



SPRINT: An Assistant for Issue Report Management

Based on Research by Ahmed Adnan , Antu Saha & Oscar Chaparro

07 December, 2025

Problem Statement



Manual issue management is time-consuming and error-prone.



Projects receive hundreds of issues across components.



Tasks like duplicate detection, severity labeling, and bug localization require high effort.



Existing tools usually support only one task.

Proposed Solution

SPRINT

- An integrated, open-source GitHub application
- Automates major issue management tasks
- Uses state-of-the-art deep learning models
- Provides real-time feedback as GitHub comments and labels

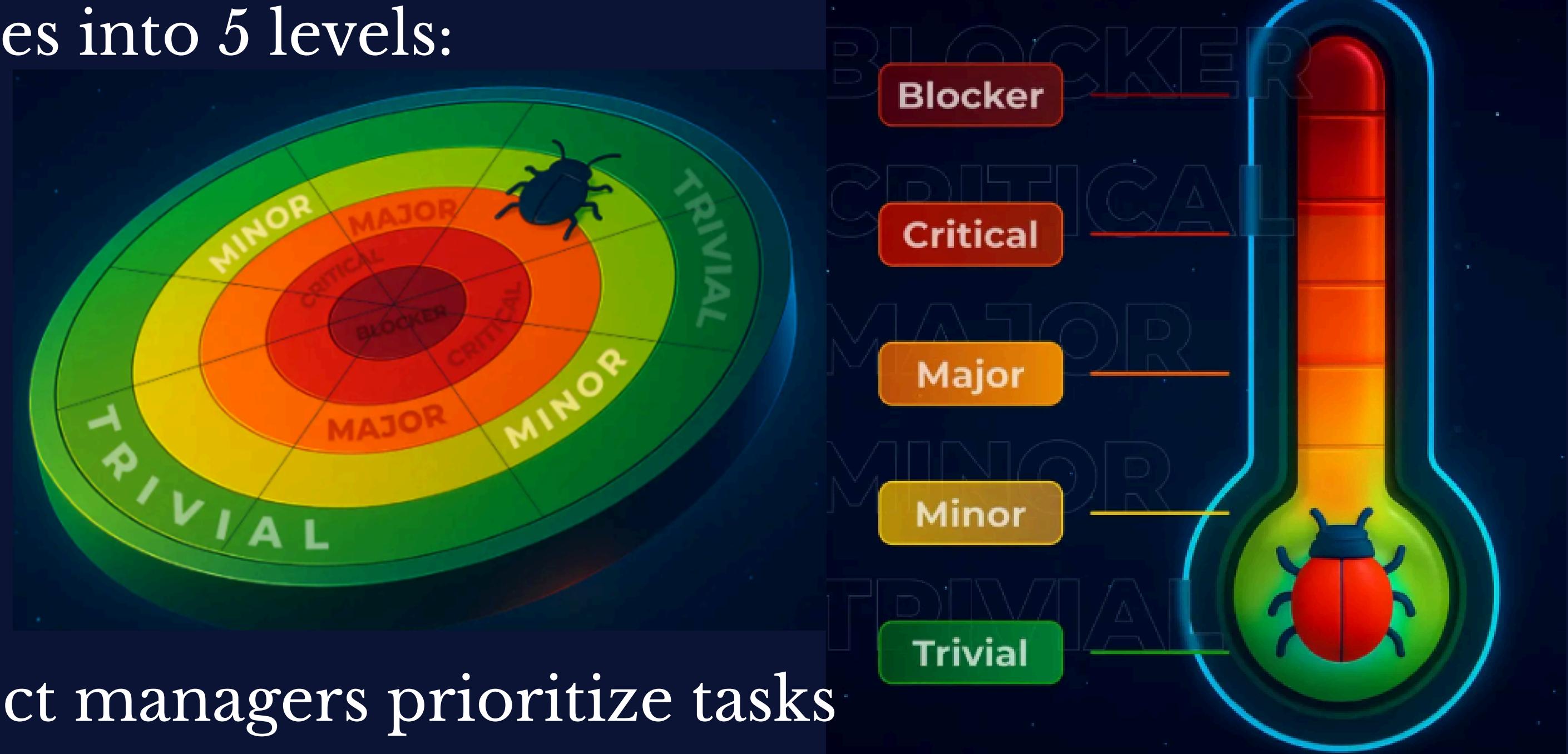
Core Features of SPRINT

- Issue Severity Prediction
- Similar Issue Identification
- Buggy Code Localization
- Fully automated and real-time
- Independent execution of all features



