3/13/2025

Julia Kloda

Quality Assurance (QA)

Plan for Agile Software Development Project

Contents

[Quality Assurance (QA) 0](#_Toc198313455)

[Introduction & Objectives 2](#_Toc198313456)

[QA Roles & Responsibilities 2](#_Toc198313457)

[Quality Standards & Best Practices 3](#_Toc198313458)

[Agile Testing Approach 3](#_Toc198313459)

[Test Strategy 4](#_Toc198313460)

[Continuous Integration & Deployment (CI/CD) 4](#_Toc198313461)

[Defect Management & Tracking 4](#_Toc198313462)

[Acceptance Criteria & Definition of Done (DoD) 4](#_Toc198313463)

[Risk Management & Mitigation 5](#_Toc198313464)

[Continuous Improvement & Feedback Loops 5](#_Toc198313465)

[Compliance & Regulatory Requirements 5](#_Toc198313466)

[Tools & Technologies 5](#_Toc198313467)

[Conclusion 6](#_Toc198313468)

[References 7](#_Toc198313469)

# Introduction & Objectives

The Movie Recommendation System is a web application designed to suggest movies to users based on their preferences. By leveraging a pretrained model, the system allows users to describe what they are looking for in natural language. This enhances user experience by providing personalized suggestions.

The purpose of the QA plan is to ensure that the movie recommendation system meets performance, accuracy, and user experience standards by identifying and resolving bugs early in the project. To achieve this, we applied Agile testing methodologies. Our project scope includes unit testing, integration testing, functional testing, system performance testing, allowing for iterative enhancements and efficient issue resolution.

Scope of Quality Assurance Activities:

* Covers unit testing, integration testing, functional testing, and system performance testing.
* Focuses on scalability and optimized data loading by replacing CSV loading with JSON-based index and metadata storage.
* Implements a quality assurance pitfalls file to identify vulnerabilities and strengthen user data protection.

Key Quality Goals and Alignment with Agile Principles:

* Ensures performance, accuracy, and user experience standards are met.
* Detects and resolves bugs early through Agile testing methodologies.
* Supports continuous improvement and adaptability with iterative testing cycles.
* Enhances efficiency by optimizing data handling for faster server startup.

# QA Roles & Responsibilities

Me as the Project Owner define the product vision, prioritize the backlog, and ensure the team delivers value to users.

Martin, as the Scrum Master, facilitates Agile processes, removes roadblocks, and helps the team stay productive.

Both of us also contribute as developers and testers, writing and reviewing code. We strive to ensure software quality in accordance with best practices. we balance leadership, development, and quality assurance to drive the project’s success.

# Quality Standards & Best Practices

We use meaningful variable and function names to enhance code readability and maintainability, ensuring clarity in our logic. To maintain code consistency, we use Prettier for formatting, enforcing a uniform style across the project. To avoid hardcoding sensitive information by using environment variables or secrets management, and we follow secure authentication and authorization mechanisms to protect our applications. For performance optimization, we implement caching and optimize database queries to minimize load times.

We follow best practices in version control by writing meaningful commit messages, creating separate branches for each task, and using pull requests combined with code reviews. (Kloda, 2025a)

To ensure user protection, we hash or encrypt user IDs, implement API rate limiting, and secure access using OAuth authentication for safe interactions. (Kloda, 2025b)

For software reliability, we use Pytest to test our Flask server’s functionality, endpoints, and logic. For frontend testing, we use Jest to test our UI components and ensure they behave as expected. We follow recommended best practices during testing, such as using clear test names, keeping test logic simple, and focusing on a single behavior per test.

We are also planning to automate our tests using CI/CD to improve efficiency and maintain consistent quality.

# Agile Testing Approach

Agile methodology is an iterative approach to project management and software development that emphasizes flexibility, collaboration, and continuous improvement. We choose this approach in every meeting to adapt to changes effectively and ensure efficient progress.

We use Test-Driven Development (TDD) in our project, which is a software development approach that ensures code quality and maintainability by writing tests before implementing functionality. While we don’t always manage to follow TDD, we strive to apply it in most cases to catch bugs early and improve design. This approach helps us build a reliable and scalable. We also focus on Behavior-Driven Development and exploratory testing, which helps us to get feedback, improve our work and discover defects.

# Test Strategy

We focus on manual testing approaches. We use Jest and pytest for unit testing to ensure individual components function accurately. (Kloda, 2025c; Kloda, 2025d) We use Postman, which follows structured API testing methodologies that ensure comprehensive validation APIs. (Kloda, 2025e) Currently, we perform manual testing for exploratory testing. In future I’d like to use Selenium for user interface testing and emphasize usability testing.

