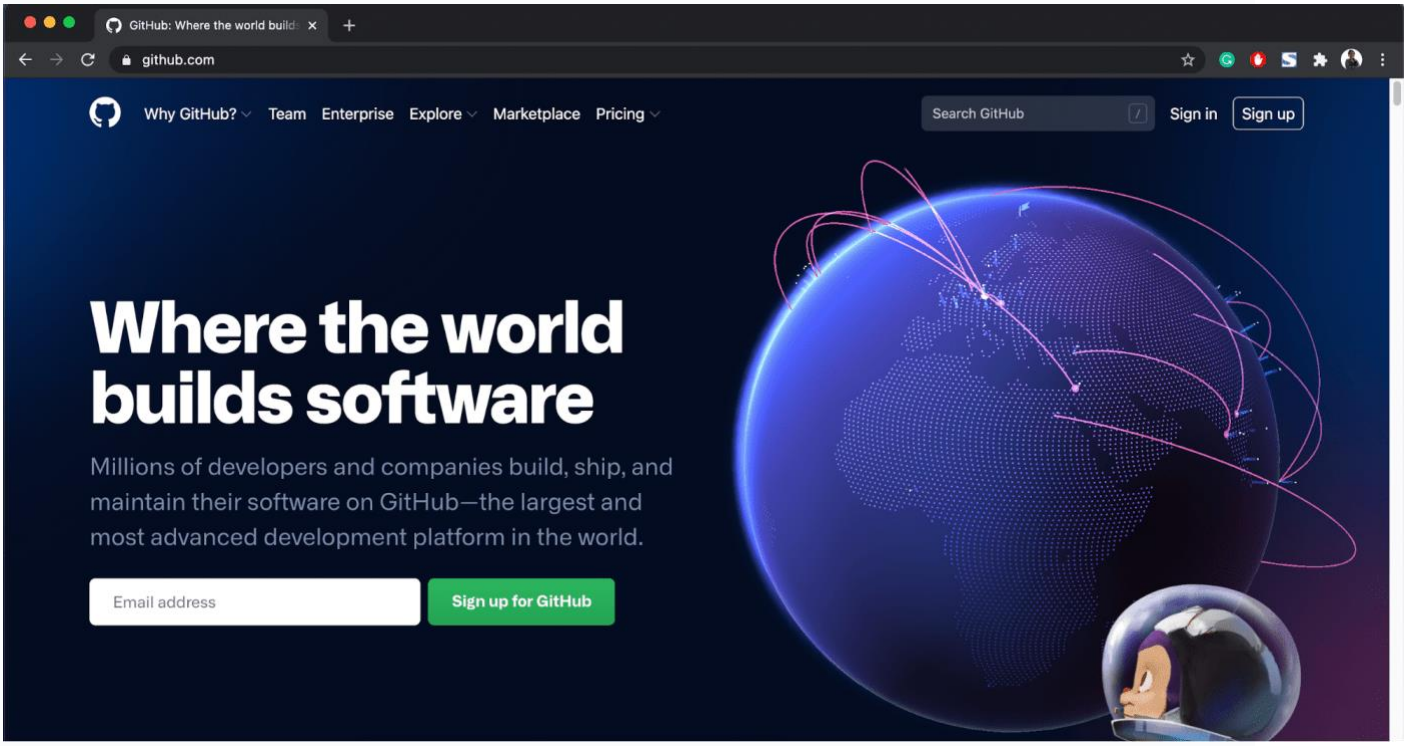
According to the needs of each organization and team, the agile methodology can be further split up into kanban and scrum methodologies – even hybrids such as scrumban, which is what we use at Justinmind.

Kanbanize

There are several agile tools available in the market. Some of them are listed below:

1)Clickup 2)Jira 3)Kanbanzie 4)Github 5)PlanBox 6)Active Collab 7)Proggio

. ii) GitHub.



Via Github

GitHub is the largest hosted Git server….but what does that mean?

Basically, your developers can store all of your code for a vast number of projects there. GitHub is so great because it records edits across an entire team in real time.

This Agile project management software also integrates with many other tools so that multiple team members (from your developers to the product owner) can work in the same code simultaneously, making it one of the better Agile tools for development teams.

