**Jira by Altassian**

Jira is a project management tool developed by Atlassian that's primarily used for issue tracking, bug tracking, and agile project management.

**Key Uses of Jira:**

1. **Issue/Bug Tracking**: Teams use Jira to track bugs or tasks from identification to resolution.
2. **Agile Project Management**: Supports agile frameworks like:
   * **Scrum**: Sprint planning, backlog grooming, burndown charts.
   * **Kanban**: Boards to visualize workflow and limit work in progress.
3. **Custom Workflows**: You can create workflows that match your team’s process.
4. **Roadmapping & Planning**: Helps in planning software releases, tracking progress, and managing deadlines.
5. **Reporting**: Built-in reports (like velocity charts, sprint reports) help teams measure performance.
6. **Integration**: Jira integrates with tools like **Confluence, Bitbucket, GitHub, Slack, and CI/CD tools**.

**Who Uses Jira?**

* Software developers
* Project managers
* QA teams
* IT support teams
* Business teams (via Jira Work Management)

**Jira Products:**

* **Jira Software** – for software development teams.
* **Jira Service Management** – for ITSM and support teams.
* **Jira Work Management** – for business teams (marketing, HR, finance)

**Project, Workflow and Issues**

**1. Project**

*A Project in Jira is a container that holds a group of issues related to a specific goal, product, or team initiative.*

**Key Points:**

* Projects can represent software development efforts, service desks, marketing campaigns, etc.
* Each project has a key (e.g., ABC) that prefixes its issues (e.g., ABC-123).
* You can customize:
  + Permissions (who can do what)
  + Workflows
  + Issue types
  + Screens and fields

**Example:**

*A software team might have a project called "Mobile App Development", with issues like bug reports, tasks, and user stories.*

**2. Issues**

*An Issue is the* ***basic unit of work*** *in Jira.*

**Common Issue Types:**

* **Story** – A user requirement or feature
* **Task** – A general piece of work
* **Bug** – A defect or error in the system
* **Epic** – A large body of work that includes multiple stories or tasks
* **Sub-task** – A smaller part of a task or story

**Issue Structure:**

Each issue typically contains:

* A **summary**
* **Description**
* **Assignee**
* **Status** (e.g., To Do, In Progress, Done)
* **Priority**
* **Labels**, **attachments**, **comments**, etc.

*Example: An issue like APP-101 could be "Fix login button bug".*

**3. Workflow**

*A Workflow* ***defines the lifecycle of an issue*** *— how it moves from start to completion.*

**Workflow Components:**

* **Statuses**: The stages an issue can be in (e.g., "To Do", "In Progress", "In Review", "Done")
* **Transitions**: The paths/issues take between statuses (e.g., "Start Progress", "Resolve Issue")
* **Conditions, Validators, Post Functions**: Rules and automation that control how transitions work

**Workflow Types:**

* **Simple** (To Do → In Progress → Done)
* **Complex** (with approvals, reviews, multiple paths)

Example: A bug might move from:  
Open → In Progress → Code Review → Testing → Closed

**How They Work Together:**

| **Component** | **Role in Jira** |
| --- | --- |
| **Project** | Organizes and groups related work |
| **Issue** | Represents individual tasks/bugs/features |
| **Workflow** | Defines how issues move through statuses in a project |
| **Agile** |  |

**1. Introduction to Agile Methodology**

Agile methodology is a modern, flexible approach to software development and project management that ***focuses on iterative development, collaboration, flexibility, and delivering high-value outcomes quickly.*** Unlike traditional methodologies like Waterfall, which are linear and rigid,

***Agile allows teams to respond to changes in real time, delivering small, working chunks of software in a shorter timeframe.***

**2. The Agile Manifesto**

The Agile Manifesto, created in 2001 by a group of 17 software developers, emphasizes four key values and twelve principles to guide the Agile process.

**Agile Values:**

1. Individuals and interactions over processes and tools.
2. **Working software over comprehensive documentation.**
3. Customer collaboration over contract negotiation.
4. Responding to change over following a plan.

**Agile Principles:**

1. Customer satisfaction through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development.
3. Deliver working software frequently, with a preference for a shorter timescale.
4. Business stakeholders and developers must work together daily throughout the project. (Daily Scrum meetings – 15 minutes)
5. Build projects around motivated individuals, provide the environment and support they need, and trust them to get the job done.
6. Face-to-face communication is the most effective and efficient method of conveying information within a development team.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development, with teams being able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity—the art of maximizing the amount of work not done—is essential.
11. Self-organizing teams are the best way to develop the desired architectures and designs.
12. Regular reflection on how to become more effective, followed by adjustments to the team’s behavior accordingly.

