INVEST

"INVEST" is an acronym commonly used in agile software development to describe key characteristics of well-formed user stories or product backlog items. Each letter stands for a different attribute:

1. **Independent**:
   * **Description**: User stories should be independent of each other, meaning they can be developed, tested, and delivered in isolation without being dependent on other stories.
   * **Benefits**: Independence allows for better prioritization, flexibility, and parallel development. It reduces bottlenecks and dependencies between team members.
2. **Negotiable**:
   * **Description**: User stories should be negotiable between the development team and stakeholders. They should serve as a starting point for discussion and collaboration, rather than being fixed requirements.
   * **Benefits**: Negotiability encourages collaboration and flexibility, enabling stakeholders to provide feedback and refine requirements based on changing priorities or new insights.
3. **Valuable**:
   * **Description**: User stories should deliver value to the customer or end-user. They should focus on addressing user needs, solving problems, or delivering features that contribute to the overall goals of the project.
   * **Benefits**: Focusing on value ensures that development efforts are aligned with the needs and priorities of the stakeholders. It helps teams prioritize work based on the impact it will have on the user or business.
4. **Estimable**:
   * **Description**: User stories should be estimable, meaning the development team should be able to estimate the effort required to implement them. They should be clear and well-defined enough for the team to provide accurate estimates.
   * **Benefits**: Estimability helps in planning and resource allocation. It allows teams to make informed decisions about prioritization, scheduling, and capacity management.
5. **Small**:
   * **Description**: User stories should be small enough to be completed within a single iteration or sprint. They should focus on delivering specific functionality or addressing specific user needs.
   * **Benefits**: Small stories reduce risk, improve visibility, and increase the likelihood of successful delivery. They allow for incremental progress, feedback, and adaptation, which leads to faster learning and improvement.
6. **Testable**:
   * **Description**: User stories should be testable, meaning they should have clear acceptance criteria that define the conditions under which the story is considered complete.
   * **Benefits**: Testability ensures that there is a shared understanding of what constitutes a successful implementation. It facilitates collaboration between the development team and stakeholders and promotes a culture of quality and accountability.

By following the principles of INVEST, teams can create user stories that are well-defined, manageable, and focused on delivering value to the customer. This approach promotes collaboration, flexibility, and alignment between stakeholders throughout the development process.

