Section 5 - Scrum Artifacts

Artifact Transparency

**What is Transparency?**

* A key principle in Scrum, emphasized often in the Scrum Guide.
* Helps the team optimize product value, manage risks, and make informed decisions.

**Role of Transparency in Scrum**

* **Artifacts & Decisions**:
  + Product Owner makes decisions based on visible artifacts like:
    - Product Backlog: What’s done and what still needs to be done.
    - Sprint Backlog: Current sprint’s tasks and progress.
    - Increments: Previous iterations and the current work being delivered.
  + Visible work helps identify risks and areas needing improvement.

**How to Make Work Transparent**

* Use tools like Jira, Asana, Trello, or even sticky notes/whiteboards.
* Maintain clear boards for:
  + Product Backlog
  + Sprint Backlog
  + Workflow stages: "To Do," "In Progress," "Done"

**Scrum Master’s Role in Transparency**

* Inspects artifacts to detect gaps in transparency.
* Uses experience and listening skills to identify mismatches between expected and actual results.
* Ensures the team adopts Scrum properly.
* Encourages learning, communication, and gradual improvement in transparency.

**Effective Communication for Transparency**

* Use simple, clear, and shared language.
* Avoid unexplained acronyms or jargon.
* Create a safe environment for team members to ask for clarification.
* Ensure everyone understands:
  + **Definition of Done**: A shared understanding of when work is considered complete.
  + **Sprint Goal**: Team-defined goal crafted together at the start of the sprint.

**Steps to Improve Transparency**

* Frequently inspect Scrum artifacts (backlogs, boards, increments).
* Use Scrum events to discuss and adapt:
  + Daily Scrum: Daily updates and alignment.
  + Sprint Review: Present progress and get feedback.
  + Sprint Retrospective: Identify and implement process improvements.
* Keep the Scrum board updated with current work status.

**Inspection & Adaptation**

* Inspect often to detect issues or variances from the goals.
* Adapt processes based on findings during events like:
  + Sprint Planning
  + Backlog Refinement
* Continuous improvement helps the team stay aligned and achieve goals.

**Important Notes from the Scrum Guide**

* **2017 Guide**:
  + Everyone must share the same language and understand what “done” means.
* **2020 Guide**:
  + Transparency develops over time and requires effort.
  + Work must be visible to both those performing it and those receiving it.

**Key Takeaways**

* Transparency is a process—it improves gradually.
* Make work visible and ensure clarity in communication.
* Scrum events are vital for promoting transparency and continuous adaptation.
* Inspect scrum artifacts.
* Inspection enables adaptations.

Product Backlog

#### **1. Introduction to the Product Backlog**

* The **Product Backlog** is a critical **Scrum artifact**.
* As a **Product Owner**, you are responsible for managing the backlog and ensuring its accuracy and relevance.
* Understanding the backlog is essential for:
  + **Passing the Product Owner certification exam**.
  + Becoming an **effective Product Owner**.

#### **2. Importance of the Product Backlog**

* The Product Backlog is an **ordered list** of everything needed for the product.
* It serves as the **single source of requirements** for any product changes.
* Key elements:
  + Ensures no confusion by centralizing all requirements in one place.
  + Prevents multiple versions of requirement documents, which could lead to redundancy and missed information.

#### **3. Definitions and Responsibilities**

* According to the **Scrum Guide**:
  + The **Product Owner** is **accountable** for:
    - Content.
    - Availability.
    - Ordering of the backlog.
  + While backlog maintenance can be **delegated**, the Product Owner must:
    - Review the backlog.
    - Ensure it is correct, transparent, and effectively communicates priorities.

#### **4. Prioritization**

* The Product Owner has **ultimate decision-making power** over prioritizing backlog items.
* **Higher-priority items**:
  + Contain more detail and are often placed at the top (common in tools like **JIRA**).
* The priority order ensures smooth transition into sprints or upcoming development cycles.

#### **5. Scrum Guide Versions: 2017 vs. 2020**

* The **2020 Scrum Guide** is more concise but retains core information.
* The **2017 Scrum Guide** offers more detailed context and background, which is:
  + Still relevant for certification.
  + Helpful for understanding the evolution of Scrum practices.

#### **6. Backlog Characteristics**

1. **Continuous Existence**:
   * The backlog exists as long as the product exists.
   * After the product's release, the backlog remains active for potential updates or iterations.
2. **Evolving Nature**:
   * The backlog changes iteratively based on:
     + Market feedback.
     + Stakeholder input.
     + Team reviews.
   * It identifies what is needed for the product to remain **competitive and useful**.