Features 1- Issue Severity Prediction

Classifies issues into 5 levels:



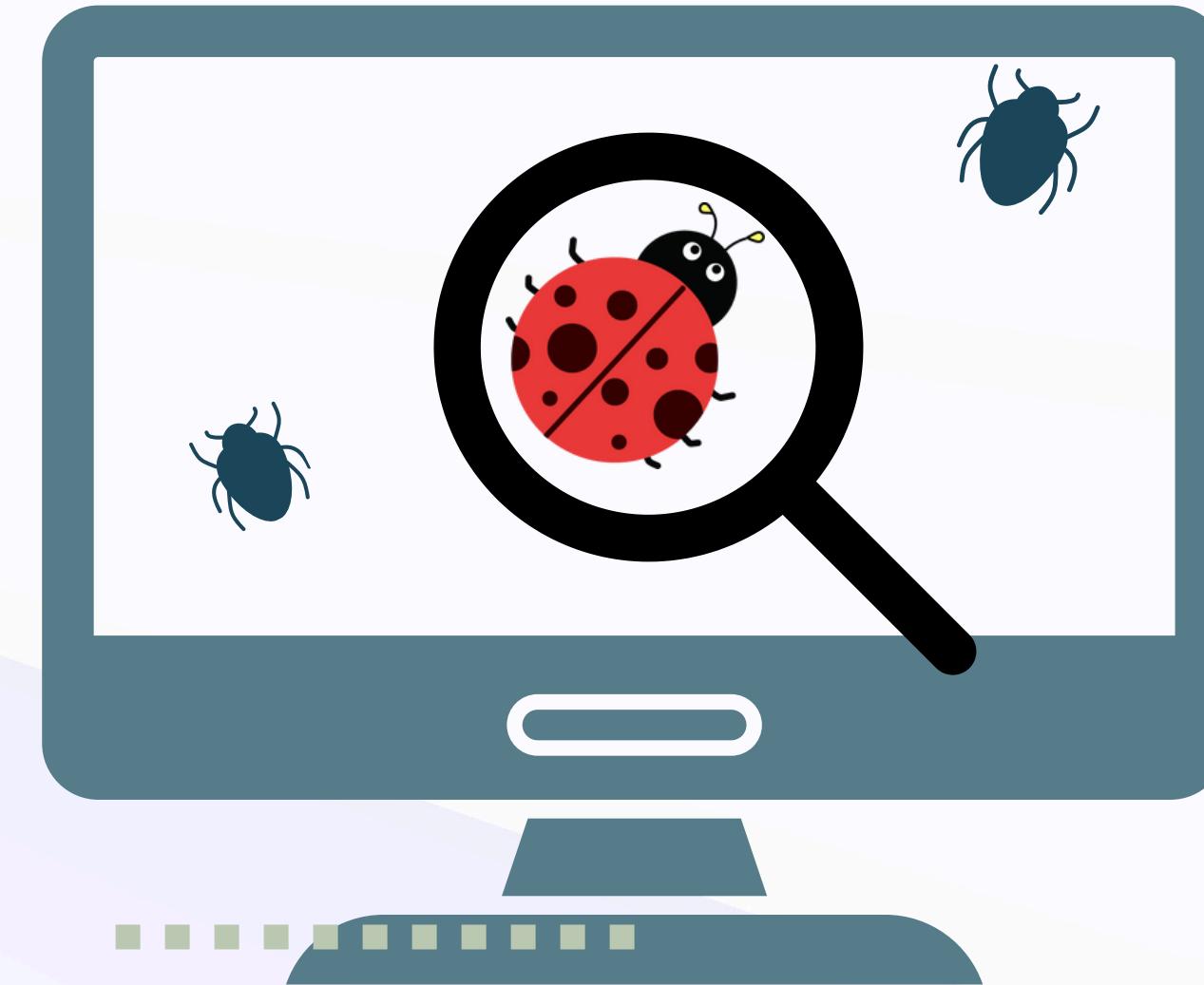
- Helps project managers prioritize tasks
- Implemented using RTA deep learning model

