

Secure Software Design and Engineering Lab Manual



**Jazia.sajid Lab Engineer FCSE
Ghulam Ishaq Khan Institute of Engineering and Technology, Pakistan**

Lab 01: Security in the SDLC

Objectives

- Integrate security into SDLC phases and document requirements in Jira.
- Map security considerations to SDLC phases using case studies.

SDLC

Security is a critical aspect of software development, and it should be integrated into every phase of the Software Development Lifecycle (SDLC). This lab introduces secure software development practices and teaches students how to document and address security considerations in a real-world scenario using Jira.

Introduction to JIRA

JIRA is a project management and bug-tracking tool that allows software developers to plan, track, and work faster. JIRA is the main source of information for future software releases. Developers can plan new features to be added and bugs to be fixed in the next release.

JIRA is an Incident Management Tool used for Project Management, Bug Tracking, Issue Tracking and Workflow. JIRA is based on the following three concepts – Project, Issue and Workflow.

Important Points to Note

The following points explain some interesting details of JIRA.

- JIRA is an incident management tool.
- JIRA is developed by Atlassian Inc., an Australian Company.
- JIRA is a platform independent tool; it can be used with any OS.
- JIRA is multi-lingual tool – English, French, German, Japanese, Spanish, etc.
- JIRA supports MySQL, Oracle, PostgreSQL and SQL server in the backend.
- JIRA can be integrated with many other tools – Subversion, GIT, Clearcase, Team Foundation Software, Mercury, Concurrent Version System and many more.

License and Free Trial

The following points describes the legalities of the JIRA Tool.

- JIRA is a commercial tool and available as a Trial version for a limited time.
- To utilize JIRA services, a license is required.
- JIRA provides free license for academic projects.
- A 15-day trial version is available for an individual person to use.

Use of JIRA

Following are some of the most significant uses of JIRA.

- JIRA is used in Bugs, Issues and Change Request Tracking.
- JIRA can be used in Helpdesk, Support and Customer Services to create tickets and track the resolution and status of the created tickets.
- JIRA is useful in Project Management, Task Tracking and Requirement Management.
- JIRA is very useful in Workflow and Process management.

Role of JIRA

A role is a set of permissions granted to JIRA users or groups to view or modify data within Assets. Roles can have three scopes: access all of Assets (JIRA admin) for a single project schema (Object Schema Manager, Developer, or User) and all object types within that schema.

First Understand SCRUM and Agile Frameworks

Agile and Scrum are frameworks used in project management, particularly in software development, to enhance collaboration, flexibility, and efficiency.

- **Agile Approach:** Encourages adaptive planning, iterative development, early delivery, and continual improvement, promoting flexibility and quick responses to change.
- **Scrum:** One of the most popular Agile frameworks designed to break complex projects into smaller, manageable pieces called sprints (typically 2–4 weeks long). Scrum teams focus on delivering incremental parts of the product at the end of each sprint.

JIRA Interface: Kanban and SCRUM

Kanban

- Kanban is a visual framework used to manage tasks and workflows. It originated from lean manufacturing but has become popular in software development and other industries to reduce inefficiencies and eliminate bottlenecks.

- In JIRA, Kanban projects (tasks, features, bugs, etc.) are represented on a Kanban board, which uses visualizations like columns that represent different stages of the workflow (e.g., "To Do," "In Progress," "Done").

SCRUM

Definition:

Scrum is an Agile framework that focuses on delivering work in fixed, time-boxed iterations called sprints, typically lasting 2–4 weeks.

Key Elements:

- **Sprints:** Short, consistent periods of time in which the team completes a set of work items.
- **Roles:**
 - Product Owner: Manages the backlog.
 - Scrum Master: Ensures the team follows Scrum practices.
 - Development Team: Completes the work.
- **Events:**
 - Regular meetings like sprint planning, daily stand-ups (15-minute daily check-ins), sprint reviews, and sprint retrospectives.
- **Artifacts:**
 - **Product Backlog:** A prioritized list of all tasks and features for the project.
 - **Sprint Backlog:** A list of tasks that the team commits to completing during a sprint.
 - **Increment:** The final product or deliverable completed at the end of the sprint.

Process:

Work is divided into sprints, and the team focuses on completing a specific set of tasks during each sprint. Progress is reviewed, and adjustments are made after each sprint in the retrospective.