GitHub gives you a private space for each team member and a public space where members of the community can come and help you improve your code.

GitHub also includes many agile project management tools to help project managers manage what the team members are working on.

Agile Software Features:

- Issue tracking
- Mentions for teamwork

- Labels
- Link issues and pull requests

Pricing: Free to $21 per user/per month

Q5) a) Explain release planning and iteration (Sprint) planning in brief.[6 marks]

-->There are 3 levels of planning in Agile. They are Release Planning, Iteration Planning and Daily Planning. These planning meetings help the **Scrum Master, Product Owner** and the rest of the team in understanding how the product will be delivered, the complexity involved and their day to day responsibility in the delivery of the product, among other things.

Now we will go into detail of **Release Planning**, **Iteration Planning** and **Daily Planning**.

Release Planning in Agile Development

- Release Planning is usually performed during the Sprint zero, where there is no product increment delivered.
- The whole Sprint is dedicated for planning the next release.
- It is a way of looking ahead on defining what the release goal is, what features that need to delivered during the release, defining the release backlog, breaking features and **epics into user stories**, writing acceptance criteria for all stories, and **estimating the user stories**.
- Release planning also helps team members in defining the test strategy and test approach planning for all iterations.
- Release plans may change based on new stories added or deleted.
- The status of the release is tracked prospectively every sprint to understand what it takes to meet the release

At the start of the release planning, the Product Owner sets the release goal and release time frame. The Product Owner also collaborates with the team (see **Collaborative User Story Creation**), based on the user stories the team performs high architecture evaluation and high level **effort estimation in agile**.

Testers are involved in release planning and perform the following activities:

- Write User stories and acceptance criteria
- Seek clarification on those user stories where there is insufficient information.
- Determine the high level test strategy for the whole release
- Point out any testing risks they might occur
- Do a high level test planning
- Define the number of test levels to be performed.

Iteration Planning in Agile Development

Iteration 1 planning is done **after release planning**.

- The team pulls the stories into the sprint backlog from the product backlog and groups them into independent tasks of fewer 8 hours each.
- They also performs risk assessment of stories and decides on a light weight plan on how to address the risks as the sprint moves forward.
- The team asks questions to the Product Owner on clarifications that might make them to understand the stories in more realistic way.
- The number of stories that the team pulls in the sprint may depend on team's capacity or velocity (if known).
- The capacity planning is done in hours and the velocity planning is done in story points.
- The team makes a commitment to the sprint backlog and changes the sprint backlog as it emerges.

The testers are also involved in the iteration and can contribute to the same in the following ways:

- Breaking user stories into testing tasks
- Determining test coverage of every story
- Creating acceptance tests for user stories
- Estimate the testing tasks like creating test strategy, test plans and test cases specific to user stories
- Understand and plan for automating user stories and support various levels of testing

Release plans are not static documents and they are liable to change as the external and internal factors change as the iteration executions take place.

- The release plans may also change due to various internal factors like delivery team capabilities, velocity of the team and technical competencies.
- The examples of external factors are change in target segments, threat from substitutes, change triggered by superior competitor product.
- The release dates may be adjusted or scope may be cut down to adjust for the changes.
- As per Scrum, no changes are allowed during the sprint by the Product Owner.
- The Product Owner cannot add new stories to the sprint backlog during the sprint and distract the team from the executing the sprint goal.

The Scrum master must help the team to see that the product owner does not disturb the team, by inserting new work into the team.

The team, especially testers must understand the big picture in order to release a fine quality product. Any ambiguity regarding the test planning must be carefully considered after due consultations with the rest of the team.

Release and iteration planning broadly helps the team to prepare test planning checklist related to the following items:

- The team members responsible for testing
- The scope of testing and testing goals to be carried out
- The test environment that needs to be setup
- The test data to be collected that helps the testing team
- The sequence of test activities, dependencies and interfaces related to testing
- Any project related quality tests that surface during testing (see **Assess Quality Risks in Agile**).

