Scrum in Scenario:

Background:

The DT-Global Incorporation wants to digitize and automate their HR processes (recruitment, onboarding, payroll, leave, performance management) by developing a customized **HR Management System**. They sign a contract with **iTechnology Company**, a software development firm.

Key Scrum Roles:

The first step in implementing Scrum is to **define the Scrum Team**, which consists of **three key roles** — the **Product Owner**, the **Scrum Master**, and the **Development Team** — each with distinct responsibilities that collectively ensure successful product delivery.

1. **Product Owner**: Mr. Ahmad, an HR Process Specialist from DT-Global, is appointed as the **Product Owner (PO)**. He is responsible for defining the product backlog, prioritizing requirements, and ensuring the product meets DT-Global's business needs.
2. **Scrum Master**: Ms. Laila, a senior Agile expert from iTechnology Company, facilitates the Scrum process, removes blockers, and coaches the team.
3. **Development Team**: 5 developers for instance, (2 backend, 2 frontends, 1 QA/tester) from iTechnology Company.

Product Owner:	Scrum Master:	Development Team:
• **Define Product Vision:** Clearly articulate the product goal aligned with DT-Global's HR digitization strategy. • **Manage Product Backlog:** Create, refine, and maintain the backlog in a prioritized and transparent manner. • **Prioritize Value Delivery:** Order backlog items based on business value, risk, and dependencies. • **Stakeholder Engagement:** Communicate with HR managers, employees, and leadership to gather and validate requirements. • **Clarify Requirements:** Ensure the team understands the "why" and "what" behind backlog items. • **Accept or Reject Work:** Verify whether completed backlog items meet the Definition of Done (DoD). • **Collaborate in Sprint Planning:**	• **Facilitate Scrum Events:** Organize and moderate Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective. • **Remove Impediments:** Identify and eliminate obstacles that slow down the Development Team. • **Coach Agile Mindset:** Mentor team members and the Product Owner on Scrum principles and self-organization. • **Protect the Team:** Shield the team from external interruptions and scope creep. • **Ensure Transparency:** Encourage visibility through information radiators (burn-down charts, task boards). • **Support Product Owner:** Help the PO with backlog refinement, stakeholder communication, and goal clarity.	• **Sprint Planning Participation:** Collaborate to forecast work and commit to Sprint Goals. • **Self-Organize:** Decide internally how best to accomplish work without external direction. • **Create Increments:** Design, code, test, integrate, and document features in each sprint. • **Maintain Quality:** Follow coding standards, DoD, and QA/testing procedures. • **Daily Scrum:** Inspect progress and adapt plans daily for transparency and synchronization. • **Cross-Functionality:** Collaborate across roles (backend, frontend, QA) to ensure smooth integration. • **Estimate Work:** Provide

Define Sprint Goals and clarify scope with the Scrum Team. - **Inspect and Adapt:** Participate in Sprint Reviews to gather feedback and adjust backlog priorities. - **Maintain Transparency:** Communicate progress and performance metrics to DT-Global's management. - **Align Product with Strategy:** Ensure the HR Management System aligns with organizational objectives and ROI expectations.	- **Foster Collaboration:** Promote a culture of respect, openness, and continuous improvement. - **Drive Continuous Improvement:** Facilitate retrospectives and implement actionable process improvements. - **Monitor Team Health:** Observe morale, workload balance, and communication efficiency. - **Align with PMP Practices:** Ensure alignment with project governance, reporting, and stakeholder engagement as required by DT-Global's contract.	realistic and consistent estimates for backlog items. - **Continuous Improvement:** Implement feedback from reviews and retrospectives. - **Transparency & Communication:** Keep progress visible using Scrum tools (boards, burn-downs). - **Deliver Value:** Focus on delivering high-value, working software that meets acceptance criteria.