**3. Agile Frameworks**

Agile includes various frameworks to implement its methodology.

Some of the most popular frameworks are:

**Scrum**

* Scrum is an Agile framework that works in iterative cycles **called Sprints**, typically lasting **2-4 weeks.** Scrum focuses on defining roles, ceremonies, and artifacts.
* **Roles:**
  + **Product Owner:** Responsible for maximizing product value by managing the product backlog.
  + **Scrum Master:** Facilitates the Scrum process, removes obstacles, and ensures Scrum is followed.
  + **Development Team:** Responsible for delivering the product increment.

**Kanban**

* Kanban is a flow-based Agile method that focuses on visualizing the work process, limiting work in progress (WIP), and continuously improving efficiency.
* There are no fixed-length sprints like Scrum; work items are pulled through the system as capacity allows.

**Extreme Programming (XP)**

* XP emphasizes technical excellence and collaboration. It focuses on practices such as pair programming, continuous integration, and frequent releases to improve software quality.

**Lean**

* Lean development focuses on eliminating waste, optimizing processes, and delivering value faster. It draws on Lean manufacturing principles.

**4. Key Concepts of Agile Methodology**

4.1. **MOST IMPORTANT KEY CONCEPT** - Iterative Development

Agile development follows an iterative process, meaning the product is developed in small, manageable **chunks called increments or iterations.** Each iteration delivers a working product increment that can be tested and reviewed by stakeholders.

**4.2. Continuous Feedback**

One of the central tenets of Agile is the continuous feedback loop. Stakeholders are involved regularly in the process, ensuring the product is meeting their needs and can be adjusted based on their feedback.

**4.3. Self-Organizing Teams**

Agile promotes self-organizing teams that are empowered to make decisions, solve problems, and adapt to changes without heavy top-down management.

**4.4. Flexibility and Adaptability**

The Agile methodology allows teams to respond to changes in requirements and priorities, often through flexible planning and incremental delivery. This is in stark contrast to the more rigid structure of traditional Waterfall methods.

**5. Agile Roles**

In Agile, particularly within frameworks like Scrum, there are several important roles that support the development process:

**5.1. Product Owner**

* The Product Owner is responsible for managing the product backlog, which is a list of all desired features and enhancements. The Product Owner is also the point of contact for stakeholders and ensures the team is working on the most valuable features.

**5.2. Scrum Master**

* The Scrum Master ensures that Scrum practices are followed, helps remove impediments that block progress, and **acts as a coach** **to the development team** **and the Product Owner.** The Scrum Master also facilitates Scrum ceremonies.

**5.3. Development Team**

* The Development Team consists of **cross-functional members** who are responsible for building and delivering the product increments. They are self-organizing and collaborate daily to meet the sprint goals.

**6. Scrum Ceremonies**

Scrum provides a set of ceremonies (meetings) that keep teams aligned and ensure transparency.

**These include:**

**6.1. Sprint Planning**

* At the beginning of each Sprint, the Scrum team gathers to plan the work. The Product Owner presents the top items from the product backlog, and the Development Team estimates and commits to what can be accomplished in the upcoming Sprint.

**6.2. Daily Scrum (Stand-Up)**

* A daily 15-minute meeting where team members discuss what they accomplished, what they plan to work on, and any impediments they’re facing. It keeps the team aligned and focused.

**6.3. Sprint Review**

* At the end of each Sprint, the team demonstrates the working product increment to stakeholders, who provide feedback.

**6.4. Sprint Retrospective**

* After the Sprint Review, the team holds a retrospective meeting to reflect on the process, discuss what went well, identify areas for improvement, and develop action plans for the next Sprint.

**7. Agile Artifacts**

In Scrum and other Agile frameworks, **artifacts are used to track progress and ensure transparency.**

***The key Agile artifacts are:***

**7.1. Product Backlog**

* The Product Backlog is a prioritized list of features, enhancements, fixes, and requirements for the product. The Product Owner maintains and prioritizes this backlog.

**7.2. Sprint Backlog**

* The Sprint Backlog contains a subset of items from the Product Backlog that the team has committed to completing during the current Sprint.

**7.3. Increment**

* The Increment is the potentially shippable product that the team delivers at the end of each Sprint. It includes all completed backlog items and should meet the **Definition of Done (DoD).**

**8. Agile Estimation and Velocity**

*Agile teams use estimation techniques to plan and track progress.*

Common estimation methods include:

**8.1. Story Points**

* Story Points are a unit of measure used to estimate the effort required to complete a user story or task. Teams use relative estimation to assign story points based on complexity, effort, and risk.