Feature 2- Similar Issue Identification

- ✓ Detects duplicate or related issues using RTA model
- ✓ Uses textual similarity between reports
- ✓ Adds a “Duplicate” label automatically
- ✓ Displays -
 - Issue ID
 - Title
 - Direct URL
- ✓ Reduces redundant issue handling

Feature 3- Buggy Code Localization

- Suggests potential source code files
- Implemented using fine-tuned LLaMA-2 model
- Ranks files by likelihood of being buggy
- Uses issue text + file paths
- Assists developers during debugging



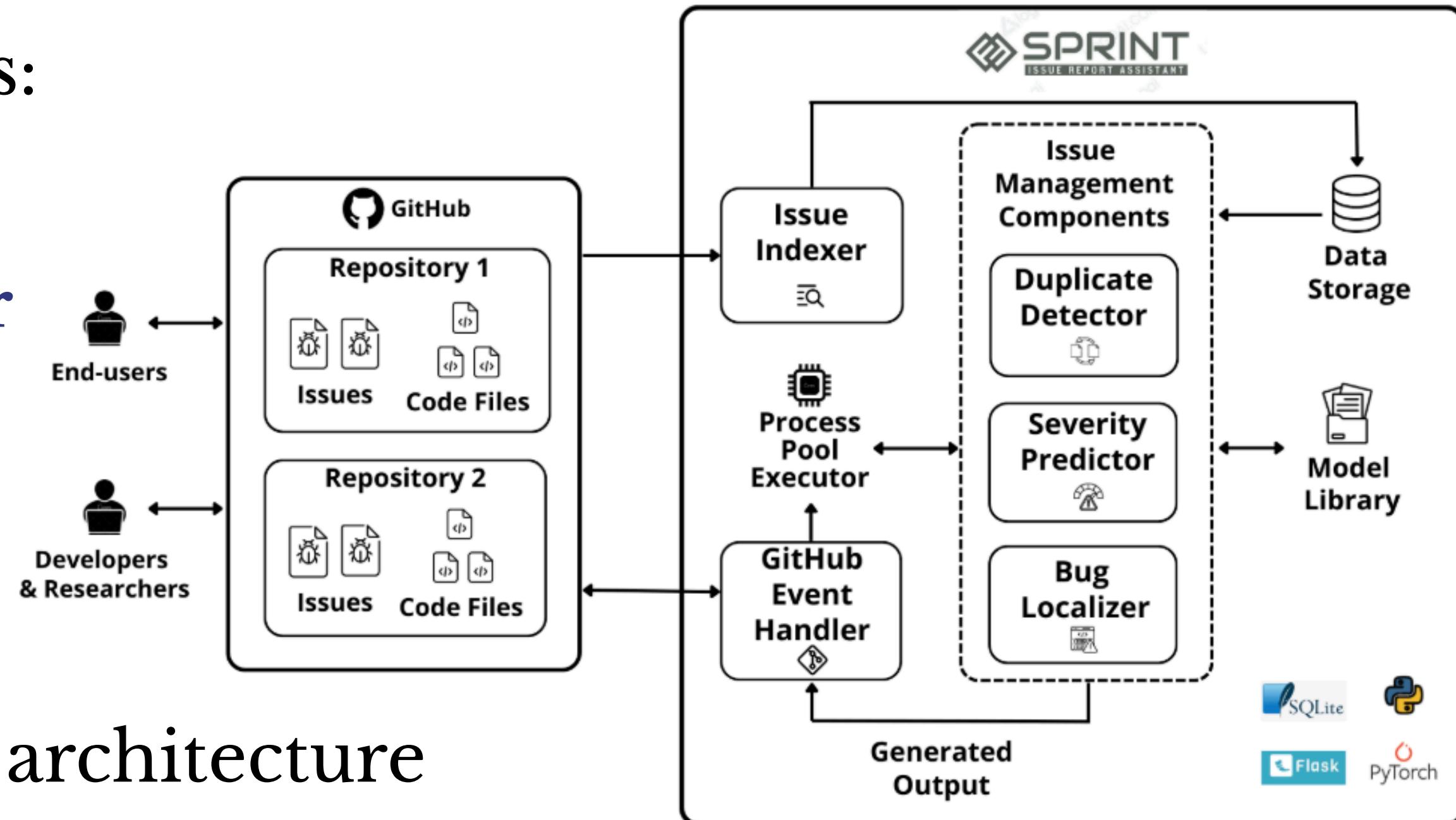
Tool Architecture

Three Main Components:

- Issue Indexer
- GitHub Event Handler
- Issue Management Components

Modular & plugin-based architecture

Supports extensibility and scalability



Architecture Overview

Issue Indexer

- Fetches all existing issues using GitHub Webhooks
- Stores in local SQLite database
- Uses page-based indexing
- Keeps database synchronized with GitHub 24/7

GitHub Event Handler

- Listens for new issue events
- Fetches new issue and source code
- Sends data to analysis components
- Posts back:
 - Labels
 - Comments
 - File recommendations

Issue Management Components

- Similar Issue Detection → RTA model
- Issue Severity Prediction → RTA model
- Bug Localization → LLaMA-2 fine-tuned model
- Uses multiprocessing for performance

How Validation Was Done ?

Validation

Two types of validation:

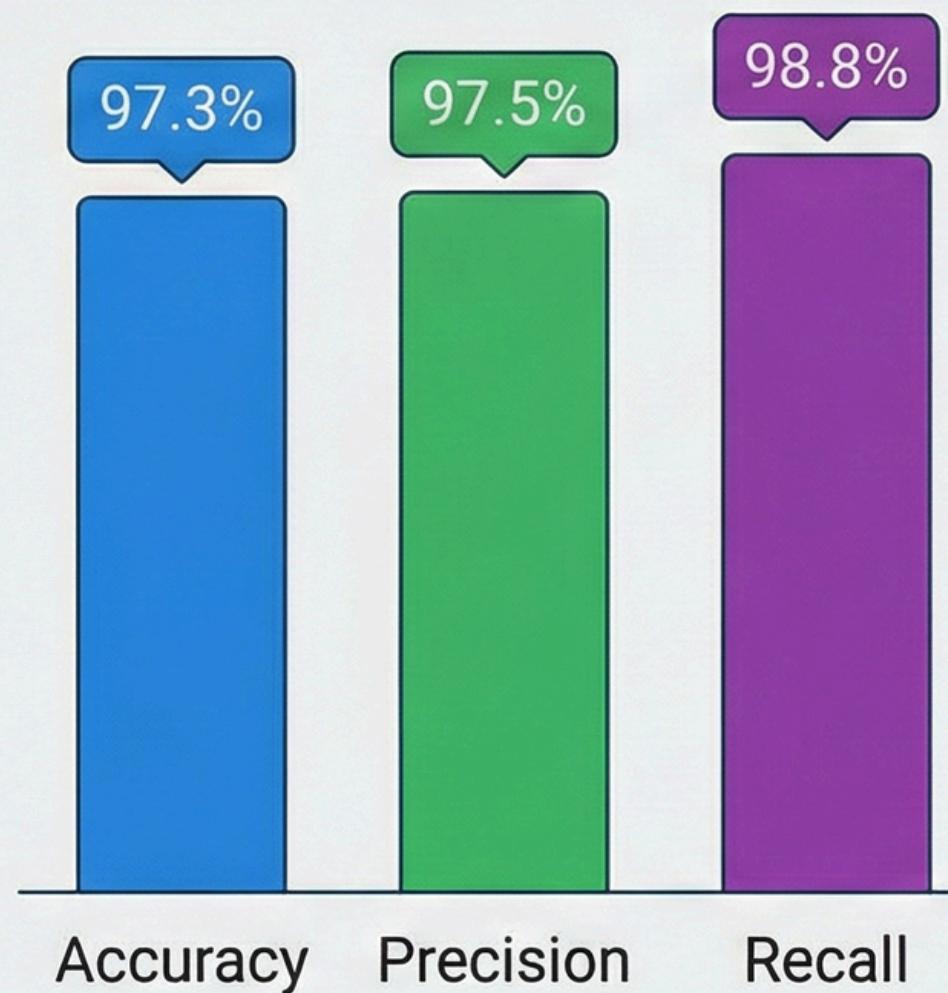
1. Offline Model Evaluation

2. User-Based Empirical Validation

- Used publicly available benchmark datasets from prior studies
- Maintained same train-test splits as original models
- Compared predicted outputs with ground truth labels
- Used standard IR and ML evaluation metrics
- Replicated original research methodologies