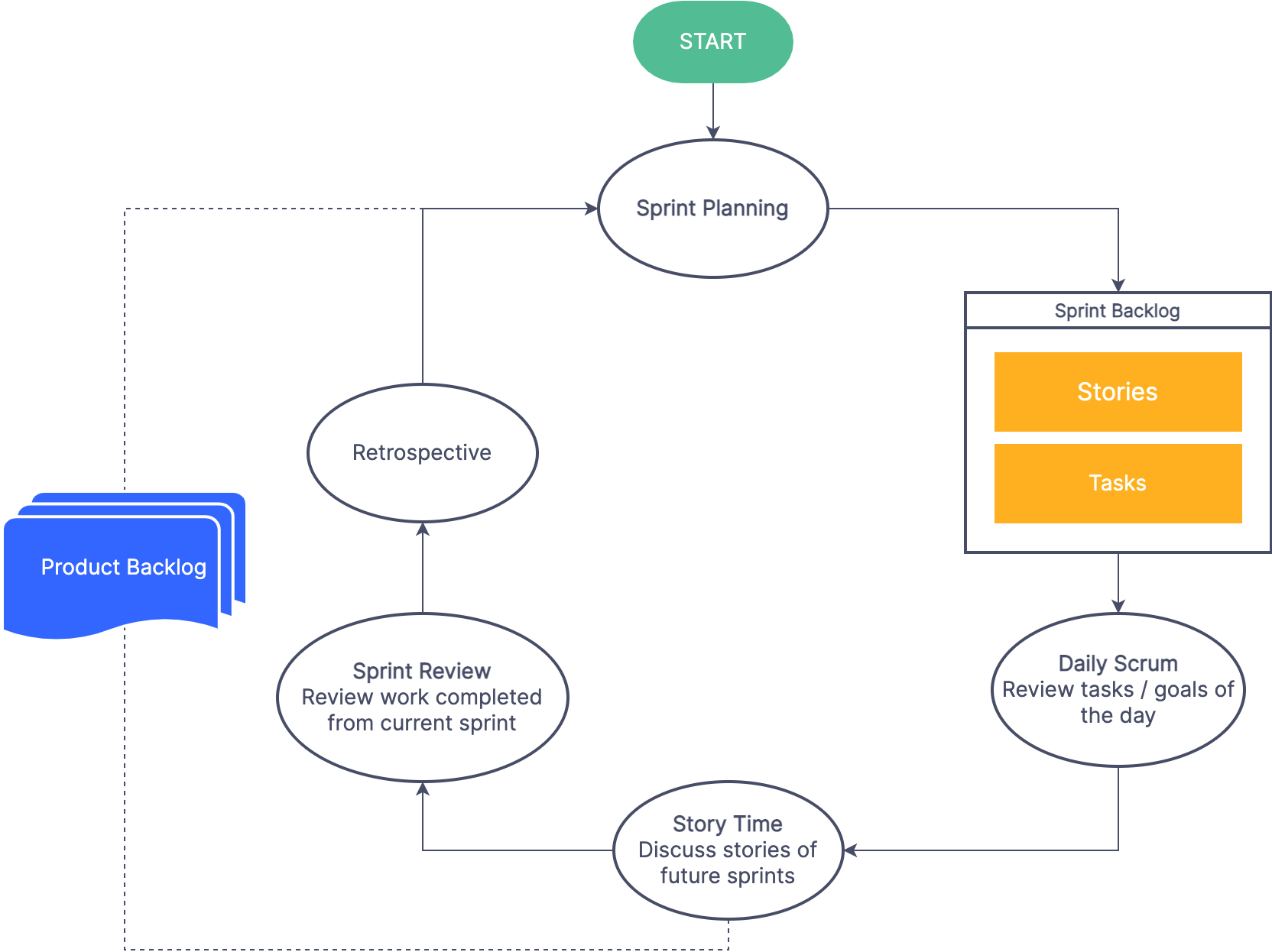
SCRUM

Scrum is a widely used framework in agile software development that provides a structured approach to project management. It emphasizes iterative development, collaboration, and flexibility to deliver high-quality products efficiently. Here's an overview of Scrum:

1. **Roles**:
   * **Product Owner**: Represents the stakeholders and is responsible for defining and prioritizing the product backlog.
   * **Scrum Master**: Facilitates the Scrum process, removes impediments, and ensures that the team adheres to Scrum principles and practices.
   * **Development Team**: Cross-functional team responsible for delivering the product increment.
2. **Artifacts**:
   * **Product Backlog**: A prioritized list of all desired features, enhancements, and fixes for the product. Managed and prioritized by the Product Owner.
   * **Sprint Backlog**: A subset of the Product Backlog items selected for implementation during the sprint. Managed by the Development Team.
   * **Increment**: The sum of all completed Product Backlog items at the end of a sprint, potentially releasable to the stakeholders.
3. **Events**:
   * **Sprint**: A time-boxed period (usually 2-4 weeks) during which the Development Team works to deliver a potentially releasable increment of the product.
   * **Sprint Planning**: A meeting at the beginning of each sprint where the Product Owner and Development Team collaborate to select and plan the work for the sprint.
   * **Daily Standup (Daily Scrum)**: A brief daily meeting where the Development Team synchronizes activities, identifies impediments, and plans work for the next 24 hours.
   * **Sprint Review**: A meeting at the end of the sprint where the Development Team presents the increment to stakeholders and receives feedback.
   * **Sprint Retrospective**: A meeting at the end of the sprint where the Development Team reflects on the sprint and identifies improvements for the next sprint.
4. **Principles**:
   * **Empiricism**: Scrum is based on the principles of transparency, inspection, and adaptation. It relies on empirical feedback to make informed decisions and adapt to changing requirements and circumstances.
   * **Self-Organization**: The Development Team is self-organizing and cross-functional, with the autonomy to determine how best to achieve the sprint goal.
   * **Collaboration**: Scrum emphasizes collaboration between stakeholders, with regular interactions and feedback loops to ensure alignment and shared understanding.
5. **Benefits**:
   * **Flexibility**: Scrum allows teams to adapt to changing requirements and priorities quickly, enabling faster delivery of value to stakeholders.
   * **Transparency**: Scrum promotes transparency through its artifacts and events, providing visibility into progress, impediments, and potential risks.
   * **Quality**: Scrum focuses on delivering high-quality increments through continuous testing, inspection, and adaptation.