#### **7. Backlog Content**

* Includes all **features, functions, requirements, enhancements, and fixes** for future releases.
* Provides clarity about:
  + **What** is needed.
  + **Why** it is needed.
  + **Who** benefits from the change.

#### **8. Features vs. Functions**

* **Feature**:
  + The **benefit** delivered to the user (e.g., taking high-definition photos).
* **Function**:
  + The **mechanism** that enables the feature (e.g., a 12MP camera with ISO adjustments).

#### **9. Practical Application**

* Workflow tools like **JIRA** and **Trello**:
  + Help visualize the backlog.
  + Simplify task prioritization and tracking.
  + Allow for drag-and-drop reordering of items to align with sprint priorities.

#### **10. Key Certification Points**

* The Product Backlog is:
  + **Never complete** and always evolving.
  + The **single source of requirements** for the product.
  + Accountable solely to the **Product Owner**, even if maintenance is delegated.
* Understanding **features** vs. **functions** is critical for defining backlog items clearly.
* Learn to align theory with practice by using tools effectively.

### **The Product Backlog Part 2**

#### **1. Certification Tips**

* Product Backlog items must have attributes of:
  + **Description**: Provides transparency and clarity on what the item is about.
  + **Order (or prioritization)**: Indicates the sequence based on importance.
  + **Size (or estimate)**: Specifies the time and effort required to complete the item.
* Differences between Scrum Guide versions:
  + **2017 Version**: Explicitly listed **description, order, estimate, and value**.
  + **2020 Version**: Less prescriptive, suggesting "such as," leaving discretion to the Scrum Team.
  + **Recommendation**: Use **description, order, and size** as the bare minimum.

#### **2. Importance of Description, Order, and Size**

* **Description**:
  + Provides transparency to all team members.
  + Explains **what needs to be done**, **who it is for**, and **why it is valuable**.
* **Order/Prioritization**:
  + Ensures backlog items are developed in alignment with business priorities.
* **Size/Estimate**:
  + Aids in planning by evaluating time and effort.
  + Estimation accuracy improves through **inspection and adaptation**.
  + Discuss estimation challenges during **sprint retrospectives** to refine future predictions.

#### **3. Adding Completion Criteria (Optional Tip)**

* **Completion Criteria (Acceptance Criteria)**:
  + Helps define what is required for an item to be considered "done."
  + Enhances transparency by outlining expected outcomes.
  + Example: Include specific **testing** steps or ensure **quality criteria** are met.
* **Distinction from Definition of Done**:
  + **Completion criteria** apply to individual backlog items.
  + **Definition of Done** applies to the **increment** as a whole.

#### **4. Backlog as a Living Artifact**

* **Lifespan**:
  + The backlog evolves as long as the product exists.
  + Feedback from the market drives changes and additions to the backlog.
* **Empiricism**:
  + Decisions are based on observed market feedback and changing requirements.
  + Adaptations reflect new **business needs**, **market conditions**, **technology advancements**, or even **legislation changes**.

#### **5. Functional vs. Non-Functional Requirements**

* **Functional Requirements**:
  + Directly related to what the product does (e.g., a specific feature).
  + Always appear as backlog items.
* **Non-Functional Requirements**:
  + Related to **qualities** of the product, such as:
    - **Security**
    - **Scalability**
    - **Usability**
  + May be included in:
    - **Backlog items** (e.g., "ensure UI usability for colorblind users").
    - **Definition of Done** (quality criteria for increments).

#### **6. Multiple Scrum Teams, One Backlog**

* In large organizations with multiple Scrum teams working on the same product:
  + **One Product Backlog** exists for the entire product.
  + Teams coordinate through the **Nexus Framework** (or similar methodologies).

#### **7. Key Takeaways for Certification**

* **Attributes**: Description, Order, and Size are critical for backlog items.
* **Completion Criteria**: While optional, they are useful for clarity but not mandatory in Scrum.
* **Living Artifact**: The backlog evolves throughout the product lifecycle.
* **Empiricism**: Adjustments are based on observed feedback.
* **Team Coordination**: One backlog is maintained regardless of the number of Scrum teams.

#### **8. Next Steps**

* Explore **Product Backlog Refinement** in the next lecture.
* Learn how to handle estimation, prioritize tasks effectively, and refine items for upcoming sprints.

