

B. Tech CSE AY 2023-24
Project Management

Assignment No 4

Due date-13/09/2023

Name - Khushi Nitinkumar Patel
PRN - 2020BTECS00037
Batch - B3

1. Define Sprints How scrum will be planned and executed using Sprints?

Sprints are a fundamental concept in Scrum, which is an agile framework for project management and product development. A Sprint is a time-boxed iteration during which a specific set of work is planned, executed, and potentially delivered. Sprints are typically short, usually lasting between two to four weeks, with two weeks being the most common duration. Here's how Scrum is planned and executed using Sprints:

1.Product Backlog: The Scrum process begins with a Product Backlog, which is a prioritized list of all the work items (user stories, features, bug fixes, etc.) that need to be done for a product. The Product Owner is responsible for maintaining and prioritizing this backlog.

2.Sprint Planning: At the start of each Sprint, there is a Sprint Planning meeting. During this meeting, the Scrum Team (which includes the Development Team and Scrum Master) selects a subset of items from the Product Backlog that they believe can be completed within the Sprint. These selected items are called the Sprint Backlog.

3.Sprint Goal: The Scrum Team defines a Sprint Goal, which is a high-level objective or outcome they aim to achieve by the end of the Sprint. The Sprint Goal provides focus and direction for the team during the Sprint.

4. Daily Scrum: Throughout the Sprint, the Development Team holds daily stand-up meetings known as the Daily Scrum or Daily Standup. Each team member briefly discusses what they did yesterday, what they plan to do today, and any obstacles they are facing. The Scrum Master facilitates these meetings.

5.Development: During the Sprint, the Development Team works on completing the items in the Sprint Backlog. They collaborate closely and use Scrum practices to ensure progress and quality.

6.Incremental Development: The idea is to produce a potentially shippable product increment by the end of the Sprint. This means that at the end of each Sprint, there should be a working, usable product feature or set of features that could be released if desired.

7.Review and Demo: At the end of the Sprint, there is a Sprint Review meeting where the Scrum Team demonstrates the completed work to stakeholders and gathers feedback. This helps ensure that the product is meeting user expectations.

8.Retrospective: Following the Sprint Review, there is a Sprint Retrospective meeting where the team reflects on their processes and identifies areas for improvement. The goal is to continuously improve the team's effectiveness and efficiency.

9.Repeat: The process repeats for the next Sprint. The Sprint Review and Sprint Retrospective feed into the next Sprint Planning meeting, and the cycle continues until the project is complete or the product is retired.

Scrum relies on these time-boxed iterations (Sprints) to provide transparency, inspection, and adaptation throughout the project. This iterative and incremental approach allows teams to respond to changing requirements and deliver value to customers more frequently and predictably.

2. What is the purpose of

a. Daily Stand-up (Daily Scrum):

Purpose: The Daily Stand-up is a short, daily meeting (typically 15 minutes or less) held by the Development Team. Its primary purposes are:

- Daily Synchronization: To keep the team synchronized by providing updates on what each team member accomplished yesterday, what they plan to do today, and any impediments or blockers they are facing.
- Identify and Resolve Issues: To identify and address obstacles or issues early, promoting quick problem-solving and collaboration.
- Visibility: To increase transparency within the team, making it easier to track progress and ensure everyone is aligned with the Sprint Goal.

b. Iteration Review (Sprint Review):

Purpose: The Sprint Review is held at the end of each Sprint and involves stakeholders, including the Product Owner, customers, and other interested parties. Its primary purposes are:

- Demonstration: To showcase the product increment developed during the Sprint and provide stakeholders with a tangible view of the progress made.
- Feedback Gathering: To gather feedback from stakeholders regarding the product increment, which helps in validating and adapting the product's direction.
- Inform Decision-Making: To provide information that can guide decisions about what to do next, such as adjusting priorities in the Product Backlog or making changes to the product roadmap.

- Transparency: To maintain transparency and ensure that the product aligns with stakeholder expectations.

c. Retrospective:

Purpose: The Retrospective is held at the end of each Sprint and is attended by the Scrum Team (Development Team, Scrum Master, and Product Owner). Its primary purposes are:

- Continuous Improvement: To reflect on the previous Sprint, team processes, and interactions with the aim of identifying what went well and what could be improved.
- Actionable Insights: To generate actionable insights and action items for process improvement in subsequent Sprints.
- Team Collaboration: To promote open and constructive communication among team members and foster a culture of continuous learning and adaptation.
- Adaptation: To facilitate the adaptation of the team's practices and processes to enhance efficiency, quality, and overall performance.