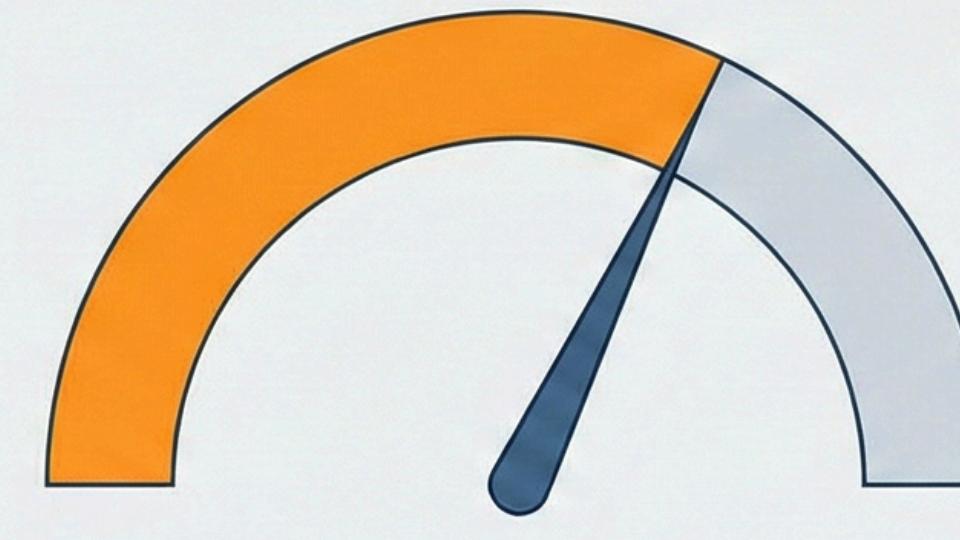
Model Evaluation Results

Duplicate Detection Metrics



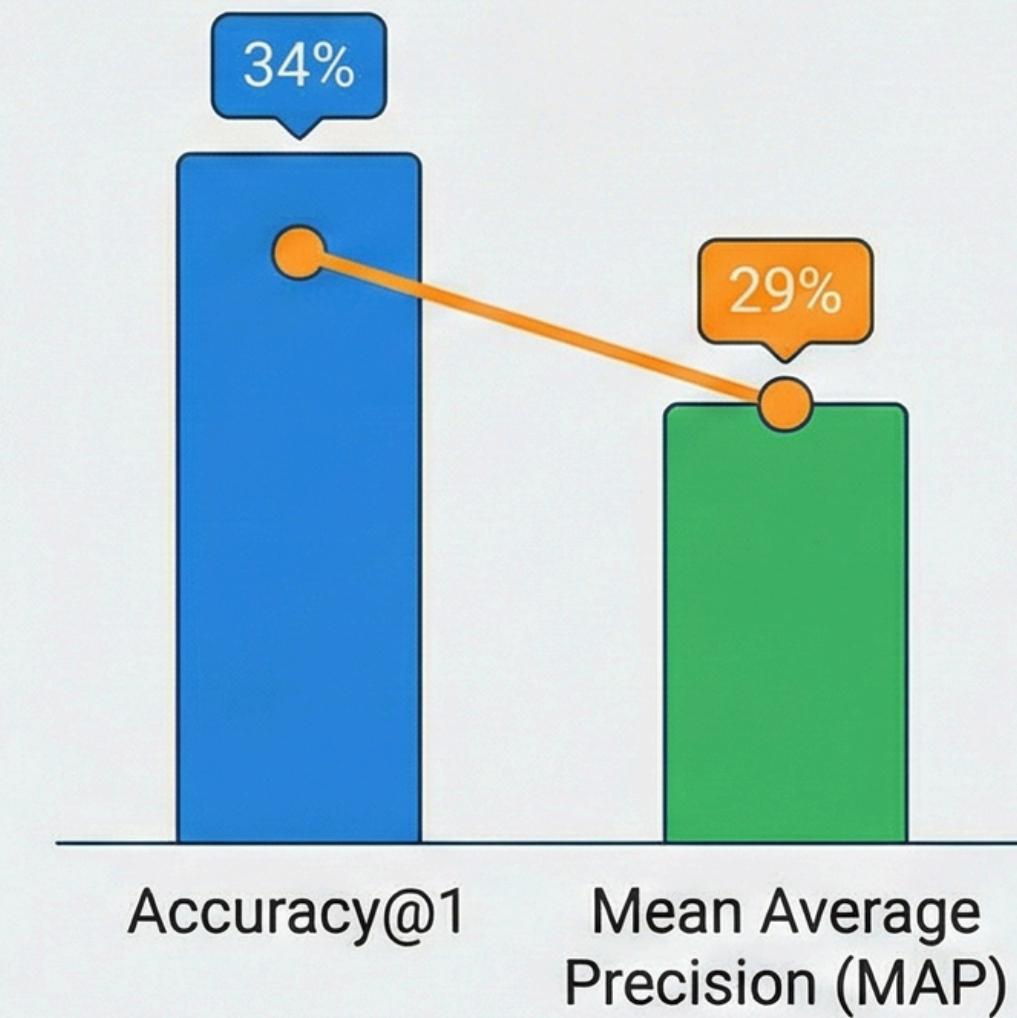
Extremely reliable
duplicate identification