Product Refinement

### **Definition and Purpose**

* Product backlog refinement is the ongoing process of breaking down and clarifying product backlog items (PBIs) into smaller, more precise items.
* The goal is to ensure items are small enough to be completed within one sprint, making them ready for selection during sprint planning.

### **Participants and Responsibilities**

* **Product Owner (PO):**
  + Responsible for ordering the backlog items based on priority.
  + Ensures items are detailed and transparent for the team.
* **Developers:**
  + Estimate the time and effort required for each item.
  + Provide input on breaking down and clarifying PBIs.
* **Scrum Master:**
  + Facilitates the process and ensures it aligns with Scrum principles.

### **Key Elements of Backlog Refinement**

1. **Detail:**
   * PBIs should include specifics about **who** will perform the task, **what** needs to be done, and **why** it’s necessary.
   * The **user story framework** can help clarify details (e.g., "As a [user], I need [functionality] so that [goal]").
2. **Ordering:**
   * The Product Owner determines the order of items, often prioritizing the highest-value or most urgent items at the top.
   * High-priority items require more detail and precise estimates.
   * Tools like Jira allow numerical prioritization for easy sorting.
3. **Sizing:**
   * PBIs must be sized appropriately to ensure they are achievable within a single sprint.
   * Developers are responsible for estimating effort and time, as they are closest to the work.

### **Process and Timing**

* Refinement is an **ongoing activity** throughout the sprint, not a formal Scrum event.
* Teams may schedule regular meetings for backlog refinement, but these meetings are not mandatory Scrum events.
* PBIs can be updated at any time at the Product Owner’s discretion or through delegation. However, the Product Owner retains accountability for the backlog.

### **Tips and Best Practices**

* Use **acceptance criteria** to enhance transparency.
  + These criteria define what must be achieved for a backlog item to be considered "done."
  + While not part of the official Scrum Guide, they help clarify expectations.
* Do not confuse acceptance criteria with the **Definition of Done** (which applies to the increment as a whole).

### **Refinement and Sprint Planning**

* A PBI is considered "ready" for sprint planning when it:
  1. Is sufficiently detailed.
  2. Has been ordered and prioritized.
  3. Includes accurate time and effort estimates from developers.
  4. Is broken down into items small enough to be completed within one sprint.

### **Key Takeaways**

* Backlog refinement ensures the team is prepared for sprint planning by maintaining a clear, prioritized, and actionable backlog.
* The Product Owner owns the backlog but collaborates with the developers and Scrum Master to refine it.
* Developers estimate time and effort to ensure items are feasible within one sprint.
* Refinement meetings, though common, are optional and not considered formal Scrum events.

Definition of Done (DoD)

#### **Key Concepts:**

1. **Definition of Done (DoD):**
   * The criteria that must be met for an increment to be completed and considered releasable.
   * Acts as quality criteria and ensures the increment integrates with the product successfully.
2. **Scrum Guide 2020 Definition:**
   * A formal description of the state of the increment when it meets the quality measures required for the product.
3. **Key Characteristics:**
   * **Transparency:** Everyone in the Scrum Team(s) must agree on and understand the DoD.
   * **Consistency:** If multiple teams work on the same product, DoD should align across teams, often following organizational conventions.
   * **Integration Testing:** Ensures compatibility with all prior increments.
   * **Continuous Improvement:** DoD evolves over time, adding higher quality standards.
4. **Evolving the DoD:**
   * Retrospectives are a good time to review and refine the DoD.
   * Additions to DoD mid-sprint must be agreed upon by the Scrum Team to avoid jeopardizing sprint goals.
   * Previous increments may need to be updated to align with new DoD standards.

#### **Example Criteria for DoD:**

* **Testing:**
  + Unit tests, integration tests, user acceptance tests, performance tests, etc.
* **Documentation:**
  + Must be adequate and verified by another developer.
* **Compliance:**
  + Must adhere to security, usability, or legislative requirements.

#### **Scrum Team and DoD:**

* All Scrum Team members should contribute to and agree on the DoD.
* The Product Owner plays a vital role in ensuring the DoD supports high-quality standards and maximizes product value.

#### **Handling New Additions to DoD:**

* New items should not disrupt the sprint goals unless agreed upon by the team.
* Changes should be documented and tasks added to the product backlog for future consideration.

#### **Commitment:**

* The DoD is a **quality commitment** to ensure increments meet releasable standards.
* Work that does not meet the DoD cannot be considered part of the increment.

