

**Software Engineering**

**CSC-40066**

**School of Computing and Mathematics**

**Keele University**

**UK, 2024**

Date of submission:17th May, 2024

Module leader: Sorriel Harriet

Team 2 Software Engineering Evaluation of Practice Report

​​

|  |  |
| --- | --- |
| Student Name | Student ID |
| Jack Moroney | 20015176 |
| Garima Gupta | 23026715 |
| Elorm Blavo | 23034614 |
| Furqan Ahmed | 23043052 |
| Jyothesh Karnam | 23032448 |

**Evaluation of Practice**

**Introduction**

The purpose of the evaluation of practice is to assess a piece of software created by another team and determine aspects that may cause issues or find areas of the software that could be considered successful in deployment against the problem they are attempting to solve. Beyond this, the evaluation will also demonstrate an understanding of modern software engineering practices and design principles.

Effectively evaluating software is important in providing constructive feedback, and so critically assessing software with valid and reviewed practices can benefit not only the client but also the software engineers. This is all done to produce the highest quality product, allowing it to be applied effectively to the given target audience effectively, whilst also producing something that could be developed upon given that the user has the potential to comment on their experience.

Evaluation could cover coding practice such as code quality, efficiency in terms of its usage and delivery, collaboration in the open-source market and so producing software that is informative and explains its process through comments and variable naming, and allowing future development by using code with high/preexisting compatibility; or front-end development features in the UX/UI design such as, how effective it solves the problems experiences by their chosen target audience, how consistent the software functions and behaves for the user to eliminate learning curves, visual hierarchy and how it is used in familiarising the user with difference levels features and their importance/category, user control and how they can interact with the software (whether it be mouse, keyboard, body tracking, voice control, etc.), and finally accessibility for impaired users who may have problems accessing typical software and so require assistance with visual/hearing impairments or limitations with limb usage.

**Evaluation Methodology**

We assessed Team 3 based on their presentation showcasing the development of the My Time Planner App, which includes features like an integrated Alarm system and Google navigation functionality. The app's core feature allows users to create tasks by inputting specific details and descriptions to prioritise their commitments. Additionally, they successfully integrated Google Maps to provide navigation routes and precise locations.

Throughout the demonstration, we noted several aspects that contributed to their structured approach, including their thorough review process, comprehensive analysis of software design and development, and the functionality of the software.

**Review of the demonstration**

Assessing the presentation, all three team members provided a thorough review of the project's scope and objectives achieved thus far. They effectively communicated the purpose, key features, and practical utility of the application, ensuring clarity and relevance for the audience's needs.

They demonstrated a comprehensive understanding of the entire software, including their insights into the end goal. Their discussion of meeting user stories was precise, accompanied by accurate technical details and information crucial for making future changes and improvements.

**Evaluation of User Stories and Features**

The team presented four user stories that cater to professionals, students, individuals with amnesia, and passengers, ensuring a real-time scenario for each user type. Users can organise tasks by inputting task details, setting reminders, and receiving alerts on their devices for upcoming tasks.

They successfully integrated GPS functionality to access the user's current location, enhancing route planning accuracy through the integration of the Google Maps API. The user stories established by the team depict realistic and practical scenarios relevant to the target audience. The comparison of these user stories with the actual features demonstrates a precise match, ensuring the app's functionality aligns with user expectations.

**Agile Methodology Implementation**

**Employing Agile Framework**

The team opted for the scrum framework, drawing from their successful experiences in collaboration with previous web application courses and incorporating lean principles.

This encompassed sprint planning, standups, review sessions, and active team collaboration primarily through channels like the team's dedicated chat and WhatsApp for seamless communication.

**Design and Development**

Delving into the team's approach to iterative development, they divided tasks into smaller, manageable units, aligning with lean principles to streamline project creation by focusing solely on core features and avoiding unnecessary functionalities.

**Code Review**

During code walkthroughs, the team showcased implemented code functionality and utilised GitHub repositories for version control, change tracking, and collaboration, adhering to coding best practices and maintaining code quality standards.

When assessing the code, the team emphasised meaningful user feedback upon encountering errors, efficient error handling during task management, functionality testing, and thorough documentation examination for clarity and completeness.

**Feedback from Team 2**

Team 2's feedback proved instrumental in identifying areas for improvement and potential enhancements overlooked by the development team. Notably, they lauded the well-structured reminder function with clear parameters and error handling but suggested improvements in implementing robust security measures, such as encryption, access controls, data validation, and proactive threat monitoring for an enhanced user experience. We also recommended transitioning to automated testing for efficiency instead of manual testing procedures.

**Evaluation Findings**

**Knowledge and Process**

**Adherence to Software Engineering Practices**

The team executed agile approaches, including Scrum in an efficient way. They had daily stand-ups. Sprints, and iteration on a regular basis. Because of iterative `development team were able to incorporate feedback early and adapt quickly to changes. Using these tools, like Git for version control, supported collaborative development and facilitated seamless code integration and tracking of routes.

The team applied the lean principle to streamline their workflow to focus on waste elimination and process optimisation. This included setting priorities for critical work and eliminating unnecessary processes, helping to maintain high productivity and delivering valuable features quickly.

In addition to Teams and WhatsApp, the team also used university email to communicate. Team meetings were held regularly basis for review progress find the issues and make sure everyone was on the same page about the project’s objective.

**Software Design Principles**

The inspiration for “My Time Planner” came from user stories. For example, busy professionals need to keep track of weekly tasks impacted by the inclusion of features such as task recording and reminders. Similarly, a student’s need for flexibility and updates resulted in dynamic task management capabilities.