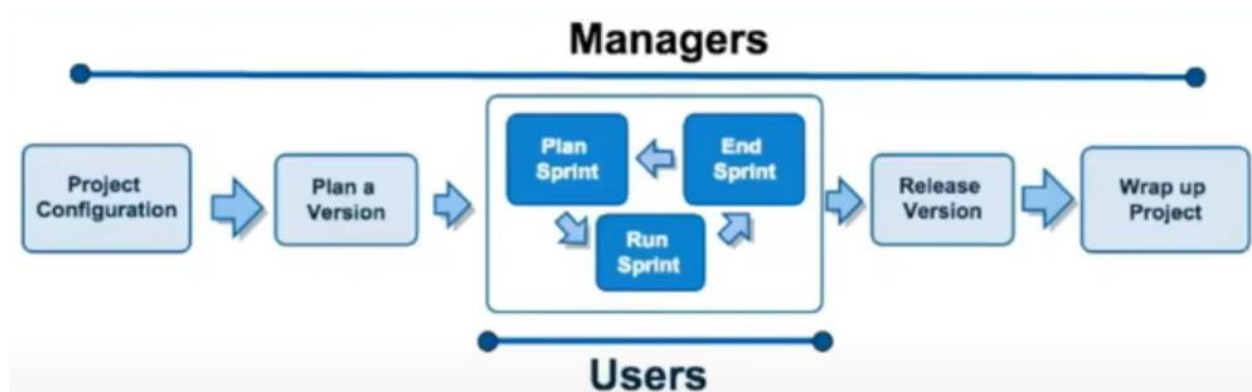
Teams with clear project goals and a product that can benefit from incremental development and frequent reviews.

S. No.	Core Features	Description
1	Boards	<p>JIRA supports Scrum and Kanban boards.</p> <p>These boards provide an immediate snapshot of the project to the team.</p> <p>Helps to quickly review the progress of the project and see the status of the individual tasks.</p> <p>Board workflow can be customized to fulfil the way a team wants to proceed.</p>
2	Business Project Template	<p>JIRA supports a number of business templates to manage simple tasks and complex tasks like workflow.</p> <p>Template can be customized based on the team and their approach. Ex: Workflow can be customized based on each team's approach.</p> <p>Every step is accounted and team can move to achieve their goals.</p>
3	Task Details	<p>Tasks can be defined at the individual level to track the progress.</p> <p>Status of every task, comment, attachment and due dates are stored in one place.</p>
4	Notifications	<p>An email can be sent for a particular task to the users.</p> <p>Voting and watching features to keep an eye on the progress for the stakeholders.</p> <p>Use @mention to get the attention of a specific team member at Comments/Description.</p> <p>User will instantly notify if something is assigned or if any feedback is required.</p>
5	Power Search	<p>JIRA supports a powerful search functionality with Basic, Quick and Advanced features.</p> <p>Use the search tool to find answers like due date, when a task was last updated, what items a team member still needs to finish.</p> <p>Project information at one place, search within a project.</p>

6	Reports	<p>JIRA supports more than a dozen reports to track progress over a specific timeframe, deadlines, individual's contribution, etc.</p> <p>Easy to understand and generate different reports those help to analyse how the team is going on.</p> <p>Easy to configure these reports and display the matrices to the stakeholders.</p>
7	Scale with Team Growth	JIRA supports any business team and any project irrespective of size and complexity.
8	Add -Ins	<p>JIRA supports more than 100 add-ins to connect with different software to make work easy.</p> <p>Wide range of add-ins makes it as universal across the globe.</p>
9	Multilingual	JIRA supports more than 10 languages those are widely used as English (US, UK, India), French, German, Portuguese, Spanish, Korean, Japanese and Russian.
10	Mobile App	<p>JIRA is available as a Mobile Application as well.</p> <p>It is available on Google Play Store and App Store (iTunes) of Apple.</p> <p>Easy to stay connected with the team while moving anywhere with notification, comments and project activity.</p>

A Project contains issues; a JIRA project can be called as a collection of issues. A JIRA Project can be of several types. For example

- Software Development Project
- Marketing Project
- Migration to other platform project
- Help Desk Tracking Project
- Leave Request Management System
- Employee Performance System
- Website Enhancement



Tool Overview



Select a template to get started


Your choice won't limit what you can do in Jira



Project management
Manage & track agile work plus integrate developer tools like GitHub



Scrum
Plan, prioritize & schedule sprints using scrum framework



Business collaboration
Manage work with list & calendar view plus get project summaries

Continue

Create a New Project

Click on your projects then go to create project Then there is an option of Create Issue

Welcome!

Your first project is ready to kick off. It's where you'll track tasks across teams, turning big ideas into real outcomes.