**8.2. Velocity**

* Velocity is a measure of **how much work a team can complete in a Sprint.** It is typically calculated based on the number of story points completed during the Sprint.

**9. Benefits of Agile**

*Adopting Agile methodology brings numerous benefits, including:*

**9.1. Faster Delivery**

* Agile focuses on delivering small, working pieces of software quickly, allowing teams to release frequently and get feedback early from the stakeholder.

**9.2. Improved Customer Satisfaction**

* By involving customers regularly (meetings with the stakeholders daily) and responding to their feedback, Agile teams can better align the product with user needs and expectations.

**9.3. Flexibility and Adaptability**

* Agile teams can adapt to changing requirements, allowing them to pivot as necessary in response to market conditions or customer demands.

**9.4. Increased Collaboration**

* Agile fosters a culture of collaboration among team members and between developers and stakeholders, improving communication and decision-making.

1. **Challenges in Agile Adoption**

*While Agile brings many benefits, it also presents some challenges:*

1. **Resistance to Change:** Teams may be reluctant to abandon traditional approaches and adopt Agile methods.
2. **Lack of Proper Training:** Teams and stakeholders need adequate training to effectively implement Agile.
3. **Scope Creep:** Without proper backlog management, Agile teams can face scope creep due to continuous requests for changes.
4. **Inconsistent Practices:** Agile may not be implemented consistently across all teams or departments.

**Summary**

Agile methodology provides an effective framework for teams to deliver high-quality software and products quickly while adapting to changes along the way.

By focusing on customer collaboration, iterative development, and continuous feedback, Agile fosters a dynamic environment that enhances team collaboration and drives better results.

**Scrum Framework – Roles and Key Responsibilities**

In the **Scrum framework**, there are three primary roles: **Product Owner**, **Scrum Master**, and the **Development Team**. Each role has specific responsibilities to ensure the team delivers high-quality, valuable software through iterative progress.

**1. Product Owner**

The Product Owner (PO) is responsible for maximizing the value of the product resulting from the work of the Development Team.

**Key Responsibilities:**

* **Defines and prioritizes the Product Backlog** based on business value and customer needs.
* *Acts as the voice of the customer and key stakeholders.*
* **Ensures clarity of backlog items**, so the team understands what needs to be done.
* **Accepts or rejects work results** during Sprint Reviews.
* **Collaborates closely** with stakeholders, customers, and the Development Team.
* **Makes decisions** about what features will be built and in what order.

**2. Scrum Master**

The Scrum Master (SM) is **a servant-leader** who ensures the Scrum process is followed and removes impediments that affect the team’s performance.

**Key Responsibilities:**

* **Facilitates Scrum ceremonies**: Daily Scrum, Sprint Planning, Sprint Review, and Sprint Retrospective.
* **Coaches the team** in self-organization and cross-functionality.
* **Removes impediments** to the Development Team’s progress.
* **Shields the team from external interruptions and distractions.**
* **Guides the Product Owner** and organization in effective Scrum adoption.
* **Ensures Scrum principles** are understood and enacted by all team members.

**3. Development Team**

The Development Team consists of cross-functional professionals who build and deliver a potentially releasable Increment of the product at the end of each Sprint.

**Key Responsibilities:**

* **Self-organizes** to complete the Sprint Backlog.
* **Delivers a "Done" product increment** each Sprint.
* **Participates in all Scrum events** and collaborates actively.
* **Estimates and plans work** during Sprint Planning.
* **Improves continuously** through Sprint Retrospectives.
* **Owns the quality** of the product increment.

**Characteristics:**

* Cross-functional (includes all skills needed to deliver the product increment).
* No sub-teams or titles—everyone is collectively responsible.
* Typically, 3–9 members.

Agile Methodology

**Which of the following is NOT one of the four Agile values?**  
A. Working software over comprehensive documentation  
B. **Contract negotiation over customer collaboration**  
C. Responding to change over following a plan  
D. Individuals and interactions over processes and tools

**2. Agile principles promote:**  
A. Following strict processes  
B. Detailed upfront planning  
C. **Customer collaboration throughout the project**D. Contract-based relationships

**3. According to the Agile Manifesto, what is valued more?**  
A. Tools over individuals  
B. Processes over working software  
C. **Working software over comprehensive documentation**D. Documentation over collaboration

**4. Which principle emphasizes sustainable development?**  
A. Simplicity  
B. Technical excellence  
C. **Agile processes promote sustainable development**  
D. Face-to-face conversation