These ceremonies are integral to the Scrum framework, as they enable teams to collaborate, learn, and adjust their approach throughout the project. They help teams to maintain focus on delivering value, respond to changes effectively, and foster a culture of continuous improvement.

3. What are the responsibilities of Scrum master?

1. Servant Leadership: The Scrum Master is a servant leader who serves the Scrum Team by helping them understand and apply Scrum principles and values. They also serve the organization by coaching and supporting the adoption of Scrum.

2. Facilitator: The Scrum Master facilitates Scrum events (such as Sprint Planning, Daily Stand-up, Sprint Review, and Sprint Retrospective) to ensure they are productive, time-boxed, and achieve their intended purpose.

3. Removing Obstacles: Scrum Masters identify and remove impediments or obstacles that are hindering the team's progress. They work to create a productive and collaborative work environment.

4. Coaching and Training: Scrum Masters coach the Scrum Team on Scrum practices, help them improve their performance, and encourage self-organization. They may also provide training to the team and stakeholders as needed.

5. Scrum Framework Expertise: Scrum Masters have a deep understanding of the Scrum framework, its principles, and its roles. They ensure that the team understands and follows Scrum practices correctly.

6. Protecting the Team: Scrum Masters shield the team from external disruptions and interference so that the team can focus on their work during the Sprint.

7. Continuous Improvement: Scrum Masters facilitate continuous improvement by conducting Sprint Retrospectives and helping the team implement improvements in their processes.

8. Stakeholder Collaboration: They facilitate communication and collaboration between the Scrum Team and stakeholders, including the Product Owner and external parties.

9. Conflict Resolution: Scrum Masters help resolve conflicts within the team and encourage a positive and productive working environment.

10. Metrics and Transparency: They encourage the use of metrics and data to assess the team's performance and promote transparency by ensuring that relevant information is visible to the team and stakeholders.

11. Promoting Scrum Values: Scrum Masters advocate for the Scrum values of commitment, courage, focus, openness, and respect, helping the team and organization embody these values.

12. Scaling Scrum: In organizations using Scrum at scale (e.g., Large-Scale Scrum, SAFe), Scrum Masters may also have responsibilities related to coordinating with other Scrum Masters and ensuring alignment between multiple Scrum Teams.

4. How Scrum of Scrum is different from Scrum meeting? What are benefits of it?

Aspect	Scrum of Scrums (SoS)	Scrum Meeting (Daily Stand-up)
Purpose	Coordinate multiple Scrum Teams on a large project or product.	Synchronize and plan the daily work of a single Scrum Team.
Frequency	Less frequent, typically once a day, every other day, or as needed.	Daily, held every working day.
Participants	Representatives from each Scrum Team attend.	Members of a single Scrum Team participate.
Focus	Cross-team coordination, dependencies, and alignment with project goals.	Team-level progress, daily planning, and issue identification within the team.
Content	Discussion includes progress toward project objectives, cross-team impediments, and dependencies.	Team members answer three questions: What did I do yesterday? What will I do today? Are there any impediments?
Benefits	Improved coordination, early issue detection, transparency, alignment, risk mitigation.	Daily synchronization, issue identification, and improved team collaboration.

Benefits of Scrum of Scrums (SOS):

- 1. Improved Coordination:** SoS meetings help ensure that multiple Scrum Teams are working cohesively toward common goals, reducing duplication of effort and improving overall project coordination.
- 2. Early Issue Detection:** They provide a forum for identifying and addressing cross-team impediments or dependencies early in the project, reducing the risk of delays or disruptions.
- 3. Transparency:** SoS meetings increase transparency at a higher project level, allowing stakeholders to understand the status and progress of the entire project or product.
- 4. Alignment:** They help maintain alignment with the product vision and ensure that all teams are working toward the same objectives.
- 5. Risk Mitigation:** By regularly discussing and resolving inter-team issues, SoS meetings can help mitigate risks associated with complex projects.

In summary, Scrum of Scrums is a coordination and integration technique used in scaled agile environments to facilitate communication and collaboration between multiple Scrum Teams, while a Scrum meeting (Daily Stand-up) is a daily event within a single Scrum Team to synchronize and plan their work. Both serve important purposes in agile project management.