Product Vision, Persona and Roadmap:

After forming the Scrum Team, the **first task** is for the **Product Owner** to clearly **share the product vision** with the team, ensuring everyone understands the overall purpose, goals, and value the product aims to deliver.

After the Product Owner shares the product vision, the **Product Owner and the Development Team** collaboratively **create Agile personas** to represent key user types and their needs. Based on these personas and the product vision, they then **develop the Product Roadmap**, outlining the major features, milestones, and delivery timeline that guide the product's evolution.

Product Vision Statement	Agile Persona	Product Roadmap
What Is a Product Vision Statement? A Product Vision Statement is a short, clear, and inspiring description of the future state of the product. It explains what the product will achieve, who it will serve, and why it matters. It provides direction and purpose to the entire Scrum Team and stakeholders.	**What Is an Agile Persona?** An Agile Persona is a fictional, evidence-based representation of a typical user or stakeholder of the product. It helps the Scrum Team understand who the users are, what they need, and how they will use the product. Each persona summarizes user goals, behaviors, challenges, motivations, and pain points in a human-centered way.	**What Is a Product Roadmap?** A Product Roadmap is a high-level visual summary that outlines the product's direction, key features, and delivery milestones over time. It connects the product vision to the work that the Scrum Team will deliver in upcoming releases or sprints. It shows how the product will evolve to meet business goals and user

Who Creates It?

The Product Owner (PO) creates the Product Vision Statement,
but it is developed collaboratively with input from:

- Stakeholders (management, end-users, HR staff, etc.)
- Scrum Team members (Scrum Master and Development Team)

The Product Owner is responsible for finalizing and communicating it.

Why Is It Created?

The Product Vision Statement is created to:

- Give a common understanding of what the product will achieve.
- Provide direction for all Scrum activities.
- Help the team make decisions aligned with business goals.
- Motivate and inspire the team by showing the value and impact of their work.
- Serve as a foundation for the Product Roadmap and Product Backlog.

Example — Product Vision for the DT-Global HR Management System:

For DT-Global's HR department and

Who Creates It?

Agile Personas are created collaboratively by:

- **Product Owner (Mr. Ahmad)** → Leads the process and gathers user insights.
- **Development Team** → Contributes technical and user-interaction understanding.
- **Stakeholders and Users** → Provide real data through interviews, surveys, and observation.

The **Scrum Master (Ms. Laila)** facilitates the session to ensure collaboration and focus.

Why Are Personas Created?

They are created to:

- Build empathy for real users.
- Guide feature prioritization and backlog refinement.
- Ensure that every function or design choice serves a real user's need.
- Support the Product Vision and Roadmap with a user-centered perspective.
- Prevent the team from building features based on assumptions.

Example Agile Personas for

needs (defined in personas).

Who Creates It?

The **Product Owner (Mr. Ahmad)** leads the creation of the roadmap, but it is built collaboratively with:

- **Scrum Master (Ms. Laila)** → ensures the roadmap remains realistic and agile.
- **Development Team** → provides effort estimates and technical feasibility.
- **Stakeholders (DT-Global HR leadership)** → provide business priorities and feedback.

Why Is It Created?

The roadmap is created to:

- Translate the vision into a strategic plan.
- Show how the product will deliver value incrementally.
- Align the Scrum Team and stakeholders on priorities and timing.
- Help in sprint planning and backlog refinement.
- Communicate progress and direction transparently.

Example:

Month	Focus Area	Expected Outcome
Month	Recruitment &	Digital recruitment and

employees who need a faster, more accurate, and transparent way to manage HR operations, the DT-Global HR Management System is a digital platform that automates recruitment, onboarding, payroll, leave, and performance management. Unlike manual spreadsheets and paperwork, this system offers real-time access, accuracy, and efficiency — empowering DT-Global to manage its workforce with confidence and agility.

the DT-Global HR Management System:

Persona 1 — Fatima (HR Officer)

- **Goal:** Simplify HR operations by automating leave and payroll management.
- **Pain Point:** Wastes hours each week compiling attendance and approvals manually.
- **Need:** A dashboard for real-time data and auto-generated reports.