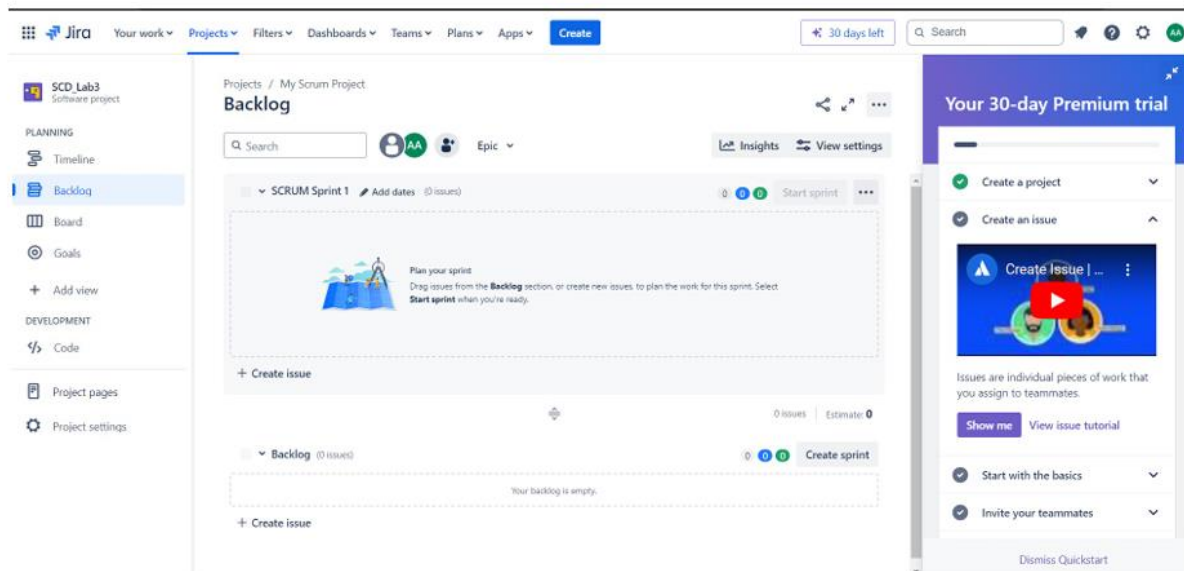
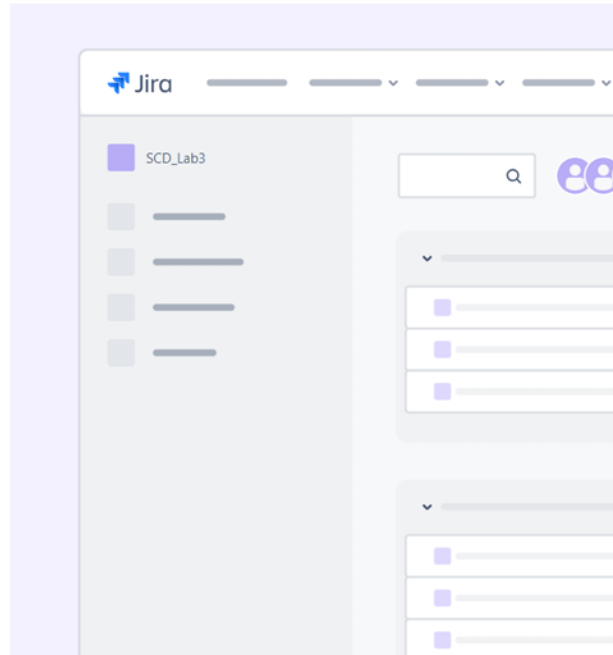
Name your project

SCD_Lab3

How familiar are you with Jira? *

- ☒ Not familiar
☐ Somewhat
☐ Familiar

Get started



When you click on create Issue There are further options

• Task • Bug • Story

Types of Issues in Java:

Task:

A task represents a unit of work that needs to be completed. For example, "Write documentation for the login module."

Bug:

A bug is a problem or defect in the software that needs to be fixed. For example, "Fix the login button not responding in mobile view."

Story (User Story):

A user story is a feature or functionality described from the end-user's perspective. For example,

"As a user, I want to be able to reset my password."

Epic:

An epic is a large body of work that can be broken down into smaller tasks or user stories. For example, "Implement user authentication system" could be an epic with multiple tasks or stories related to it.

Sub-task:

A sub-task is a smaller part of a larger task or story. It helps break down complex issues into manageable chunks. For example, "Design UI for the login page" could be a sub-task under a larger task of **"Develop login functionality."**

What is Assignee?

