Software Engineering Lab Task 3

10-12-2024

Eshwar Deshmukh Chavan

HU22CSEN0100999

**Introduction to JIRA Software:**

* JIRA is a proprietary product developed by Atlassian.
* Used for bug tracking, issue tracking, and agile project management.
* Popular among software development teams for planning, tracking, and releasing software projects.
* Provides a centralized platform for managing tasks and organizing work.
* Integrates with other software development tools and offers customizable features.

**Uses of JIRA Software**

1. **Agile Project Management**:
   * Supports Scrum (sprints) and Kanban (continuous flow) methodologies.
   * Helps in creating user stories, managing sprints, and tracking progress.
2. **Bug and Issue Tracking**:
   * Logs, prioritizes, and resolves software bugs effectively.
   * Allows assigning tasks with clear deadlines to team members.
3. **Task Management**:
   * Breaks large projects into manageable tasks.
   * Monitors status through workflows.
4. **Software Development**:
   * Tracks code changes and manages version control.
   * Integrates with CI/CD tools for streamlined development.
5. **Tracking and Reporting**:
   * Uses dashboards and charts to monitor project performance.
6. **Team Collaboration**:
   * Facilitates seamless communication between teams.
   * Centralizes project data for alignment and transparency.

**Key Features of JIRA**

* Issue Tracking and Custom Workflows
* Reporting and Analytics
* Integrations with other tools
* Automation capabilities
* Task prioritization and user management

**JIRA Scheme:**

* A JIRA scheme is a collection of values used in one or more JIRA projects.
* Schemes that apply to the entire JIRA project or all issue types within the project include:
  1. Issue Type Scheme
  2. Notification Scheme
  3. Permission Scheme
  4. Issue Security Scheme

**System Administration:**

JIRA offers several useful features for administrators and users:

* Audit Log:
  + View details about issues created and changes made.
  + Helps in tracking the history and modifications of issues.
* Issue Linking:
  + Shows if issues are linked to other existing issues.
  + Allows deactivating issue linking from the panel.
* Mail in JIRA:
  + Send issues to an account on a POP or IMAP mail server.
  + Messages can be written to the file system generated by an external mail service.
* Events:
  + Describe the status, default template, notification scheme, and workflow transition post-function associations.
  + Classified into System events (JIRA defined) and Custom events (User defined).
* Watch List:
  + Allows users to watch specific issues and receive notifications about updates.
  + Click "watch" in the issue window to start watching an issue.
  + Click on the number in brackets to see who is watching your issues.

**JIRA Issues and Issue Types:**

* Issue: A unit of work that needs to be tracked and managed.
* Issue Types: Categorize and manage work items effectively.
  1. Bug: Error, flaw, or problem in the software.
  2. Improvement: Enhancements or modifications to improve an existing feature.
  3. Task: Specific action or piece of work that needs to be completed.
  4. Epic: Large, high-level body of work broken down into smaller components.
  5. Story: User-focused requirement or feature.
  6. Incident: Disruptions, failures, or unexpected problems in services.
  7. Sub-Task: Smaller work item created under a larger issue.

**JIRA Issue Attributes:**

* Attributes describe, organize, and track work efficiently.
  + Summary: Concise title describing the issue.
  + Description: Detailed explanation of the issue.
  + Status: Current progress of the issue (e.g., Open, In Progress, Resolved, Closed).
  + Assignee: Team member assigned to work on the issue.
  + Reporter: Person who identified or reported the issue.
  + Priority: Importance or urgency of the issue (High, Medium, Low).
  + Labels: Tags or keywords for grouping and filtering issues.
  + Components: Sections or sub-parts of a larger project.
  + Fix Version: Software version where the issue will be resolved.

**JIRA Software Statuses:**

* Statuses track the progress of issues and tasks.
  1. Open: New issue, not yet assigned.
  2. In Progress: Work has begun on the issue.
  3. Resolved: Issue has been fixed but needs verification.
  4. Closed: Issue is completed and finalized.
  5. Reopened: Issue reopened after being previously resolved.
  6. Pending: Issue waiting for input from another team member.
  7. On Hold: Issue temporarily paused.
  8. Deferred: Issue postponed for later.
  9. Done: Issue completed and ready for closure.

**JIRA Components:**

* Components are core features of JIRA Software, enabled by default.
* Components page displays a list of components and details for each component.
* Includes Name, Description, Lead, Default assignee, Associated issues, etc.
* Add, edit, or delete components on the project's Components page.

**Sub-Tasks:**

* Split a parent issue into smaller, manageable tasks.
* Components:
  + Sub-Task Issues: Break down issues into smaller tasks for comprehensive addressing.
  + Creation Methods:
    - Create a sub-task under a parent issue.
    - Convert an issue into a sub-task.

**Workflows:**

* Define the lifecycle of an issue through a set of statuses and transitions.
* Components:
  + Statuses: Represent the positions of issues within a workflow (e.g., Open Issue, Resolved Issue, In Progress Issue, Reopened Issue, Close Issue).
  + Transitions: Bridges between statuses, showing how an issue moves from one status to another.
  + Assignee: The person responsible for the issue.
  + Resolution: Explains why an issue transitions from open to closed.
  + Conditions: Control who can perform a transition.
  + Validators: Ensure the transition can happen given the state of the issue.
  + Properties: Recognized by JIRA on transitions.
* Usage:
  + Assign the status of the issue from the window.
  + The workflow maps the progress of the project and provides a quick overview of the work in progress.

**Plug-ins in JIRA:**

* Enhance JIRA's functionality with additional features.
* Examples:
  + Zendesk, Salesforce, GitHub, Gitbucket.
  + Enable support teams to report issues directly into JIRA.
  + Create unlimited private repositories with full-featured issue and test management support.

**JIRA Agile:**

* Supports Agile methodologies like Scrum and Kanban.
* Components:
  + Agile Board: Shows the progress of sprints and issues.
  + Backlog: Plan sprints and estimate stories.
  + Workflow: Map columns to the statuses of your workflow.

**Clone and Link in JIRA:**

* Cloning Issues:
  + Allows different teams to work separately on the same issue.
  + Helps resolve issues quickly.
* Issue Linking:
  + Create associations between two existing issues on the same or different JIRA servers.
  + Helps in tracking related issues and their progress.

**Reports in JIRA:**

* Track progress and provide insights into project performance.
* Types of Reports:
  + Burndown Chart: Shows the actual and estimated amount of work to be done in the sprint.
  + Sprint Report, Epic Report, Version Report, Velocity Chart, Control Chart, Cumulative Flow Diagram.
  + Use different chart options to represent project progress.