Persona 2 — Ahmad (HR Manager)

- **Goal:** Track employee performance and manage recruitment efficiently.
- **Pain Point:** Lacks an overview of staff performance metrics.
- **Need:** Performance management and recruitment tracking modules.

Persona 3 — Zainab (Employee)

- **Goal:** Submit leave, view payslips, and update personal data easily.
- **Pain Point:** Often waits days for HR confirmation and updates.
- **Need:** Employee self-service portal with instant notifications.

Persona 4 — Mr. Farid (Finance Officer)

- **Goal:** Ensure accurate payroll synchronization with finance

1	Onboarding	onboarding module live
Month 2	Payroll & Leave	Automated payroll and leave management integrated
Month 3	Performance & Self-Service	Employee portal and evaluation tools activated
Month 4	Integration & Go-Live	Full HRMS release, staff training, and feedback collection

	records. - **Pain Point:** Re-checking calculations due to mismatched HR data. - **Need:** Integration between HR and finance systems.	

Product Scope and Backlog:

A **Product Backlog** is a **master list of all user stories, features, and requirements** that define what needs to be developed for the product. It serves as the **single source of truth** for everything the Scrum Team will build, prioritize, and refine throughout the project. For creating product backlog, consider the following steps.

Step 1:
Begin with the Product Vision:

The Product Owner revisits the approved product vision and ensures the Scrum Team clearly understands the product's purpose, expected benefits, and target users. This ensures that all backlog items will stay aligned with the overall business goal.

Step 2:
Identify and Define Epics:

The Product Owner lists the large business goals called **Epics**. Each Epic represents a broad area of functionality that supports the product vision and cannot be completed in a single sprint. For example, "Recruitment Management," "Onboarding," or "Payroll Automation."

Step 3:
Break Epics into Features:

Each Epic is divided into smaller and more specific functionalities known as **Features**. Features deliver measurable value to users and are often achievable within one or a few sprints. For example, under "Recruitment Management," features may include "Job Posting," "Candidate Tracking," and "Interview Scheduling."

Step 4:
Decompose Features into User Stories:

The Product Owner and Development Team collaboratively break down each feature into **User Stories**.
A User Story describes one small, user-focused requirement using this format:
"As a [user], I want [goal], so that [benefit]."
These stories represent the smallest units of work that deliver user value.

Note: Each User Story is checked using the **INVEST** model — it must be **Independent, Negotiable, Valuable, Estimable, Small, and Testable.** This ensures each story is clear, actionable, and can be completed within a sprint.

Step 5:

Define Acceptance Criteria (Definition of Done): DoD:

For every User Story, the team defines **Acceptance Criteria** — the specific conditions that must be met for the story to be considered complete. This provides a shared understanding of “done” and guides testing and validation. The team defines a Definition of Done, a checklist confirming all development, testing, documentation, and quality checks are completed before any story is marked as done.

Step 6:

Prioritize the Backlog:

The **Product Owner** prioritizes all items in the Product Backlog based on **business value**, **user impact**, and **dependencies**. The highest-value, most urgent, and most ready items appear at the top.

Step 7:

Establish Definition of Ready (DoR):

The Scrum Team agrees on a **Definition of Ready**, which outlines when a User Story is considered ready to be included in a sprint. Stories that meet this definition are clear, estimated, and testable.