#### **Practical Notes:**

* Review and refine the DoD regularly during retrospectives.
* Add new DoD tasks to the product backlog for systematic improvement.
* Avoid shifting goals mid-sprint without team consensus.

Definition of Done example

In this section, we look at examples of definitions of "done" which can be applied to both software and non-software products.

1. **Clear, Testable, Measurable Definition:**
   * The definition should be clear to everyone involved, using language that avoids ambiguity.
   * It should be testable and measurable to avoid subjectivity, making it easier to assess if the task is done.
   * The definition should be concise and achievable. It should not set unrealistic expectations for the team.
2. **Software Example:**
   * **Corporate style guide**: The product must meet branding consistency.
   * **Optimization**: Images should be optimized for fast web loading.
   * **Proofreading**: All text should be spell-checked and proofread.
   * **Code quality**: Peer-reviewed code.
   * **Testing**: Unit, integration, system, and acceptance testing.
   * **Accessibility**: Meeting accessibility standards for users with special needs.
   * **Performance testing**: Ensuring adequate performance and stress-testing to find potential limits.
   * **Architecture best practices**: Ensuring scalability and security.
   * **Documentation**: Ensuring process and product documentation is up-to-date for user understanding.
3. **Non-Software Example:**
   * **Article publication**: Ensuring style guide compliance, spelling/grammar check, and fact-checking.
   * **Feedback**: Product owner feedback on the article.
   * **Logging**: Documenting the creation of the article, including the author, sources, time of publication, etc.

The "definition of done" is a vital tool for refining the team's work as the project progresses, ensuring that all requirements are met. As the team becomes more experienced, they will be able to refine their definition to make it more practical and achievable.

### **Sprint Backlog**

The **Sprint Backlog** is a subset of the **Product Backlog**, consisting of work items selected during the **Sprint Planning** session for completion in the current sprint. Here's a breakdown of key aspects covered in this lecture:

#### **Key Components of the Sprint Backlog**

1. **Selected Product Backlog Items**:  
   Items chosen by the Scrum team (including the Product Owner and Developers) that align with the sprint goal and are needed to deliver at least one increment.
2. **Plan for Delivery**:  
   A detailed plan by the developers outlining the steps to deliver the selected items as "done" increments.
3. **Forecast of Functionality**:  
   Developers forecast the functionality needed in the next increment and identify the work required to deliver it.
4. **Continuous Improvement Items**:  
   High-priority process improvements identified in the **Sprint Retrospective** are included in the sprint backlog.

#### **Can Items Be Added to the Sprint Backlog After the Sprint Starts?**

Yes, new items can be added if they are required to achieve the sprint goal. Here’s how to handle them:

* **Who Adds Them?**Developers, who are responsible for maintaining the sprint backlog, add the new items.
* **Transparency and Estimation**:  
  New items should be transparent, detailed, and have an estimated effort and time for completion.
* **Adjustments**:  
  Unnecessary items can be removed or deferred for later sprints.

#### **Managing the Sprint Backlog**

* **Focus on the Sprint Goal**:  
  Ensure all modifications to the backlog support achieving the sprint goal.
* **Daily Updates**:  
  Maintain transparency by updating the backlog daily during **Daily Scrums**.
* **Adaptability**:  
  Be prepared to:
  + Add unforeseen items.
  + Remove items that are no longer necessary.
  + Shift focus based on emerging insights or challenges.

#### **Progress Tracking**

* **Work Status**:  
  Items move from "In Progress" to "Done" as they are completed.
* **Burndown Chart**:  
  Tracks the estimated remaining work, which should decrease steadily throughout the sprint.

By effectively managing the Sprint Backlog, the team ensures alignment with the sprint goal and maximizes productivity, even as unforeseen challenges or new requirements arise.

The **Sprint Backlog** is a dynamic, real-time representation of the work the **developers plan to accomplish** during the sprint. Here’s a breakdown of its key aspects:

### **Key Characteristics**

1. **Real-Time Visibility**:
   * The sprint backlog provides a transparent picture of the team's progress toward the sprint goal.
   * Work is often visualized through a **workflow board** (e.g., To Do → In Progress → Done).
2. **Daily Updates**:
   * The **Daily Scrum** is a critical event for the developers to discuss progress and plan work for the next 24 hours.
   * Transparency is maintained by updating the sprint backlog regularly.
3. **Flexible Structure**:
   * New work items can be added or removed as needed to meet the sprint goal.
   * Developers focus on achieving the **sprint goal**, not necessarily completing all backlog items.
4. **Columns for Workflow Management**:
   * **To Do**: Tasks planned but not started.
   * **In Progress**: Tasks currently being worked on.
   * **Done**: Completed tasks.
   * **Optional Review Column**: Adds visibility for tasks needing review before moving to Done.