Severity Prediction Accuracy



Accuracy: 65.6%

Bug Localization Metrics



Shows ranking is useful
but still imperfect

Misclassifications mostly
occurred between adjacent
classes (Major vs Minor).

User Study Results

Participants: 5 Professional Developers

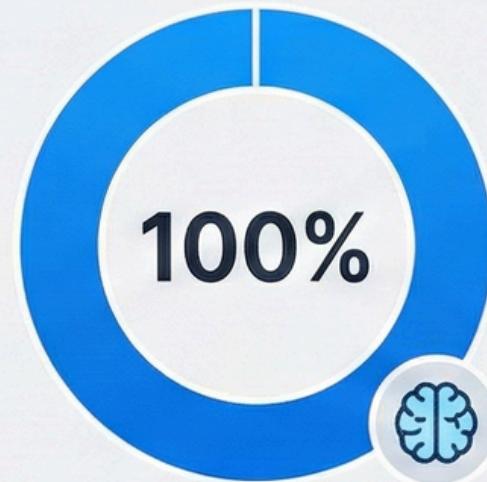
Experience: 3–8 years

USER STUDY RESULTS – USABILITY



4

Found SPRINT Easy to Use



Suggestions Easy
to Understand



Average Response Time



Rated as Practical & Helpful



USER STUDY RESULTS – ACCURACY



Duplicate Detection



Severity Prediction



Bug Localization



Users appreciated FAST RESPONSES
despite imperfect predictions.

Novelty of SPRINT

Related Work

Tools for duplicate detection: [Probot](#), [Find Duplicates](#), [NextBug](#)

Tools for prioritization: [Jira Priority Scheduler](#)

Bug localization tools: [BugLocalizer](#)

- ✓ SPRINT Combines three major issue management tasks in one tool
- ✓ Uses state-of-the-art deep learning models
- ✓ Fully open-source
- ✓ GitHub-native integration
- ✓ Plugin-based extensible architecture

Limitation

- Bug localization accuracy is still relatively low
- Only supports GitHub
- Cannot analyze historical versions of code, Only latest version
- Requires high computational resources for LLM inference
- Performance depends heavily on issue text quality
- Needs stronger cloud infrastructure for scaling

Future Design Improvement

Design Patterns

- Singleton Pattern → Central database and configuration manager
- Observer Pattern → For GitHub event handling
- Factory Pattern → For AI model loading
- Strategy Pattern → For switching ML models

Refactoring

- Improve bug localization using newer LLMs
- Add CI/CD integration
- Multi-repository support
- Intelligent developer recommendation system
- Support platforms beyond GitHub
(such as GitLab, Bitbucket)



Thank You