Epic	Feature	User Story
What Is an Epic? An Epic is a large body of work that represents a major business goal or capability. It is too big to be completed in a single sprint and must be broken down into smaller parts (features and user stories). **Example:** “Automate HR operations in DT-Global” or “Develop the Recruitment Management Module.” **Who Creates It:** • Product Owner (leads) • Scrum Team (support by providing input and technical insights)	**What Is a Feature?** A Feature is a distinct piece of functionality that delivers a specific value to the user or organization. It is smaller than an Epic but may still require several user stories to complete. **Example:** Under the Epic *Recruitment Management*, possible features could include: • Job Posting and Advertisement • Candidate Tracking System • Interview Scheduling **Who Creates It:** • Product Owner — defines features from business and	**What Is a User Story?** A User Story is the smallest, actionable item in Agile development. It represents one requirement from the user’s perspective, small enough to be designed, developed, tested, and completed within a single sprint. **Format:** **As a [user role], I want [goal or action], so that [benefit or value].** **Example:** As an HR officer, I want to post a job opening online so that candidates can apply through the system. **Who Creates It:** • Product Owner — writes

How It's Created: - Derived directly from the Product Vision and Roadmap. - Written during backlog creation or early refinement. - Defined as a high-level statement describing the intended outcome or function. **Purpose:** - Provide a strategic view of the product. - Align the team around large, high-value goals. - Organize and group multiple related features or user stories.	user needs. - Development Team — contributes by identifying technical or supporting features. **How It's Created:** - The Product Owner breaks down each Epic into logical, valuable functions. - Each Feature focuses on a complete and testable part of the system that provides visible value. **Purpose:** - Bridge between Epics (strategic) and User Stories (operational). - Help in planning releases and structuring the Product Backlog. - Ensure that functionality is comprehensive and cohesive.	the initial user stories based on product vision and user needs. - Development Team — refines and estimates the stories for clarity and feasibility. - Stakeholders/Users — provide insights and feedback. **How It's Created:** 1. The Product Owner reviews Features and identifies what specific user interactions or functionalities are needed. 2. Each function is written as a User Story describing what the user wants to do and why. 3. The Scrum Team collaborates to add acceptance criteria (conditions of satisfaction). **Purpose:** - Represent a clear, testable, and valuable piece of work. - Keep focus on user value and outcomes rather than technical tasks. - Serve as the foundation for sprint planning and delivery.

Estimation in Agile:

After creating the Product Backlog, the Scrum Team proceeds to **estimate the User Stories**. Estimation helps the team understand the **effort, complexity, and time** required to complete each story. It is usually done using **Story Points** or other Agile estimation techniques, allowing the team to plan sprints realistically and deliver value at a sustainable pace.

What Is a Story Point?

A Story Point is a unit of relative measure used by Agile and Scrum teams to estimate the total effort required to complete a User Story.
It does not represent time (like hours or days); instead, it reflects how complex, large, and risky a task is compared to others.

Story Points help the team understand how difficult a story is relative to others — not how long it will take.

What Story Points Measure?

When estimating Story Points, the team considers **three main factors:**

1. **Complexity:**
   How difficult is the work technically or logically?
   Example: integrating a third-party API is more complex than creating a static page.
2. **Effort:**
   How much work is involved? How many steps or sub-tasks are required?
3. **Uncertainty / Risk:**
   How familiar is the team with this task? Are there unknowns, new tools, or unclear requirements?

How Story Points Work?

Story Points provide a **relative comparison** between different stories.
If one story is assigned **2 points** and another **4 points**, the team agrees that the second story requires roughly **twice the effort or complexity**.

Agile teams commonly use the **Fibonacci sequence** (1, 2, 3, 5, 8, 13, 21…), because it captures increasing uncertainty as work grows larger.

Who Assigns Story Points?

Story Point estimation is done collaboratively by the Scrum Team, not by the manager or Product Owner.

- **Development Team:** Estimates and agrees on Story Points.
- **Product Owner:** Clarifies requirements and priorities but doesn’t assign points.
- **Scrum Master:** Facilitates the estimation session and ensures consensus.

Teams often use **Planning Poker** or **Affinity Estimation** techniques to assign points democratically.

Why Scrum Teams Do Not Use Exact Time Estimates?