5. Explain agile workflows w.r.t. Project management.

Agile workflows, in the context of project management, refer to the processes and practices used to manage and deliver projects using agile methodologies such as Scrum, Kanban, or Extreme Programming (XP). Agile workflows are designed to accommodate changing requirements, promote collaboration, and deliver value to customers in a flexible and iterative manner. Here's an overview of how agile workflows work:

1. Project Initiation:

- Vision and Strategy: The project starts with a clear vision and strategic goals, providing direction to the project team.
- Product Backlog: In Scrum, a Product Owner maintains a prioritized list of all project requirements, features, and user stories. In Kanban, this is often represented as a backlog or queue of work items.

2. Release and Sprint Planning:

- Release Planning (Scrum): The Product Owner and team determine what features and functionality will be delivered in upcoming releases, usually a few months ahead.
- Sprint Planning (Scrum): Before each Sprint, the team selects a subset of items from the Product Backlog to work on. In Kanban, work items are pulled from the backlog as capacity allows.

3. Execution:

- Sprints (Scrum): The project is divided into time-boxed iterations called Sprints, typically lasting two to four weeks. During each Sprint, the team develops, tests, and potentially delivers a potentially shippable product increment.
- Kanban (Kanban): Work items flow continuously through the Kanban board, from the backlog through various stages (e.g., to-do, in progress, done) based on capacity and prioritization.

4. Daily Stand-up (Scrum):

- Teams hold daily stand-up meetings to synchronize their work, discuss progress, identify impediments, and plan their tasks for the day. It fosters collaboration and quick issue resolution.

5. Continuous Delivery (Kanban):

- In Kanban, there are no fixed iterations like in Scrum. Instead, work items are continuously delivered as soon as they are completed and meet the defined criteria.

6. Review and Demo (Scrum):

- At the end of each Sprint, a Sprint Review is held to demonstrate the completed work to stakeholders and gather feedback.

7. Retrospective (Scrum):

- Following the Sprint Review, a Sprint Retrospective meeting is conducted to reflect on the team's processes and identify areas for improvement.

8. Backlog Refinement:

- Regularly, the Product Owner and team engage in backlog refinement sessions to add, remove, or reprioritize items based on feedback and evolving requirements.

9. Monitoring and Metrics:

- Agile workflows often involve tracking progress and team performance using metrics such as velocity, cycle time, and lead time.

10. Adaptation and Iteration:

- Based on feedback and insights from stakeholders, team retrospectives, and continuous monitoring, the project adapts and iterates on its plans, priorities, and processes to optimize outcomes.

11. Customer Collaboration:

- Agile workflows emphasize customer collaboration throughout the project to ensure that the product aligns with customer needs and expectations.

12. Scaling (For Large Projects):

- For larger projects involving multiple teams, frameworks like SAFe (Scaled Agile Framework) or LeSS (Large Scale Scrum) may be used to coordinate and align the efforts of multiple agile teams.

Agile workflows prioritize flexibility, customer-centricity, and responsiveness to change, making them suitable for projects where requirements are not fully known upfront or where adaptability is crucial to success. The iterative and incremental approach allows for early and frequent delivery of value, which can lead to higher customer satisfaction and better project outcomes.

6. What are stories, epics, and initiatives? Explain with suitable examples.

In Agile project management, "stories," "epics," and "initiatives" are terms used to describe different levels of work items or features within a project. They help in organizing and managing the scope of a project or product. Here's an explanation of each term with suitable examples:

1. User Stories:

- Definition: User stories are small, granular units of work that represent a single piece of functionality or a specific user need. They are typically written from the perspective of an end user or customer and describe what the user wants to achieve.
- Format: User stories follow a common format, often known as the "As a [user], I want [an action] so that [a benefit or reason]."
- Example:
- User Story: "As a website visitor, I want to be able to create an account so that I can access exclusive content."

2. Epics:

- Definition: Epics are larger units of work that encompass multiple user stories. They represent high-level, often cross-functional features that are too large to be completed in a single iteration or Sprint.
- Scope: Epics are typically too broad to implement directly and need to be broken down into smaller, more manageable user stories during backlog refinement.
- Example:
- Epic: "User Profile Management"
- User Stories within the Epic:
- "As a registered user, I want to update my profile information."
- "As an admin, I want to be able to view and manage user profiles."