An assignee is the person responsible for working on or resolving a particular issue (task, bug, story, etc.) in Jira.

Create Story:

- Task
- Bug
- Epic

Bugs and tasks describe a single piece of work, while epics are used to describe group of issues relate to the same, larger body of work. Epics are typically completed over several sprints or a longer time frame if you don't use sprints

Click on Task:

The screenshot shows the 'Create' dialog in Jira. The 'Project' is set to 'SCD_Lab3 (SCRUM)' and the 'Issue type' is 'Story'. The 'Status' dropdown is open, showing 'To Do' as the selected option. A red border highlights the 'Summary' field, which is currently empty. A red error message 'Summary is required' is displayed below the field. The 'Create' button is visible at the bottom right.

Required fields are marked with an asterisk *

Project *
SCD_Lab3 (SCRUM)

Issue type *
Story

Learn about issue types

Status
To Do

This is the initial status upon creation

Summary *
Summary is required

☐ Create another

Cancel Create

- In Progress • To Do

The screenshot shows the 'Create' dialog in Jira. The 'Project' is set to 'SCD_Lab3 (SCRUM)' and the 'Issue type' is 'Story'. The 'Status' dropdown is open, showing 'In Progress' as the selected option. The 'Summary' field is empty. The 'Create' button is visible at the bottom right.

Required fields are marked with an asterisk *

Project *
SCD_Lab3 (SCRUM)

Issue type *
Story

Learn about issue types

Status
To Do
In Progress
Done

☐ Create another

Cancel Create

Dashboard

Create Dashboard

The screenshot shows the Jira interface for a project named 'SCD_Lab3'. The 'Dashboards' menu is open, and a modal dialog titled 'Create a dashboard to track the status of your projects.' is displayed. The dialog includes a 'More about dashboards' link, a 'View all dashboards' button, and a 'Create dashboard' button. The background shows the 'Backlog' view with a search bar, a 'Start sprint' button, and a 'Create issue' button. On the right, there is a 'Your 30-day Premium trial' sidebar with a list of tasks: 'Create a project', 'Create an issue', 'Start with the basics', and 'Invite your teammates'. The sidebar also includes a 'Dismiss Quickstart' button.

Projects / SCD_Lab3
Backlog

Create a dashboard to track the status of your projects.
[More about dashboards](#)

View all dashboards
Create dashboard

+ Create issue

0 issues | Estimate: 0

Backlog (0 issues) | Create sprint

Your backlog is empty.

+ Create issue

30 days left

Search

Insights | View settings

Start sprint

SCD_Lab3
Software project

PLANNING
Timeline
Backlog
Board
Goals
+ Add view

DEVELOPMENT
Code

Project pages
Project settings

Your 30-day Premium trial

- Create a project
- Create an issue
- Start with the basics
- Invite your teammates

Explore classes

Dismiss Quickstart

Create dashboard

Required fields are marked with an asterisk *

Name *

Description

Viewers

 Private ▼ Only you

Add

 Private

Editors

 Private ▼ Only you

Add

 Private

Save

Cancel

HOME | YOUR WORK | PROJECTS | FILTERS | **DASHBOARDS** | EDITORS | FIELDS | APPS | **EDIT**

Default_Dashboard




Add gadget

Change layout

Done



 You are currently editing your dashboard. Changes will be saved automatically.



Drag a gadget to this column or [add a new gadget](#)



Drag a gadget to this column or [add a new gadget](#)

Select Owner

- Project
- Group

Workflow:

- In Progress
- Resolved
- To Do
- Closed

Backlog:

The backlog of a SCRUM board shows the issues for your project grouped into a backlog and sprints. In the Scrum backlog, you can create and update issues, drag and drop issues to rank them or assign them to sprints.

Walkthrough Task

Steps to Integrate Security into SDLC

1. Planning Phase

- **Goal:** Identify security goals and requirements at the outset of the project.
- **Steps:**
 1. Define the scope of the project.
 2. Identify stakeholders and regulatory requirements (e.g., GDPR, HIPAA).
 3. Log into Jira and create a new project:

<https://confluence.atlassian.com/jira064/jira-user-s-guide-720416011.html>

(You can go through their official guide for more details)

- Navigate to Projects > Create Project.
 - Choose the appropriate project template (e.g., Scrum or Kanban).
 - Name the project and configure its settings.
4. Create an Epic titled Planning Phase:
 - Go to Backlog > Create Epic.
 - Enter a description of the planning phase objectives.

5. Add child issues under the Epic:

- Create tasks for listing security objectives and documenting compliance requirements.