1. **Time Estimates Create False Precision:** Predicting in hours or days gives an illusion of accuracy — but software development includes unknowns and unpredictable obstacles (bugs, dependencies, rework, or changing requirements).
   Story Points accept uncertainty as part of the process.
2. **People Work at Different Speeds:** Not all developers have the same speed or experience. Story Points measure team effort, not individual speed, so estimates remain consistent regardless of who performs the work.
3. **Time Estimation Causes Pressure:** When tasks are estimated in hours, teams often feel pressured to meet the clock instead of focusing on quality. Story Points reduce this stress and promote focus on value delivery rather than deadlines.
4. **Hours Focus on Output, Not Outcome:** Time-based planning encourages checking boxes ("8 hours done") instead of asking whether the user's goal is achieved. Story Points emphasize value delivered, not time spent.
5. **Story Points Enable Continuous Improvement:** By tracking velocity (the average number of story points completed per sprint), the team learns its real capacity over time. Velocity becomes the basis for sprint planning and forecasting, far more accurate than guessing hours upfront.

Common Estimation Techniques in Scrum:

T-Shirt Sizing:

A **high-level estimation method** used when details are still unclear.
The team assigns sizes like **XS, S, M, L, XL** to indicate relative effort.

How It Works:

- XS → Very small story
- S → Small story
- M → Medium story
- L → Large story
- XL → Very large story

Once refined, each size can later be converted into Story Points (e.g., S=3, M=5, L=8).

When to Use:

- Early stages of a project.
- During roadmap or release planning.

Planning Poker (Scrum Poker):

Planning Poker is a consensus-based technique for agile estimating. It is a fun and engaging way for teams to apply relative estimates to planned work.

To start a poker planning session, the product owner or customer reads one of the desired user stories or describes a feature to the estimators.

Each estimator has a physical or virtual deck of cards. These Planning Poker cards display values like 1, 2, 3, 5, 8, 13, 20, 40 and 100 (the modified Fibonacci sequence). The values represent the number of story points, ideal days, or other units in which the team estimates.

The estimators discuss the feature, asking questions of the product owner as needed. When the feature has been fully discussed, each estimator privately selects one card to represent their estimate. The estimators then reveal all of their cards at the same time.

If all estimators all selected the same value, that becomes the estimate. If not, the estimators discuss their estimates. The high and low estimators, especially, should share their reasons. After further discussion, each estimator reselects an estimate card, and all cards are again revealed at the same time.

The team repeats the process until they achieve consensus on an estimate. If they lack enough information to agree on an estimate, the estimators can defer a particular item until those details can be acquired.

When do teams use Planning Poker?

Team can use planning poker anytime they need to estimate product backlog items.

Teams use Planning Poker when the initial product backlog is created. Most teams hold a story-writing workshop to write and estimate the initial product backlog. To understand the scope and size of a project, the whole team is encouraged to write user stories and create initial estimates of the work. The whole team includes the developers (team members), Scrum Master, and product owner.

Teams also use Planning Poker as new product backlog items are added. This activity, called product backlog refinement, typically happens about once per iteration.

Why is Planning Poker Used in Agile?

1. Promotes teamwork and shared understanding.
2. Prevents bias in estimation decisions.
3. Encourages equal participation from all members.
4. Builds consensus through discussion.
5. Makes estimation fun and engaging.
6. Improves accuracy through collective intelligence.
7. Increases transparency in planning.
8. Helps align different perspectives across roles.

Starting Sprints:

As described in the Scrum Guide, Sprints are the heartbeat of Scrum, where ideas are turned into value. The Sprint is the Scrum event that encompasses all of the other Scrum events.

They are fixed length periods of work that last one month or less to create consistency and ensure short iterations for feedback in order to inspect and adapt both how work is done and what is being worked on. If cycles are longer, then the spirit of frequent feedback cycles can be lost. Longer Sprint may also get too complex and may increase risk. A new Sprint starts immediately after the conclusion of the previous Sprint.