Explain the process to plan and execute iteration in agile with suitable example.[6 marks]

--> Define Iteration Goals: Start by identifying the specific goals and objectives for the upcoming iteration. These goals should align with the overall project objectives and address the highest priority items from the product backlog. For example, let's say you're developing a mobile app for

a ride-sharing service, and one of the iteration goals is to implement a new payment integration.

Backlog Refinement: Collaboratively review and refine the product backlog items (PBIs) that are relevant to the iteration. This involves breaking down large items into smaller, manageable tasks, estimating effort, and adding any necessary details. For our example, the payment integration PBI may involve tasks such as API research, UI design, backend development, and testing.

Sprint Planning: Select a subset of PBIs from the refined backlog items to form the sprint backlog for the iteration. The team collectively decides which PBIs they can commit to completing during the iteration based on their capacity and velocity. Continuing with the example, the team might select the payment integration PBI along with a few other related tasks.

Task Breakdown: With the selected PBIs, the team further breaks them down into specific tasks. Each task should be small enough to be completed within a few hours to a couple of days. For instance, the UI design task could be broken down into wireframing, visual design, and prototyping.

Estimation: Estimate the effort required for each task using a technique like story points or hours. The team can collectively assign relative estimates based on their past experience and knowledge. Estimation helps in planning and prioritizing work effectively.

Sprint Execution: The team works on the tasks identified in the sprint backlog. They collaborate, track progress, and address any obstacles that arise. Daily stand-up meetings are conducted to provide updates, discuss challenges, and plan for the day's work.

Continuous Integration and Testing: As tasks are completed, the developed features are integrated into the main codebase, and automated or manual testing is conducted to ensure their quality. In our example, the payment integration feature would be tested thoroughly to verify its functionality, security, and compatibility.

Review and Feedback: At the end of the iteration, the team reviews the work completed during the sprint with stakeholders, such as the product owner or clients. Feedback is gathered, and any necessary adjustments or improvements are identified. This feedback loop helps in refining the product and aligning it with the stakeholders' expectations.

Retrospective: The team conducts a retrospective meeting to reflect on the iteration's successes and challenges. They discuss what worked well, areas for improvement, and actionable steps to enhance their processes and collaboration. These insights feed into future iterations.

Repeat: The above steps are repeated for each iteration, typically with a fixed time frame (e.g., two weeks) until the project's completion. The team continuously adapts, learns, and iteratively builds the product based on feedback and changing requirements.

By following this iterative process, Agile teams can efficiently plan and execute their work, delivering value incrementally and adapting to evolving needs.

b) Write short note (any one). [4]

i)Role of project manager:

Role of a Project Manager:

1.Leader

A project manager must lead his team and should provide them direction to make them understand what is expected from all of them.

2. Medium:

The Project manager is a medium between his clients and his team. He must coordinate and transfer all the appropriate information from the clients to his team and report to the senior management.

3. Mentor: He should be there to guide his team at each step and make sure that the team has an attachment. He provides a recommendation to his team and points them in the right direction.

Responsibilities of a Project Manager:

1.Managing risks and issues.

2.Create the project team and assigns tasks to several team members.

3.Activity planning and sequencing.

4.Monitoring and reporting progress.

5.Modifies the project plan to deal with the situation.

.ii) Software configuration management:

When we develop software, the product (software) undergoes many changes in their maintenance phase; we need to handle these changes effectively.

Several individuals (programs) works together to achieve these common goals. This individual produces several work product (SC Items) e.g., Intermediate version of modules or test data used during debugging, parts of the final product.

The elements that comprise all information produced as a part of the software process are collectively called a software configuration.

These are handled and controlled by SCM. This is where we require software configuration management.

A configuration of the product refers not only to the product's constituent but also to a particular version of the component.

Therefore, SCM is the discipline which

o Identify change

o Monitor and control change

o Ensure the proper implementation of change made to the item.

o Auditing and reporting on the change made.

Configuration Management (CM) is a technic of identifying, organizing, and controlling modification to software being built by a programming team.

Why do we need Configuration Management?

Multiple people are working on software which is consistently updating. It may be a method where multiple version, branches, authors are involved in

a software project, and the team is geographically distributed and works concurrently. It changes in user requirements, and policy, budget, schedules need to be accommodated.