3. Initiatives:

- Definition: Initiatives are even higher-level groupings of work that align with strategic goals or business objectives. They can encompass multiple epics and represent a substantial effort that spans multiple Sprints or releases.
- Scope: Initiatives help in organizing and prioritizing work at a strategic level. They may involve a combination of features, improvements, and other activities.
- Example:
- Initiative: "Enhance E-commerce Checkout Experience"
- Epics within the Initiative:

- "Streamline the checkout process for guest users."
- "Improve payment gateway integration for a smoother payment experience."
- "Enhance order confirmation and tracking functionality."
- Example Summary:
Suppose you're working on an e-commerce website. Here's how these terms could be applied:
- User Story: "As a website visitor, I want to be able to create an account so that I can access exclusive content."
- Epic: "User Profile Management"
- User Stories within the Epic:
 - "As a registered user, I want to update my profile information."
 - "As an admin, I want to be able to view and manage user profiles."
- Initiative: "Enhance E-commerce Checkout Experience"
- Epics within the Initiative:
 - "Streamline the checkout process for guest users."
 - "Improve payment gateway integration for a smoother payment experience."
 - "Enhance order confirmation and tracking functionality."

7. What are the responsibilities of the Program manager and Project manager?

Responsibilities of a Program Manager:

1. Program Definition: Define the strategic objectives and scope of the program. Ensure alignment with organizational goals and objectives.

2. Portfolio Management: Oversee a portfolio of related projects and initiatives within the program, ensuring that they collectively deliver value and align with the program's goals.

3. Stakeholder Management: Engage with key stakeholders, including executives, sponsors, and other program managers, to communicate progress, address concerns, and manage expectations.

4. Resource Allocation: Allocate resources (human, financial, and other assets) across various projects and initiatives within the program to optimize their delivery.

5. Risk Management: Identify, assess, and manage risks and issues that affect the program's success. Implement risk mitigation and contingency plans.

6. Governance: Establish and maintain governance structures and processes to ensure that projects and initiatives within the program are executed in a controlled and consistent manner.

7. Strategic Planning: Develop and refine the program's strategic roadmap, including defining milestones, timelines, and key performance indicators (KPIs).

8. Alignment: Ensure that all projects within the program are aligned with the program's objectives and that they collectively contribute to the program's success.

9. Communication: Foster clear and effective communication within the program and with external stakeholders. Ensure that information flows smoothly between teams and individuals.

10. Quality Assurance: Oversee the quality of project deliverables and ensure that they meet the program's standards and requirements.

Responsibilities of a Project Manager

1. Project Planning: Develop a detailed project plan, including scope, schedule, budget, resources, and risk management strategies.

2. Scope Management: Define and manage the project's scope, ensuring that it stays within the predefined boundaries and objectives.

3. Team Management: Build and lead the project team, assign tasks, and ensure that team members have the necessary resources and support to carry out their work.

4. Risk and Issue Management: Identify, assess, and manage risks and issues specific to the project. Develop and execute mitigation plans.

5. Execution: Oversee the execution of the project, monitor progress, and ensure that it stays on track in terms of schedule, budget, and quality.

6. Stakeholder Communication: Maintain effective communication with stakeholders, including regular status updates, issue resolution, and addressing concerns.

7. Quality Control: Ensure that project deliverables meet quality standards and comply with project requirements.

8. Budget Management: Manage the project budget, track expenditures, and ensure that financial resources are used efficiently.

9. Change Management: Manage changes to project scope, schedule, or objectives, and assess their impact on the project.

10. Closure: Formalize project closure, including documentation, handover of deliverables, and lessons learned.

In summary, a Program Manager focuses on strategic alignment and oversight of multiple projects and initiatives, whereas a Project Manager is responsible for the detailed planning and execution of a single project. Both roles are essential in ensuring that work is delivered efficiently and effectively in organizations.

8. What is Gantt chart? Explain its usefulness in managing a project.

A Gantt chart is a visual project management tool used to plan, schedule, and track tasks and activities over time. It provides a graphical representation of a project's timeline, showing the start and end dates of various tasks or activities, their dependencies, and the overall project schedule. Gantt charts are named after Henry L. Gantt, who developed this tool in the early 20th century.

Usefulness of Gantt Charts in Managing a Project:

1. Visual Planning: Gantt charts provide a clear, visual representation of the project's schedule. Project managers and team members can quickly grasp the project's timeline, tasks, and their sequence.

2. Task Dependencies: Gantt charts help identify task dependencies, showing which tasks are prerequisites for others. This ensures that tasks are scheduled in the correct order, preventing bottlenecks and delays.