All the work necessary to achieve the Product Goal, including Sprint Planning, Daily Scrums, Sprint Review, and Sprint Retrospective, happen within Sprints.

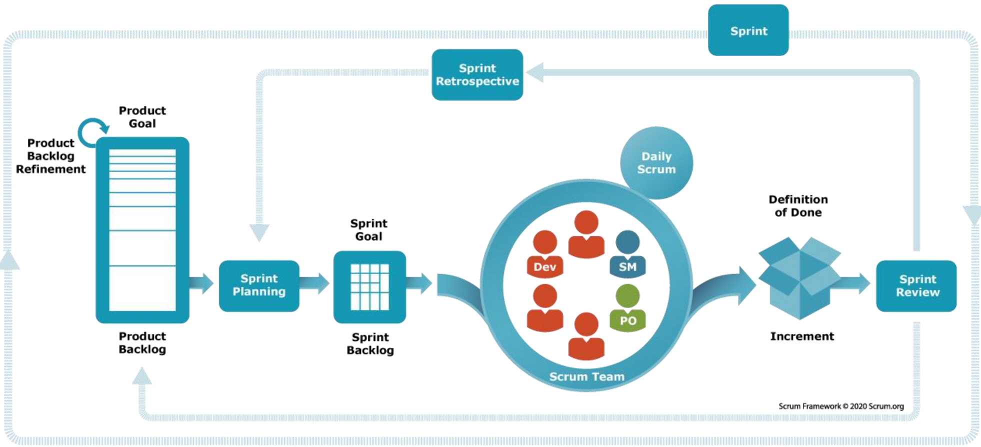
During the Sprint:

- No changes are made that would endanger the Sprint Goal
- Quality does not decrease
- The Product Backlog is refined as needed
- Scope may be clarified and renegotiated with the Product Owner as more is learned

Sprints enable predictability by ensuring that the Scrum Team inspects and adapts toward the **Product Goal** and **Sprint Goal** each Sprint:

- **Product Goal**- The Product Goal describes a future state of the product which can serve as a target for the Scrum Team to plan against. The Product Goal is in the Product Backlog. The rest of the Product Backlog emerges to define "what" will fulfill the Product Goal. The Product Goal is the long-term objective for the Scrum Team. They must fulfill (or abandon) one objective before taking on the next.
- **Sprint Goal** - The Sprint Goal is the single objective for the Sprint. Although the Sprint Goal is a commitment by the Developers, it provides flexibility in terms of the exact work needed to achieve it. The Sprint Goal also creates coherence and focus, encouraging the Scrum Team to work together rather than on separate initiatives.

You and your Scrum Team can choose the Sprint length that works for your team, as long as you keep it one month in length or less. Shorter Sprints generate more learning cycles and limit risk to a smaller time frame. Each Sprint may be considered a short stretch of work.



Scrum Framework © 2020 Scrum.org

Step 1: Sprint Zero – Foundation & Setup:

Purpose: Prepare everything required before development begins.
Why Needed: It ensures the Scrum Team (Product Owner – Mr. Ahmad, Scrum Master – Ms. Laila, and the Development Team from iTechnology) has all tools, environments, and priorities ready to start real work.
What They Do:

- Finalize the Product Vision, Agile Personas, and Product Roadmap for DT-Global's HR Management System.
- Establish development tools (Jira, Git, Test Server, CI/CD pipeline).
- Define Definition of Ready (DoR) and Definition of Done (DoD).
- Prepare technical architecture and security framework.
- Identify initial top-priority backlog items (e.g., "Login Module", "Recruitment Workflow").

Step 2: Backlog Refinement:

Purpose: Transform the product roadmap into clear, prioritized, ready-to-develop stories.
What They Do:

- Mr. Ahmad (PO) refines Epics like **Recruitment**, **Payroll**, and **Leave Management** into Features and User Stories.
- Ms. Laila (SM) facilitates team discussions to clarify acceptance criteria.
- The Development Team estimates Story Points and identifies dependencies.
  Duration: Ongoing activity, usually 1 session per week throughout the project.

