Automated Task Management

Project Proposal

Ethan Werner
Computer Science
Virginia Tech
Blacksburg, VA, US
ethanw21@vt.edu

Jake Frohlich
Computer Science
Virginia Tech
Blacksburg, VA, US
jakefrohlich00@vt.edu

Neha Bangari
Computer Science
Virginia Tech
Blacksburg, VA, US
nehabangari@vt.edu

Vrishank Bangari
Computer Science
Virginia Tech
Blacksburg, VA, US
vrishank@vt.edu

ABSTRACT

In today's fast-paced software development environments, effective task management is essential for maintaining productivity and meeting project deadlines. Software engineers often deal with organizing, prioritizing, and distributing tasks manually, which can lead to inefficiencies and uneven workload distribution. Additionally, unexpected interruptions can disrupt workflow and can lead to delays within teams. Our solution addresses the problem of manual task allocation by introducing an automated task scheduler designed specifically for software engineering teams and individuals. The proposed solution uses algorithms to delegate tasks automatically throughout the week, considering factors such as task priority, individual skill sets, and current workloads. By automating the delegation process, the scheduler aims to optimize resource utilization and increase team productivity. Additionally, the product will reduce bias within teams in terms of deciding task importance. This system will be designed with the intent of fostering a more balanced and efficient work environment for software engineers. Ultimately, this will increase motivation, reduce burnout, and help software engineers prioritize their actual work.

INTRODUCTION

In the modern world of software engineering, managing many tasks and projects simultaneously is often subject to increased complexity for the engineer. Software engineers are often faced with the difficulty of manually prioritizing these tasks, allocating resources and balancing their own workload to finish deadlines on time and efficiently. This can lead to stress and decreased productivity. Additionally, these manual approaches not only consume valuable time for the engineer but can also result in inefficacious task distribution. With the dynamic nature of the general scope of software engineering–in which timelines, tasks and requirements can change at any given moment based on investor needs–there would naturally need to be a tool in place to maintain efficient task management without constant manual adjustments to the system.

The Task Scheduler we propose seeks to address this issue by automating the entire process of task delegation from the user's perspective. The solution we propose leverages AI to consider how long a given engineer should take to complete tasks, which tasks should come first and the deadlines associated with each task. With this system, the software engineer can adapt to shifting projects, demands or interruptions instantly through our tool, increasing the overall productivity and throughput of the engineer. In another perspective, with our product, a software engineer can allocate more focus to the development and problem-solving related work than spending time figuring out the delegation of the work at hand.

RELATED WORK

There are some existing tools that software developers use to address similar concerns. Trello is a task management tool that uses boards, checklists, and cards to help teams organize their work [1]. Trello allows users to manually enter tasks and assign due dates and team members to them. There are multiple features to organize Trello boards, and the platform also offers templates so users don't have to create a board from scratch. Another similar task management tool is Monday.com, which offers the ability to create custom dashboards to view various features such as team progress and hours, budgets, and task assignments [2]. Both these tools allow users to manually create visual representations of task distribution for a given project, and offer some support through templates and existing designs. However, these applications do not have the ability to intelligently and automatically distribute tasks across a team.

In a research study detailed in “A Study on Task Management System,” authors Jyothi and Parkavi explain a proposed technique to define tasks using a matrix to aid in project management. The authors described the use of a task matrix based on an Eisenhower matrix to delegate tasks across a project management team [3]. In this study, the authors describe this concept as a 2x2 matrix that organizes tasks by importance and urgency, and also lists features of project deliverables. This matrix model is useful because it clearly organizes tasks and descriptions, making it easier for teams to evaluate task importance. The authors also go over a few different prioritization models in order of complexity. The first one is a HI/LO model. This model evaluates tasks based on their impact and complexity. Another similar model is the CARVER model, which uses six factors to evaluate tasks: criticality, accessibility, return, vulnerability, effect, and recognizability. Each of these factors are assigned a score on a scale of 1 to 5 to prioritize tasks. Finally, the Carpenter model uses weighted comparisons between criteria in order to rank tasks. This model typically relies on automated calculations and visual results. The paper also lists some existing task management software, including Trello, Todoist, and Taskworld.

SOFTWARE PROCESSES

The software engineering process we will be using is Agile, more specifically extreme programming. We chose this because the product we aim to create is heavily focused on user experience. The purpose of this software will be to make the lives of software engineers easier so we will want to receive constant feedback on each iteration of our plan and product so we can perfect it. This is the main reason why we chose to use Agile. Another reason is that Agile allows for high quality products at high speed and we have a relatively short time frame to complete this project. Agile will let us truly maximize our time. Additionally, it will allow us to gather feedback after each iteration in order to make changes and better the product. It will also let us be more open with communication, allowing us to convey our opinions and thoughts more frequently.

We chose to hone in on extreme programming because of its ability to support changing customer needs. This process will allow us to go through shorter development cycles, which can be useful in a group as we can get continuous feedback. Through this process, we will present every iteration to software engineers and get feedback for improvement on the next iteration. We will also hold regular scrum meetings to ensure all team members are in the loop on all work being done throughout the development process. As our product is user-centric, we hope to get feedback from our potential user group as well.

REFERENCES

[1] Trello. "Trello: Organize Anything, Together." Accessed September 27, 2024. https://trello.com/?campaign=18406634139&adgroup=143241824842&targetid=kwd-3609071522&matchtype=e&network=g&device=c&device_model=&creative=633330905204&keyword=trello&placement=&target=&ds_eid=700000001557344&ds_e1=GOOGLE&gad_source=1&gclid=Cj0KCQjwr9m3BhDHARIsANut04ZSVTouMpWjT1TBmoJ31O1RyRv7J1WUsBJfMTqgQUcF57qGTwh_FU4aAo6YEALw_wcB.

[2] Monday.com. "Dashboards-Monday.com Features." Accessed September 27, 2024. https://monday.com/features/dashboards.

[3] Jyothi, N.S., & Parkavi, A. 2016. A Study on Task Management System. *2016 International Conference on Research Advances in Integrated Navigation Systems (RAINS),* 1-6. DOI: https://ieeexplore.ieee.org/document/7764421

[4] D. Stahl, “The dynamic versus the stable team: The unspoken question in large-scale agile development,” WILEY Online Library, Jun. 17, 2023. https://onlinelibrary.wiley.com/doi/full/10.1002/smr.2589#:~:text=The%20importance%20of%20the%20team,approach%20in%20any%20given%20situation. (accessed Sep. 27, 2024).