2. Requirements Analysis Phase

- **Goal:** Document functional and non-functional security requirements.
- **Steps:**
 1. In Jira, create individual issues for each requirement:
 - Example: All user passwords must be hashed using SHA-256.
 - Example: Sensitive data must be encrypted in transit and at rest.
 2. Link these issues to the Planning Phase Epic:
 - Open the issue, go to More > Link Issue, and select the Epic.
 3. Attach supporting documents or diagrams:
 - Use the Attachments section in the issue screen.

3. Design Phase

- **Goal:** Develop a secure architecture and identify potential threats.
- **Steps:**
 1. Perform a threat modeling exercise and document findings.
 2. Create issues in Jira for design tasks:
 - Conduct threat modeling for the login module.
 - Design secure API endpoints.
 3. Link these issues to the Requirements Analysis Phase in Jira.
 4. Use Story Points or Priority fields to indicate the importance of each task.

4. Implementation Phase

- **Goal:** Write secure code and conduct code reviews.
- **Steps:**

1. Document coding tasks in Jira:
 - Example: Implement input validation for user registration.
 - Example: Remove hardcoded keys from the codebase.
2. Assign issues to developers and set due dates.
3. Conduct peer code reviews:
 - Add a Sub-task to the main issue for code review documentation.
 - Use comments to log findings and solutions.

5. Testing Phase

- **Goal:** Validate that the application meets security requirements.
- **Steps:**
 1. Create Jira issues for test cases:
 - Example: Test for SQL injection vulnerabilities.
 - Example: Validate session management.
 2. Attach test results to issues:
 - Use the Attachments section for screenshots or reports.
 3. Use the Status field to track progress (e.g., To Do > In Progress > Done).

6. Deployment and Maintenance Phase

- **Goal:** Ensure the application remains secure in production.
- **Steps:**
 1. Create Jira issues for deployment tasks:
 - Example: Set up logging and monitoring for API calls.
 - Example: Install and configure a Web Application Firewall (WAF).
 2. Plan recurring security audits:
 - Add tasks for quarterly vulnerability assessments.
 - Use Due Date fields to schedule tasks.
 3. Document maintenance efforts and lessons learned:
 - Use the Comments section to record insights.

Case Study: Secure Online Banking System

- **Scenario:** A banking application requires secure user authentication and data handling.
 - **Objective:** Map security considerations to SDLC phases based on the scenario.
 - **Tasks:**
 - Identify threats during the design phase (e.g., SQL injection).
 - Document solutions (e.g., input validation) in Jira.
-

Tasks for Students

Task 1: Document Requirements in Jira

- Log into Jira and create a new project.
- Add issues for functional and non-functional security requirements based on a provided scenario.
- Link issues to an Epic and attach supporting documentation.

Task 2: Perform Threat Modeling

- Identify three potential threats to a sample application.
- Create Jira issues for mitigating these threats.
- Use the Priority field to rank the tasks.

Task 3: Simulate Vulnerability and Document Fixes

- Simulate security vulnerability (e.g., weak password policy).
- Use Jira to document:
 - The vulnerability.
 - Its impact.
 - The solution and steps taken to fix it.
- Add screenshots or supporting files to the issue.

Case Study 2

Scenario Title: Bug Tracking and Issue Resolution in JIRA

Scenario Overview

In this scenario, you will play the role of a software development team member working on a project. Your team uses JIRA to track and manage tasks, bugs, and new feature requests. You will be assigned various tasks related to bug tracking and issue resolution using JIRA.

Scenario Steps

1. **Login to JIRA**
 - Access the JIRA platform using the provided login credentials.
2. **View the Project Dashboard**
 - After logging in, you will be taken to the project dashboard. Familiarize yourself with the layout, including the project summary, recent issues, and activity feed.
3. **Create a New Issue**
 - Click on the “Create” button and select “Issue” to create a new bug report.
 - Fill in the necessary details such as issue type (bug), summary, description, priority, and assignee.
4. **Assign an Issue**
 - Assign the issue you just created to a team member of your choice.
5. **Add Comments**
 - Navigate to the issue you created and add a comment providing additional details about the bug or issue. Mention any steps to reproduce it.
6. **Transition the Issue**
 - Move the issue through the workflow. Change its status from “Open” to “In Progress” to indicate that you’re actively working on it.
7. **Resolve the Issue**
 - Once you believe the bug is fixed, change the issue status to “Resolved.” Be sure to explain how you resolved the issue in the comment section.
8. **Create a Sub-Task**
 - Create a sub-task related to the issue you resolved. This could be a task for testing the fix.

