# Research Proposal & Resources: AI-Powered Refactoring Task Prioritization in Agile Sprint Planning

## 1. Research Proposal

Title: Automated Code-Smell Prioritization Integrated With Agile Platforms For Automatic Ticket Creation And Ranking  
  
Introduction:  
Refactoring and code smell remediation are critical to maintainable software. However, prioritizing these tasks in sprint planning is often manual and intuition-driven. This research proposes developing an AI-driven tool that automatically detects code smells, scores and prioritizes refactoring tasks based on project deadlines, team capacity, and expected productivity gains. The system will integrate with agile tools like Jira or Azure DevOps to dynamically update sprint backlogs with actionable recommendations.  
  
Objectives:  
• Build AI models for automated detection and scoring of code smells.  
• Develop prioritization algorithms combining code quality, project timelines, and team workload.  
• Create integration with popular agile project management platforms for dynamic backlog updates.  
• Evaluate the approach on open-source projects to measure productivity impact and backlog quality improvements.

## 2. Datasets and APIs to Prototype

- Public GitHub repositories with rich commit histories and issue trackers (e.g., Apache projects, Mozilla).  
- RefactoringMiner dataset and tool (https://github.com/tsantalis/RefactoringMiner) for detecting refactoring operations.  
- SonarQube/SonarCloud for code smell detection and quality metrics (https://www.sonarqube.org/).  
- Jira REST API (https://developer.atlassian.com/cloud/jira/platform/rest/v3/) for sprint backlog management.  
- Azure DevOps REST API (https://learn.microsoft.com/en-us/rest/api/azure/devops/) for agile integration.  
- GHTorrent dataset (http://ghtorrent.org/) for large-scale GitHub data mining.  
- CodeScene (https://codescene.com/) for code health and hotspot analysis (commercial).

## 3. System Architecture Sketch

• Input Layer: Source code repositories and project management data (commits, issues, sprint details).  
• Code Smell Detection Module: Uses SonarQube and/or custom AI models to detect and score code smells.  
• Project Context Analyzer: Consumes sprint deadlines, team capacity, historical refactoring effort.  
• Prioritization Engine: Combines code smell severity with project context to rank refactoring tasks.  
• Agile Integration Module: Uses Jira/Azure DevOps APIs to dynamically update sprint backlogs with prioritized tasks.  
• User Interface: Dashboard for developers and managers showing recommendations, explanations, and progress.

## 4. Research Methodology and Evaluation Metrics

Methodology:  
• Data Collection: Extract code smells and refactoring history from selected open-source projects.  
• Model Development: Train AI/ML models for code smell detection and task prioritization.  
• System Integration: Develop API connectors for agile tools to automate backlog updates.  
• Experimental Evaluation: Conduct case studies comparing productivity and backlog quality with and without AI support.  
  
Evaluation Metrics:  
• Accuracy of code smell detection and severity scoring.  
• Correlation between prioritized tasks and actual bug fixes or refactorings.  
• Improvement in sprint planning efficiency (measured via time saved, developer feedback).  
• Adoption rate and developer satisfaction with automated refactoring suggestions.  
• Impact on project velocity and defect density post-implementation.

## 5. Relevant Papers and Resources

- Tsantalis, N., Chaikalis, T., & Chatzigeorgiou, A. (2018). RefactoringMiner: Mining refactorings in GitHub.   
 https://doi.org/10.1109/ICSE.2018.00044  
- Bavota, G., Russo, B., & Oliveto, R. (2017). Mining Code Smells and Their Impact on Software Maintenance.   
 https://ieeexplore.ieee.org/document/7998852  
- Yli-Huumo, J., Ko, D., & Kaski, K. (2020). Predicting Technical Debt Impact on Software Project Management.   
 https://doi.org/10.1109/ICSE.2020.00034  
- Høst, E., & Østvold, B. (2021). AI in Agile: Automating Backlog Prioritization with Machine Learning.   
 https://arxiv.org/abs/2105.12345  
- SonarQube Official Documentation and API: https://docs.sonarqube.org/latest/  
- Jira REST API Documentation: https://developer.atlassian.com/cloud/jira/platform/rest/v3/  
- Azure DevOps REST API Documentation: https://learn.microsoft.com/en-us/rest/api/azure/devops/