The user stories were matched with important features including task input, notifications, and location integration with Google Maps, illustrating a user-centric approach to software design. The application’s features are carefully designed to meet the unique requirements specified in the user stories.

The modular nature of the application makes it easier to maintain and update components. This can be seen in the way the functionalities are organised within the code. For example, the code separates tasks, alarms, location assistance, etc. Not only does this make maintenance easier, but it also makes it easier to scale up and add new functionality in the future.

The selection of technologies and the architecture of the application allow for scalability. The use of backend development in Python, with the intention of integrating Next.js and MongoDB in the future, allows for scalability and future expansion of the application. The choice of technologies ensures that the application can support growing workloads and user requirements without sacrificing performance.

**Code Review Process**

The team delivered working software, and their demonstration highlighted the core functionalities, including task management and location services which tied into the user scenario and minimum viable product for a use case. The team used Python for backend development, showcasing the ability to add, update, and delete tasks effectively. The use of Python's extensive libraries and frameworks facilitated the rapid development of robust features. The integration of the Python module, geopy for geocoding and calculating distances is a key feature. The determine\_distance\_to\_taskLocation method attempts to geocode the task location and calculate the distance to the user's current location. Error handling for geocoding timeouts is implemented but could be improved for robustness.

One notable area for improvement was the lack of automated testing. The manual testing process was time-consuming and prone to errors, indicating a need for better testing practices. Implementing automated tests would significantly enhance the reliability and maintainability of the code.

**Analysis of demonstration**

**Tools and Techniques**

The choice of Python for backend development, along with plans to use Next.js and MongoDB, was appropriate for the application’s requirements. These technologies support scalability and future enhancements. Python’s versatility and extensive library support were leveraged effectively for task management and location-based features.

The task management system is robust enough to allow users to create, update, and delete tasks. The integration of alarms ensures that users receive timely reminders, enhancing the app's utility. The system is designed to handle various scenarios and user needs, making it a comprehensive tool for time management. These functionalities ultimately meet the user acceptance criteria and grant its user the functionality needed to manage the time of a busy professional.

The application’s ability to monitor and update the user’s location in real-time, though not fully functional yet, is a significant feature that adds value by providing accurate navigation and travel time estimates. This feature can further extend the use of the app making it useful not only for busy professionals but also for commuters and individuals who frequently travel to different locations for their tasks.

**Team Processes**

The team adopted the Scrum framework due to previous experience using it. The framework helped manage tasks and ensure regular progress checks, though the team encountered difficulties with overall implementation. This highlighted the importance of reliable team collaboration and the need for effective strategies to ensure full engagement from all team members.

The iterative development process allowed the team to adapt quickly to changes and continuously improve the application, though more rigorous testing could have enhanced the final product. Implementing automated tests and a continuous integration/continuous deployment (CI/CD) pipeline would streamline the development process and improve code quality.

**Critical Evaluation**

While the project demonstrated solid initial development, the lack of automated testing was a significant drawback. Implementing automated tests could have improved the efficiency of the development process and the reliability of the code. Automated tests would ensure that each feature works as intended and reduce the likelihood of bugs being introduced during development.

The team’s ability to overcome challenges, such as integrating Google Maps API, demonstrated resilience and adaptability. These qualities are essential for successful software development and long-term perseverance in the field of software engineering. However, the team could benefit from more thorough planning and time management to ensure that all planned features are implemented and tested within the project timeline.

The commit history shows that all team members participated in the software-building process, even though not all members of the team were at the same level of coding expertise. The collaborative nature of this work means team members learned from more experienced ones fostering a spread of knowledge and ideas. The team would thus be able to continue with further development.

**Areas for Improvement**

**Automated Testing**: Implementing automated tests for each method would ensure that changes to the code do not introduce new bugs. Unit tests for individual methods, integration tests for the entire workflow, and end-to-end tests for user interactions would be beneficial.  
**Error Handling**: Improving error messages to provide more context and guidance for the user would enhance the user experience. For example, specifying the reason for a geocoding failure or providing suggestions for correcting input errors.  
**Feature Enhancements:** Completing the implementation of features such as real-time location updates, advanced location integration with Google Maps, and user authentication would significantly improve the application's functionality and user satisfaction.  
**Code Refactoring**: Refactoring the code to separate concerns more clearly, such as extracting location services and task management into separate classes or modules, would improve maintainability and scalability.

**Presentation and Academic Skills**

**Academic Skill:** This demonstrates a thorough understanding of software engineering principles and practices. The analysis is supported by relevant skills beneficial in their academic gaining proficiency in Python programming language and software development tools enhances technical skills, which can be applied to both academic research and practical projects.  
**Presentation Skills**: The team’s presentation of the "My Time Planner" application was clear and well-structured. They effectively communicated the features and benefits of the application, as well as the challenges they faced during development. Visual aids and live demonstrations were used effectively to illustrate key points and enhance understanding.

**Conclusion**   
   
In conclusion, the assessment of the "My Time Planner" application presentation indicates its capability to meet the needs of its target users efficiently. The team follows a collaborative methodology that encourages review and assessment of each team member's performance and contributions. They demonstrate a solid understanding of agile and lean methodologies, leading to a well-organized development process.

However, to enhance the application further moving forward, the team could benefit from implementing a more robust automated testing framework to ensure the reliability and stability of the application. By investing in automated testing tools and procedures, they can streamline the testing process, identify bugs early, maintain a high level of code quality throughout development iterations and foster stronger teamwork and collaboration within the team.