TaskMate Final Report

Team TADS

Tanner Nantabutr
Virginia Tech
tannern@vt.edu

Asif Chowdhury
Virginia Tech
asifc@vt.edu

David Nguyen
Virginia Tech
dxvid@vt.edu

Sahaj Singh
Virginia Tech
sahaj@vt.edu

ABSTRACT

Task management within software development environments often lacks integration, leading to inefficiencies and increased cognitive load on developers. TaskMate proposes a VSCode extension that centralizes task management by compiling all TODO comments into a central, navigable list, integrating with tools like JIRA for enhanced workflow continuity. This solution aims to streamline development processes, reduce errors, and improve productivity.

INTRODUCTION

Today, managing tasks within a software development environment proves to be challenging and tedious. Fragmented and scattered code is common, often leaving developers to write TODO and FIXME comments across project files and code bases. A non-coherent and convoluted workflow can cause workplace problems such as bad and inefficient code. To address this, a developer productivity tool is needed. Our solution is to develop an extension for IDEs (assimilated with other tools as well) that compiles and organizes task-related comments so that developers can easily track and layout tasks to measure progress and efficiency. After receiving feedback on our proposal, we understood that our product had to be easily navigable and intuitive. Furthermore, the application must be responsive and quick—anything else may lead to inefficiency and frustration for users.

RELATED WORK

Task management within integrated development environments (IDEs) has seen various implementations

aimed at minimizing the disruption of developers' workflow. The Todo Tree extension for VSCode, for instance, offers a straightforward method of managing TODO comments directly within the IDE, representing them in a tree view to simplify navigation and prioritization [3]. Despite its utility, Todo Tree primarily focuses on individual task tracking and lacks comprehensive project management features found in more robust tools.

In contrast, external platforms like Asana and Trello [4][5] provide extensive task management functionalities,

including detailed task categorization, progress tracking, and integration with other business tools, but they require developers to switch between applications, leading to a potential loss of concentration and

productivity. The need for a more seamlessly integrated solution within IDEs like VSCode is evident, as such integration can significantly reduce cognitive load and maintain developers in their coding 'zone,' thereby enhancing overall productivity.

DESIGN

1.1 High-Level and Low-Level Design

TaskMate uses the Builder Design Pattern to modularize the process of extracting, processing, and displaying TODO comments. The design is divided into three main components. First, the extraction component parses open files to identify any TODO or FIXME comments. Next, the processing component organizes these comments based on priority, tags, or the developer they are assigned to. Finally, the display component compiles the organized comments into a centralized and navigable list within the VSCode interface.

At a lower level, the system is implemented in a modular way, as shown in the pseudocode below. The TodoExtractor class demonstrates how the program scans files for TODO comments and stores their location and content:

```
class TodoExtractor:
    def __init__(self):
        self.todos = []

    def extract(self, file):
        for line_num, line in enumerate(open(file), 1):
            if "TODO" in line:
                self.todos.append(f"{file}:{line_num} -
{line.strip()}")

    def build_list(self):
        return "\n".join(self.todos)
```

This modular approach ensures that each task—parsing, organizing, and displaying—is handled independently, making the code easy to maintain and extend.
Key Features of TaskMate include a centralized TODO list, real-time updates, integration with Jira for task management, filtering and prioritization of tasks, and seamless navigation to specific TODO comment locations in the code [2]. These features streamline workflow for developers and allow them to focus on their tasks without the need for external tools or manual searches.

1.2 Testing Approach

To ensure reliability, we developed a Black-Box Test Plan that includes 10 unique test cases to validate core functionality. These tests focus on accurately compiling TODO comments, sorting them by priority, and handling edge cases such as empty files, duplicate comments, and varying comment formats. For example, one test case verifies that the system compiles all TODO comments in a given file. The input for this test is a file containing five TODO comments, and the expected output is a list displaying all five comments. This approach ensures TaskMate meets its functional requirements and performs effectively under different conditions.

DEPLOYMENT PLAN

To deploy TaskMate, the first step is to package the extension for the VSCode Marketplace, ensuring it is easily accessible for developers. Alongside the release, we will provide detailed documentation that explains the installation process and usage instructions so users can quickly get started. For the Jira integration feature, users will be able to set up connections using API keys, enabling seamless synchronization between their TODO comments and Jira tasks.

Maintenance of TaskMate will be driven by user feedback to ensure the tool continuously meets the needs of developers. Regular updates will be released to address bugs, improve existing features, and incorporate new suggestions. To track and resolve any issues efficiently, we will utilize an issue-tracking system for logging bugs and feature requests. As the user base grows, we also plan to scale TaskMate by extending support to other popular IDEs like IntelliJ and Eclipse, ensuring broader compatibility and usability for developers across different environments.

CONCLUSION

TaskMate addresses the inefficiencies of managing scattered TODO comments by providing a centralized, navigable solution within VSCode. By integrating features like filtering, prioritization, and Jira linking, TaskMate reduces cognitive load and streamlines developer workflows. Looking ahead, future iterations of TaskMate will focus on extending its compatibility to other IDEs broadening its useability. Additional enhancements, such as customizable workflows, advanced analytics and further integrations with project management and CI/CD tools are also something we are looking to do. Through ongoing maintenance, scalability, and feature expansion, TaskMate positions itself not just as a tool but as a fundamental part of the future of software development.

REFERENCES

1. Gruntfuggly, *"Todo Tree - Visual Studio Marketplace,"* Visual Studio Marketplace. Available: https://marketplace.visualstudio.com/items?itemName=Gruntfuggly.todo-tree.
2. Atlassian, *"Jira Software Documentation,"* Atlassian. Available: https://atlassian.com/software/jira.
3. Microsoft, *"Visual Studio Code API Documentation,"* Visual Studio Code Docs. Available: https://code.visualstudio.com/api.
4. Asana, *"Asana Task Management Platform,"* Asana. Available: https://asana.com.
5. Trello, *"Trello Project Management Tool,"* Trello. Available: https://trello.com.