3. Resource Allocation: Project managers can allocate resources (e.g., team members, equipment) more effectively by seeing when and where resources are needed throughout the project.

4. Schedule Optimization: Gantt charts allow project managers to identify critical paths—the sequence of tasks that must be completed on time to prevent project delays. This helps in optimizing the schedule and focusing efforts on critical tasks.

5. Progress Tracking: As the project progresses, Gantt charts can be updated to reflect the actual start and end dates of completed tasks. This enables tracking of project progress against the planned schedule.

6. Communication: Gantt charts serve as a communication tool, enabling project managers to convey the project plan to stakeholders, team members, and clients in a simple and understandable format.

7. Risk Management: By visualizing the project schedule, project managers can identify potential schedule risks and develop contingency plans to mitigate them.

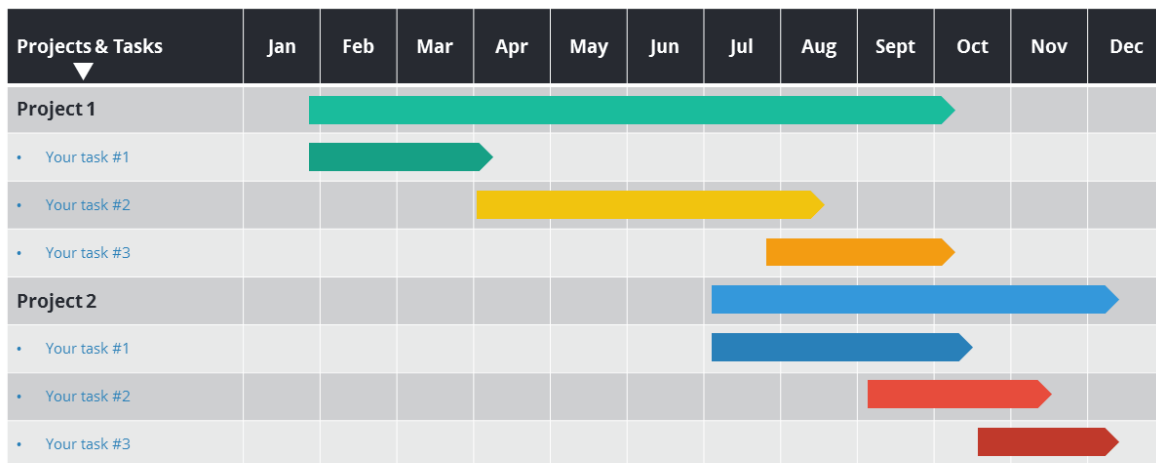
8. Resource Management: Gantt charts assist in resource allocation and workload balancing. Managers can see if resources are over-allocated or under-utilized, helping with resource management decisions.

9. Decision-Making: Project managers can use Gantt charts to make informed decisions about project adjustments, such as task prioritization, resource allocation changes, or schedule modifications.

10. Documentation: Gantt charts serve as a historical record of the project schedule. They can be valuable for post-project analysis, audits, and lessons learned.

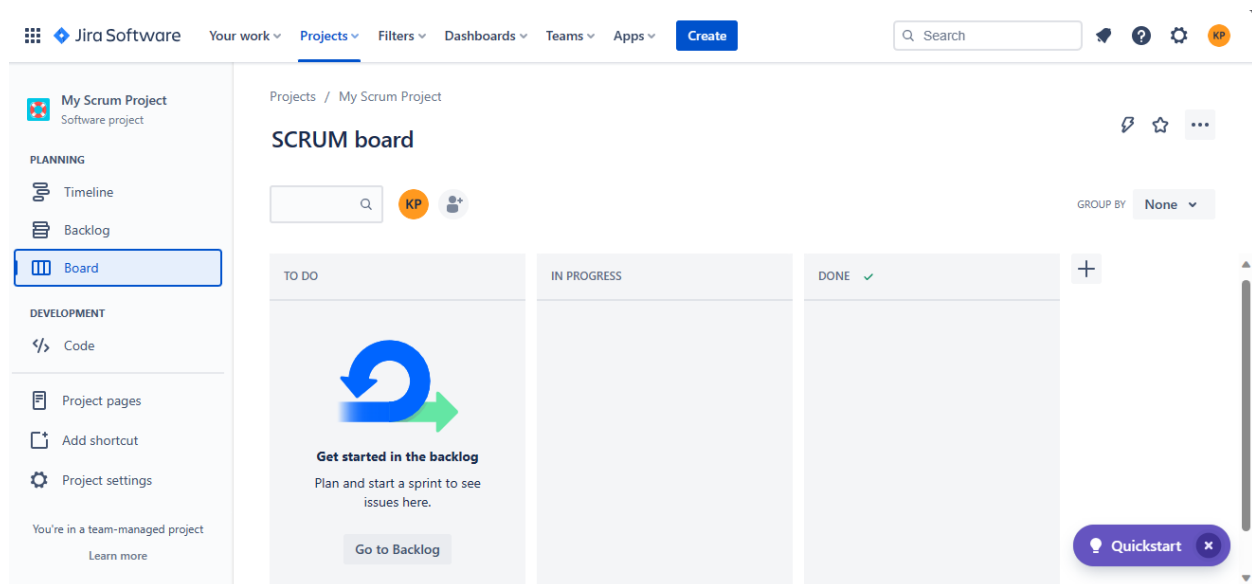
9. Differentiate use of Gantt chart in Agile and Waterfall planning.