Step 3: Sprint Planning:

Purpose: Decide what work will be delivered in the upcoming Sprint.
What They Do:

- The Scrum Team reviews the **Product Backlog** and selects items for **Sprint 1** (e.g., "User Login & Authentication").
- Define a clear **Sprint Goal** such as: *"Deliver a secure login and user dashboard."*
- Break down stories into smaller tasks with technical details.
  Duration: About half a day for a two-week Sprint.

Step 4: Sprint Execution (Development Cycle):

Purpose: Build, test, and deliver a working increment of the HR Management System.
What They Do:

- Developers code, integrate, and test assigned stories.
- The Scrum Master facilitates **Daily Scrum meetings** (15 minutes).
- The team updates the task board, tracks progress, and resolves impediments quickly.
  Example Work: Implement user login functionality and employee dashboard UI.
  Duration: 2 weeks per Sprint.

Step 5: Sprint Review:

Purpose: Demonstrate the completed HR system increment to DT-Global stakeholders.
What They Do:

- The team shows working features (e.g., login page, basic navigation).
- Mr. Ahmad gathers feedback from DT-Global's HR department.
- Product Backlog is updated based on stakeholder comments.
  Duration: 1 to 2 hours per Sprint.

Step 6: Sprint Retrospective:

Purpose: Reflect on the Sprint and identify improvements.
What They Do:

- The team discusses what went well, what didn't, and what to improve.
- Example output: *"Improve test coverage before code review"* or "*Reduce meeting time overlaps*."
  Duration: 1 hour for a two-week Sprint.

Step 7: Sprint 2 → Sprint N – Continuous Delivery:

Purpose: Continue delivering high-value increments of the HR System through iterative Sprints.
What They Do:

- Each Sprint adds new modules (e.g., Recruitment, Onboarding, Payroll, Performance Management).
- Maintain fixed timeboxes and stable velocity.
- Keep refining backlog and integrating feedback from previous reviews.

Step 8: Sprint H – Hardening & Release Preparation:

Purpose: Finalize the product for deployment to DT-Global.
What They Do:

- Conduct **integration**, **system**, and **security testing** across all modules.
- Validate non-functional requirements (performance, usability, compliance).
- Prepare user manuals and training for HR staff.
  Outcome: A fully tested HR Management System ready for go-live.
  Duration: Same as one Sprint or shorter depending on quality.

Step 9: Project Closure and Handover:

Purpose: Officially close the Scrum project and transition to operations.
What They Do:

- Conduct final review with DT-Global leadership.

- Handover code, documents, and training materials.
- Celebrate success and record lessons learned for future projects.

Scrum Ceremonies and Artifacts:

Scrum Ceremonies	Scrum Artifacts
Scrum ceremonies are the formal, time-boxed meetings that bring rhythm, transparency, and collaboration to every Sprint. They ensure the Scrum Team (Product Owner – Mr. Ahmad, Scrum Master – Ms. Laila, and the iTechnology Development Team) stays aligned, focused, and continuously improving. **There are 4 Scrum Ceremonies** **1. Sprint Planning** **Purpose:** To decide *what* will be delivered in the upcoming Sprint and *how* it will be done. **Who Attends:** Product Owner, Scrum Master, Development Team. **What Happens:** • Review the Product Backlog. • Select high-priority User Stories for the Sprint. • Define a clear **Sprint Goal** (e.g., "Develop a functional Employee Onboarding Module"). • Break stories into smaller development tasks. **Output:** • **Sprint Goal** • **Sprint Backlog** (the list of selected User Stories for this Sprint) **2. Daily Scrum (Daily Stand-Up)** **Purpose:** To synchronize the team's daily work and identify impediments quickly. **Who Attends:** Development Team (Scrum Master facilitates; Product Owner may observe). **What Happens:**	Scrum Artifacts are the key documents or outputs that provide transparency and track progress toward the product goal. **1. Product Backlog** **Definition:** A **master list** of all features, requirements, and enhancements the product might need. **Owner:** Product Owner (Mr. Ahmad). **Example for Scenario:** User login, recruitment workflow, payroll automation, leave tracking, performance evaluation. **Purpose:** To ensure all possible product work is recorded, prioritized, and visible. **2. Sprint Backlog** **Definition:** A **subset of the Product Backlog** selected for a specific Sprint, plus the plan for delivering it. **Owner:** Development Team. **Example for Scenario:** Sprint 1 might include user login, employee profile, and authentication module. **Purpose:** To focus the team on what will be delivered in the current Sprint. **3. Increment** **Definition:** The **sum of all completed work** from the current Sprint plus all previous Sprints — a working version of the product that meets the **Definition of Done**.