**5. Which of the following best describes the Agile approach to documentation?**  
A. No documentation is required  
B. Only legal documents are produced  
C. Documentation is created only at the end  
D. **Just enough documentation to meet the need/becz feedback**

**6. What is the maximum timebox for a Sprint?**  
A**. 1 week**  
B. 2 weeks  
C. **4 weeks**  
D. 6 weeks

**7. Who owns the Product Backlog?**  
A. Scrum Master  
B. **Product Owner**  
C. Development Team  
D. Project Manager

**8. What is the purpose of the Sprint Review?**  
A. Plan the next sprint  
B**. Inspect the increment and adapt the Product Backlog**  
C. Update the Scrum Board  
D. Assign tasks

**9. Which Scrum event is time-boxed to 15 minutes?**  
A. Sprint Review  
B. Sprint Planning  
C. **Daily Scrum**  
D. Sprint Retrospective

**10. What happens if the work is not finished at the end of the Sprint?**  
A. It is canceled  
B. It is removed from the product  
C. **It is reviewed and re-estimated in the next Sprint**  
D. It is ignored

**11. Who ensures Scrum is understood and enacted?**  
A. Product Owner  
B. **Scrum Master**  
C. Project Manager  
D. Stakeholder

**12. Who is responsible for maximizing the value of the product?**  
A. Scrum Master  
B. Developer  
C**. Product Owner**  
D. Project Sponsor

**13. Which of the following is NOT a role in Scrum?**  
A. Product Owner  
B. Scrum Master  
C. **Team Lead**D. Developers

**14. Who facilitates Scrum events as needed?**  
A. Stakeholders  
B. Product Owner  
C. **Scrum Master**  
D. Team Lead

**15. The Developers are responsible for:**  
A. Managing stakeholders  
B. **Creating the Sprint Backlog**  
C. Ensuring Scrum is followed  
D. Prioritizing the Product Backlog

**16. What is a user story?**  
A. A detailed technical specification  
B. A bug report  
C. **A requirement written from the user's perspective**  
D. A test case

**17. Velocity is used to measure:**  
A. Number of bugs fixed  
B. Speed of the Scrum Master  
C. **Work completed by the team per sprint**  
D. Number of meetings held

**18. Which Agile methodology uses Kanban boards?**  
A. Scrum  
B. XP  
C. FDD  
**D. Kanban**

**Create a To-Do List in Jira**

Creating a TODO list in **Jira** using **Scrum development methodology** involves setting up a **Scrum board**, creating **user stories (or tasks)**, and organizing your backlog.

**1. Create a Scrum Project in Jira**

1. Log into your **Jira** instance.
2. Click on **"Projects"** > **"Create Project"**.
3. Choose **Scrum** (under "Agile") as the project type.
4. Select **"Team-managed"** or **"Company-managed"**, depending on your team setup.
5. Name your project (e.g., Scrum TODO List) and create it.

**2. Set Up Your Scrum Board**

When you create a Scrum project, Jira automatically generates a **Scrum board**. You’ll see:

* **Backlog** tab: for grooming and prioritizing tasks.
* **Active Sprint**: where current sprint tasks are shown in columns (TODO, In Progress, Done).

**3. Create TODO Tasks (Issues)**

1. Go to your **Backlog**.
2. Click **“Create issue”** at the bottom.
3. Choose issue type: typically **Story** or **Task**.
4. Enter:
   * **Summary** (what needs to be done).
   * **Description** (optional: details or acceptance criteria).
   * Assign to sprint (if sprint already started).
   * Set **priority** or **assignee** as needed.
5. Click **Create**.
6. Repeat for all TODO items.

**4. Organize Your Backlog**

* Drag and drop issues to prioritize.
* Group related items using **epics** (e.g., Epic: "User Login", Task: "Add login form").
  + Epic: “ToDo List” Task: Create a ToDo List
  + Epic 2: “Save ToDo List” Task: Save the exisiting ToDo lIst
* Add **labels**, **components**, or **estimates (Story Points)** if needed.

**5. Start a Sprint**

1. In the **Backlog** view, click **“Create Sprint”**.
2. Drag tasks from the backlog into the sprint.
3. Click **“Start Sprint”**.
4. Set:
   * **Sprint name**
   * **Start and end dates**
   * **Goal** (optional but good practice)

**6. Manage the TODO List in the Active Sprint**

In the **Active Sprint** (your Scrum board):

* Tasks will appear in **columns**: To Do, In Progress, Done.
* Move tasks across columns as work progresses.