# Continuous Integration & Deployment (CI/CD)

Version control plays a crucial role in Continuous Integration and Continuous Deployment (CI/CD) by enabling teams to collaborate efficiently and maintain a stable codebase. After considering various CI/CD options, such as Jenkins and Azure Pipelines, we decided to go with Bitbucket Pipelines as our team primarily uses Bitbucket for version control. We define workflows using a YAML configuration file (Kloda, 2025f). I think it is an excellent fit for our project because branch management is simple and well-integrated with our workflow. Additionally, pull request integration allows us to automatically trigger builds and tests when a pull request is created, ensuring that only quality code gets merged.

# Defect Management & Tracking

Our team manages defect tracking using an Agile Scrum board in Jira, ensuring transparency and efficient issue resolution. (Kloda, 2025g) Agile is a flexible project management approach that focuses on iterative development, collaboration, and continuous improvement. We prioritize tasks using the Fibonacci scale to balance workload management effectively. This helps estimate the effort needed for tasks, considering that bigger or more complex tasks take significantly more work than smaller ones. By following a clear Definition of Done, we ensure that defects are fully addressed, tested, and meet quality standards before being considered resolved.

# Acceptance Criteria & Definition of Done (DoD)

In our team, we establish acceptance criteria during sprint planning to ensure each user story has clear conditions for completion. We also follow a Definition of Done (DoD), which includes coding, testing, reviews, and deployment steps to maintain quality standards. I created brief file where I determined when MVP is done and what criteria measure this. (Kloda, 2024h)

# Risk Management & Mitigation

While building the project, we considered several potential risks related to libraries, tools, and data collection. When using libraries, we had to ensure they were up to date and be mindful of potential security vulnerabilities.  
Collecting user data, we had to consider the risks of data leakage, breaches, and SQL injection attacks. As a team, we created a QA file to ensure our project is secure. We use encryption for sensitive data, optimize our MySQL database with queries and indexing, and implement caching mechanisms. (Kloda, 2024i)

# Continuous Improvement & Feedback Loops

Our team follows an Agile methodology, which emphasizes continuous feedback and iterative improvement. Sprint retrospectives are a key part of our Agile process, allowing us to reflect on what went well and what need to be improved. During retrospectives, we discuss challenges and define actionable steps to enhance future sprints. This ensures we continuously learn how to optimize workflows and progress. (Kloda, 2025j)

# Compliance & Regulatory Requirements

Our project complies with the General Data Protection Regulation (GDPR) to ensure the legal and ethical handling of data. We prioritize the secure management of user data and preferences. We use caching and encrypted data storage to enhance performance while ensuring user information remains secure.

# Tools & Technologies

We use Bitbucket for version control, which helps team collaborate and track changes. CI/CD pipelines to automate the process of building, testing, and deployment. Postman is like a remote control for testing, it validates APIs efficiently, ensuring smooth integration between services. We also manage our tasks, sprints, and defect tracking in Jira, keeping our Agile workflow organized and transparent. We use MySQL, a database management system, to store and manage data for our applications, ensuring that information is organized and easily accessible. We also use LLaMA, a machine learning model, to help with tasks like natural language processing, enabling our system to better understand and interact with text data. For testing, we use Jest for JavaScript code to ensure everything works correctly and Pytest to catch bugs and verify our functions run as expected.

# Conclusion

I evaluated project deliverables against quality requirements to ensure that our project meets performance and user experience standards by applying Agile methodologies, continuous testing, and secure development practices.

# References

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/Assets/PRs.png [Accessed 6 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/Assets/DB\_hashing.png [Accessed 6 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/movie\_recommendation\_system/blob/main/server/tests/test\_server.py [Accessed 6 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/movie\_recommendation\_system/blob/main/client/src/tests/SemanticSearchBar.test.js [Accessed 6 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/Assets/Postman.png [Accessed 6 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/movie\_recommendation\_system/blob/main/bitbucket-pipelines.yml [Accessed 6 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/Assets/JiraTickets1.png [Accessed 6 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/Assets/DoDJira.png [Accessed 6 May 2025].

Kloda, J. (2024). *Quality\_assurance\_Pitfalls.pdf*. [online] Bitbucket. Available at: https://bitbucket.org/JuliaKloda/movie\_recommendation\_system/src/main/team\_documents/Quality\_assurance\_Pitfalls.pdf [Accessed 21 Mar. 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/Mural.pdf [Accessed 6 May 2025].