Each member answers three questions:

1. What did I do yesterday?
2. What will I do today?
3. Are there any impediments?

Duration:
15 minutes maximum.

Output:
Updated Sprint progress and early problem visibility (Identify obstacles or impediments).

3. Sprint Review

Purpose:
To inspect the product increment and gather feedback from stakeholders.

Who Attends:
Scrum Team + Stakeholders (e.g., DT-Global HR staff, management).

What Happens:

- Demonstrate completed functionality (e.g., show the working Payroll or Leave Module).
- Discuss what went well and what still needs to improve.
- Update the Product Backlog based on stakeholder feedback.

Output:
Improved backlog, receiving feedback and stakeholder alignment.

4. Sprint Retrospective

Purpose:
To reflect on the Sprint process and find ways to improve team performance.

Who Attends:
Scrum Team only.

What Happens:

- Discuss what went well, what didn't, and what can be improved.

Example for Scenario:
After Sprint 3, the increment might include:

- Functional recruitment module,
- Employee onboarding workflow, and
- Basic payroll configuration.

Purpose: To deliver real, usable value at the end of every Sprint.

• Select a few actionable improvements for the next Sprint. **Example for Scenario:** • “Improve code review process before testing.” • “Shorten internal approval chain for leave workflow design.” **Output:** List of agreed improvements to apply next Sprint.	

Reporting in Scrum:

Agile reporting is the process of providing transparent, frequent, and visual updates on project progress, team performance, and product value delivery. Unlike traditional reporting, Agile focuses on working software, collaboration, and outcomes, not lengthy documents or status reports.

Agile reports help the Scrum Team, Product Owner, and stakeholders (like DT-Global management) see how much value has been delivered, what remains, and how the team is performing — all in real time.

1. Sprint Burndown Chart

The Sprint Burndown Chart shows the team’s daily progress by tracking the amount of work remaining in the Sprint against time, helping Ms. Laila (Scrum Master) and the iTechnology team see whether they are on schedule to achieve the Sprint Goal for the HR module and make timely adjustments if progress falls behind.

2. Burnup Chart

The Burnup Chart illustrates how much work has been completed over time compared to the total project scope, allowing Mr. Ahmad (Product Owner) to clearly present the incremental value delivered to DT-Global during each Sprint and to adapt plans easily when scope or priorities change.

3. Velocity Chart

The Velocity Chart measures how many Story Points the Scrum Team completes in each Sprint, helping the iTechnology developers and Mr. Ahmad predict future capacity—such as maintaining an average of 25 points per Sprint—and plan upcoming HR Management System modules more accurately and confidently.

4. Cumulative Flow Diagram (CFD)

The Cumulative Flow Diagram provides a visual overview of how work items move through stages such as To Do, In Progress, Testing, and Done, enabling Ms. Laila to detect bottlenecks early—for example, slow QA cycles in the Payroll module—and maintain a stable, efficient workflow throughout development.

--- End of Scrum ---