**AI Integration Plan**

This document outlines the integration of AI into the project workflow to automate test reviews, issue resolution, and test creation. The AI will work seamlessly with Jira, GitHub, and Discord to enhance productivity and efficiency.

**AI Responsibilities**

**1. Test Review and Bug Resolution**

* **Automated Test Review:**
  + Analyze results of all automated tests, including Postman API tests and UI tests.
  + Identify failed tests, their causes, and potential fixes.
* **Bug Resolution Workflow:**
  1. Create a Jira bug for failed tests.
  2. Notify the team in the Discord chat with details of the bug.
  3. If capable, attempt to fix the issue by:
     + Modifying the code.
     + Creating a push request on GitHub with the fix.
  4. If unable to fix, mark the Jira bug with a request for help and notify the team in Discord.

**2. Test Creation and Execution**

* **Trigger:**
  + Initiated when a Jira development ticket is moved to "Testing."
  + The Discord bot monitors the Jira board and triggers the AI.
* **Workflow:**
  1. Identify the scope of the task and design the necessary tests.
  2. Create corresponding Jira stories for the tests.
  3. Write the test scripts (unit tests, UI tests, API tests).
  4. Run the tests and update their statuses in Jira:
     + "In Progress" -> "In Testing" -> "Done."
  5. Submit push requests for the new test scripts to GitHub.
  6. Notify the team in Discord about test progress and completion.
  7. If tests fail, analyze the root cause and:
     + Attempt to fix the code or tests.
     + Communicate any unresolved issues in Discord.

**Steps to Implement AI Functionality**

**1. Integration with Tools**

* **Jira API:**
  + Automate ticket creation, updates, and transitions.
* **GitHub API:**
  + Automate code pushes, pull request creation, and status checks.
* **Discord Bot:**
  + Notify the team of AI actions and request assistance when needed.
* **Testing Tools:**
  + Postman for API tests.
  + Platform-specific tools (SikuliX, Espresso, XCTest) for UI tests.

**2. Core Functionalities**

* **Test Analysis:**
  + Build algorithms to parse test results and identify issues.
* **Code Fixing and Validation:**
  + Utilize AI models to generate and validate code fixes.
* **Test Generation:**
  + Leverage AI to design, write, and optimize test scripts based on requirements.
* **Issue Communication:**
  + Automate notifications and updates in Discord with detailed context.

**3. Automation and Monitoring**

* **Discord Bot:**
  + Monitor Jira ticket transitions to trigger AI workflows.
  + Communicate progress, results, and issues.
* **AI Trigger Events:**
  + Detect Jira task status changes and test results to initiate actions.

**Tools and Technologies**

* **Programming Languages:** Python for AI logic and API integrations.
* **APIs:**
  + Jira API for ticket management.
  + GitHub API for repository interactions.
  + Discord API for communication.
* **Libraries and Frameworks:**
  + AI libraries: OpenAI models for code analysis and generation.
  + Testing tools: Postman, SikuliX, Espresso, XCTest.
* **Infrastructure:**
  + CI/CD pipeline integration via GitHub Actions.
  + Cloud services for AI processing if needed (e.g., AWS, Azure).

**Benefits and Outcomes**

1. **Increased Automation:**
   * Streamline test reviews, bug resolution, and test creation.
2. **Improved Communication:**
   * Proactive updates to the team via Discord.
3. **Enhanced Productivity:**
   * Reduce manual effort in identifying and resolving issues.
4. **Continuous Improvement:**
   * AI learns from interactions and improves over time.