Aspect	Gantt Chart in Agile	Gantt Chart in Waterfall
Project Phases	Agile projects have iterative phases (Sprints) with shorter durations.	Waterfall projects have distinct, sequential phases (Requirements, Design, Implementation, Testing, Deployment).
Task Dependency	Task dependencies are often more flexible, as Agile embraces change.	Task dependencies are typically rigid, with a linear progression from one phase to the next.
Scope Changes	Agile accommodates scope changes during the project (scope is not fixed).	Waterfall has a fixed scope defined upfront, making scope changes challenging.
Iterative Planning	Gantt charts in Agile may be shorter-term, reflecting the current Sprint or release.	Gantt charts in Waterfall often cover the entire project duration from start to finish.
Detailed vs. High-Level	Agile Gantt charts may focus on high-level milestones and sprint-level tasks.	Waterfall Gantt charts detail tasks, dependencies, and timelines for each project phase.
Emphasis	Agile Gantt charts emphasize adaptability and responsiveness to change.	Waterfall Gantt charts emphasize a structured, sequential approach to project management.
Tracking Progress	Agile Gantt charts require frequent updates as Sprints progress.	Waterfall Gantt charts are updated less frequently since phases are sequential.
Stakeholder Visibility	Agile Gantt charts often provide stakeholders with a more dynamic view of progress.	Waterfall Gantt charts offer stakeholders a comprehensive but less adaptive view.
Risk Management	Agile Gantt charts help in identifying and mitigating risks associated with iterative development.	Waterfall Gantt charts help in managing risks within each phase but may struggle with late-stage discoveries.



10. Create an account on Jira for your project team (one student will create an account using college email id) and add your team members into your project, name your project.

Account created by Utkarsha Sutar



The screenshot displays the Jira Software interface for a project named "My Scrum Project". The top navigation bar includes "Jira Software", "Your work", "Projects", "Filters", "Dashboards", "Teams", "Apps", and a "Create" button. A search bar is located on the right. The left sidebar shows the project structure with "My Scrum Project" (Software project) and a "Board" view selected under the "PLANNING" section. The main area shows the "SCRUM board" with three columns: "TO DO", "IN PROGRESS", and "DONE". The "TO DO" column contains a "Get started in the backlog" card with a "Go to Backlog" button. The "IN PROGRESS" and "DONE" columns are currently empty. A "Quickstart" button is visible in the bottom right corner.

Jira Software

Your workProjectsFiltersDashboardsTeamsAppsCreate

Q Search

?

⚙

KP


People and teams

Create teamAdd people

Q Search for people and teams


People

More search options with Atlas




Your teammate


Add people




Khushi Nitinkumar ...
Designer



pragati.soley




21520004 UTKARS...



shireen

Your teams



Jira Software

Your workProjectsFiltersDashboardsTeamsAppsCreate

Q Search

?

⚙

KP

My Scrum Project
Software project

PLANNING

Timeline

Backlog

Board

DEVELOPMENT

Code

Project pages

Add shortcut

Project settings

You're in a team-managed project
Learn more

Projects / My Scrum Project

Backlog

Q KP Epic

Insights

▼ SCRUM Sprint 1 Add dates (0 issues) 0 0 0 Start sprint

Plan your sprint

Drag issues from the Backlog section, or create new issues, to plan the work for this sprint. Select **Start sprint** when you're ready.

What needs to be done?

Story

▼ Backlog (0 issues) 0 Quickstart

Your backlog is empty.