Overall, Scrum provides a lightweight, iterative approach to software development that fosters collaboration, transparency, and continuous improvement. It empowers teams to deliver value incrementally and respond effectively to changing requirements and market conditions.

WORK FLOW



SPRINT

When documenting a Sprint in the context of Scrum, you want to provide comprehensive details about the activities, events, and artifacts involved. Here's a breakdown of what to include:

1. **Sprint Overview**:
   * Define what a Sprint is: a time-boxed iteration in Scrum, typically lasting 2-4 weeks.
   * Explain the purpose of Sprints: to deliver a potentially releasable increment of the product and gather feedback for continuous improvement.
2. **Sprint Planning**:
   * Describe the Sprint Planning meeting, its attendees (Product Owner, Development Team, Scrum Master), and its objectives.
   * Outline the agenda for Sprint Planning: review and select items from the Product Backlog, define the Sprint Goal, and create the Sprint Backlog.
   * Provide guidelines for conducting Sprint Planning effectively, such as timeboxing discussions and prioritizing backlog items.
3. **Daily Standup (Daily Scrum)**:
   * Explain the Daily Standup meeting, its purpose, and its attendees (Development Team, Scrum Master).
   * Describe the format of the Daily Standup: each team member answers three questions (What did I do yesterday? What will I do today? Are there any impediments?).
   * Emphasize the importance of keeping the Daily Standup brief and focused on synchronizing activities and identifying impediments.
4. **Sprint Backlog**:
   * Define the Sprint Backlog as a subset of the Product Backlog items selected for implementation during the Sprint.
   * Explain how the Sprint Backlog is created during Sprint Planning and how it evolves throughout the Sprint as work is completed and new insights emerge.
   * Provide examples or templates for documenting the Sprint Backlog, such as a list of tasks with assigned owners and estimated effort.
5. **Sprint Execution**:
   * Detail the activities that take place during the Sprint, including development, testing, and collaboration.
   * Emphasize the importance of self-organization and cross-functional collaboration within the Development Team.
   * Highlight the role of the Scrum Master in removing impediments and facilitating the team's progress.
6. **Sprint Review**:
   * Describe the Sprint Review meeting, its attendees (Product Owner, Development Team, Stakeholders), and its objectives.
   * Outline the agenda for the Sprint Review: demonstration of the increment, feedback collection, and discussion of potential adjustments to the Product Backlog.
   * Provide guidelines for soliciting and incorporating feedback from stakeholders effectively.
7. **Sprint Retrospective**:
   * Explain the Sprint Retrospective meeting, its attendees (Development Team, Scrum Master), and its objectives.
   * Outline the agenda for the Sprint Retrospective: reflection on what went well, identification of improvement opportunities, and creation of action items for the next Sprint.
   * Encourage open and honest communication during the retrospective and emphasize the importance of continuous improvement.
8. **Increment**:
   * Define the Increment as the sum of all completed Product Backlog items at the end of the Sprint.
   * Explain the concept of a potentially releasable Increment and its role in gathering feedback from stakeholders.
9. **Sprint Closure**:
   * Summarize the key outcomes of the Sprint, including the completed work, feedback received, and lessons learned.
   * Provide guidance on documenting any remaining tasks or unfinished work, such as carrying them over to the next Sprint or reprioritizing them in the Product Backlog.
10. **Examples and Templates**:
    * Include examples of Sprint artifacts, such as Sprint Backlog, Sprint Review notes, and Sprint Retrospective action items.
    * Provide templates or guidelines for creating Sprint-related documentation, such as meeting agendas, checklists, or reports.

By documenting the Sprint in detail, you can ensure that all team members have a clear understanding of their roles, responsibilities, and expectations throughout the Sprint. This documentation serves as a valuable reference for planning, executing, and improving future Sprints in the project.

THANK YOU