### **Estimations and Velocity**

1. **Effort and Time Estimation**:
   * Work items are estimated for time and effort required, helping developers gauge workload.
   * Estimates for remaining work are recalculated daily to assess sprint progress.
2. **Burndown Chart**:
   * Tracks the remaining work and visualizes whether the sprint goal is achievable within the sprint timeline.
   * Many tools (like JIRA) automate this.
3. **Velocity**:
   * Measures how much work the team completes per sprint.
   * Improves over time, helping refine future estimations.
4. **Challenges with Estimations**:
   * Estimating complex work accurately is difficult, especially in new or evolving domains.
   * Scrum accommodates this by allowing for flexibility in the sprint backlog.

### **Focus on the Sprint Goal**

* The **primary objective** is achieving the sprint goal, not completing all backlog items.
* Adjustments to the backlog should always aim to align with and support the sprint goal.

### **Common Challenges**

1. **Unplanned Work**:
   * New information and unforeseen tasks often emerge, making initial estimates less accurate.
2. **Underestimating Time**:
   * Underestimations can lead to stress and missed sprint goals.
   * This highlights the importance of refining estimation techniques and focusing on continuous improvement.

### **Practical Tips**

* Use **workflow management tools** (e.g., JIRA, Trello) to visualize progress and manage work items.
* Incorporate a **review column** to ensure transparency and accountability in task completion.
* Regularly evaluate and improve estimation techniques to enhance planning and reduce stress.

### **Conclusion**

While the **Sprint Backlog** is simple in concept, mastering its use requires practice, flexibility, and a focus on continuous improvement. The team's ability to adapt and align their efforts with the sprint goal is the key to success.

### **The Increment**

An **Increment** in Scrum represents progress toward a broader vision or goal. Each increment builds upon previous ones, ensuring a cohesive and functional system as the product evolves.

### **Key Concepts of the Increment**

1. **Definition**:
   * An increment is a step toward achieving the **product vision** or goal.
   * It is **additive**, meaning every increment contributes to and integrates with all previous increments.
2. **Purpose in a Sprint**:
   * The **Sprint Goal** is to produce at least one increment.
   * A sprint can generate **multiple increments**, as the timebox should not limit the number of increments created.
3. **Continuous Delivery**:
   * Increments can (and often should) be released throughout the sprint, not just during the **Sprint Review**.
   * Early and frequent releases align with Agile principles, ensuring value delivery to customers without delay.

### **Definition of Done (DoD)**

1. **Commitment to Quality**:
   * The **Definition of Done (DoD)** ensures all increments meet agreed-upon quality standards.
   * DoD maintains transparency about what "done" means within the Scrum Team and to stakeholders.
2. **Transparency and Standards**:
   * DoD outlines the criteria an increment must meet to be considered ready for release.
   * Once a **Product Backlog Item** satisfies the DoD, it becomes an increment.
3. **Increment and DoD Relationship**:
   * The **Scrum Guide** emphasizes that an increment is "born" the moment a backlog item meets the DoD.
   * This underscores the importance of DoD in maintaining quality and alignment with the product vision.

### **Releasing Increments**

1. **Frequent Releases**:
   * Avoid holding increments until the **Sprint Review** for release.
   * Release increments as soon as they meet the DoD to deliver value continuously.
2. **Sprint Review Purpose**:
   * The **Sprint Review** is for assessing the work completed during the sprint, not as a bottleneck or gateway for increment releases.

### **Practical Tips**

* **Incremental and Iterative Progress**: Always aim to create increments that add value and integrate seamlessly with previous work.
* **Use the DoD Effectively**: Collaboratively define and adhere to the DoD to ensure quality and consistency.
* **Release Early and Often**: Leverage frequent releases to gather feedback and improve alignment with customer needs.

### **Conclusion**

The **Increment** is a straightforward yet vital concept in Scrum, emphasizing the continuous and quality-driven delivery of value. By focusing on creating increments that meet the DoD and embracing frequent releases, teams can effectively align with Agile principles and move closer to achieving their product vision.