**Do It**

Todo App

Akash Sukumaran  
 Virginia Tech  
 Blacksburg, Virginia  
 akashsuku@vt.edu

Kesh Reddy  
 Virginia Tech  
 Blacksburg, Virginia  
 keshr[@gmail.com](mailto:email@email.com)

Alejandro Castillo  
 Virginia Tech  
 Blacksburg, Virginia  
 alejandroc01@vt.edu

Tomas Odio  
 Virginia Tech  
 Blacksburg, Virginia  
 tomas22@vt.edu

Nathan Le  
 Virginia Tech  
 Blacksburg, Virginia  
nathanle@vt.edu

**ABSTRACT**

Software engineers often face the challenge of managing multiple tasks such as coding, code reviews, and addressing technical debt. Traditional to-do apps fail to meet the specific demands of developers, lacking essential features that integrate with their workflow and improve task management. To address this gap, we propose developing a specialized task management app tailored to software engineers called *DO IT.*

*DO IT* will simplify task tracking by offering key features such as sprint management, deadline tracking, and productivity enhancement tools. Users will be able to create, organize, and track tasks, set completion times, and receive notifications to ensure deadlines are met. By seamlessly integrating into a developer’s workflow, this app will improve efficiency and streamline task management for software engineering teams.

1**Introduction -**

As software engineers, we understand the complexity of managing day-to-day tasks in a fast-paced environment. The app helps engineers with writing code, fixing bugs, managing pull requests, and handling technical debt. It’s easy for the most important tasks to get lost in the shuffle. That’s where *DO IT* comes in—a task management app specifically designed for engineers, by engineers.

We built *DO IT* to streamline your workflow, from sprint deadlines to the smaller details that often get overlooked. Forget the clutter of traditional task managers that try to do everything and end up doing nothing well. *DO IT* is focused on making task management simple and effective for engineers who need to be both productive and efficient.

You can easily organize tasks based on your specific engineering needs. Whether it’s setting reminders for code reviews, breaking down larger projects into manageable tasks, or tracking your progress in sprints, *DO IT* has the tools to help you stay on track. The app integrates seamlessly with the tools you already use, such as GitHub, JIRA, or Trello, so you can stay connected without extra hassle.

What makes *DO IT* special is its flexibility. You can categorize tasks by priority, deadlines, or project phases. It doesn't matter whether you're fixing critical bugs, managing pull requests, or tackling a new feature, *DO IT* lets you personalize your task board to reflect how you work best. No more switching between different apps or losing track of details. Everything is in one place, clear and manageable.

We know your time is valuable, so *DO IT* is built to help you focus on what matters, which is writing great code and solving meaningful problems. The intuitive interface allows you to create and update tasks quickly without interrupting your flow. Additionally, smart notifications ensure you never miss a deadline, even when you’re working hard on a complex project.

*DO IT* isn’t just about productivity; it’s about making your engineering life easier. Whether you’re working solo or collaborating with a team, *DO IT* adapts to your workflow, making task management feel less like a chore and more like a seamless part of your day.

2**Related Work**

Task management and productivity enhancement in software development have been subjects of extensive research and tool development. A comprehensive systematic literature review examined 34 studies from 2015 to 2022, focusing on techniques, algorithms, and methods for prioritizing software defects and bugs[1]. This review highlights the growing importance of task prioritization in software engineering.

Popular task management tools in the software industry, such as Jira and Trello, offer robust features for Agile development teams. These platforms enable sprint planning, issue tracking, and workflow management using Kanban and Scrum boards. Their integration capabilities with other development platforms like GitHub and Bitbucket streamline the software engineering process, enhancing team collaboration and productivity.

Research on task prioritization in software engineering has become increasingly active. A systematic literature review by Pasikanti et al. found that most task prioritization approaches involve specific strategies, such as machine learning-based techniques[1]. This growth in research signals a positive trend, generating more ideas within the scientific community. However, it also poses challenges for researchers and developers in understanding the current state and capabilities of the field.

The SPACE framework, introduced by Forsgren et al., challenges conventional ways of assessing developer productivity. It emphasizes five dimensions: Satisfaction and well-being, Performance, Activity, Communication and collaboration, and Efficiency and flow[2]. This framework asserts that reducing productivity to a single metric is impossible due to the multidimensional nature of software development.

Task management software has become essential for businesses looking to streamline their workflows. It helps centralize the platform for managing tasks, prioritizing work, setting deadlines, and tracking progress[3]. Such software can be used for project management, time management, collaboration, and real-time monitoring, leading to increased productivity.

The development of task management software typically involves several stages, including defining the scope and requirements, designing the user interface, implementing core features, integrating with other tools, and testing and deployment[3]. Each stage plays a crucial role in ensuring that the software meets user needs, is easy to use, and provides the necessary functionality to manage tasks and projects effectively.

A comprehensive review of task management applications by Colin Wren explored 15 popular todo list apps, including Notion, Tweek, Structured, TimeBloc, and Brite. Wren evaluated these apps based on specific criteria such as the ability to assign tasks to various time blocks (days, weeks, months, years), visibility of tasks within specific time frames, ease of task movement between time blocks, integration with device operating systems, intuitive task and subtask creation, project tagging and progress tracking, customizable week views, privacy considerations, and ownership of data without subscription dependencies[4]

The review highlighted the diverse features and limitations of current task management tools. While some apps excelled in certain areas, none fully met all of Wren's requirements. This analysis underscores the challenges in developing a universally effective todo list application, as user needs vary widely. The study also emphasizes the importance of features like flexible time management, intuitive user interfaces, and data privacy in task management software, providing valuable insights for both users seeking the ideal app and developers working on task management solutions[4]

In conclusion, the field of task management in software engineering is rapidly evolving, with new frameworks, tools, and methodologies being developed to address the complex needs of modern software development teams. Future research could focus on integrating these various approaches to create more comprehensive and effective task management solutions.

3**Software Engineering Process**

For our project we plan to use the Prototyping model for software development. Since we have some basic ideas on what to include in the TO-DO application, but we are not clear on the final requirements, the prototyping model is ideal. The prototyping model will allow us to improve our model based on our peers and developers feedback without much impact on the development process.

The three main reasons why we chose the prototyping model are communication, flexibility, cost-effectiveness and education. Since we are building a to-do app for developers, communicating with the developers and being able to quickly adapt to their feedback is crucial. The Prototyping model allows us to show a visual representation of our app and easily edit. Since the Prototyping model is iterative in nature then it makes it easier to adapt to feedback. Additionally, using a software engineering model that we are not familiar with allows us the opportunity to learn and grow as software developers. Which, as students, is our main goal. Finally prototyping is probably the most cost effective process out of all of them. Since we do not spend resources until we finalize the prototyping process then we can use all of our resources super effectively.

The prototyping model allows stakeholders to engage in the development process, offering valuable feedback during the prototyping period. This reduces the risk of misunderstandings regarding the app’s features and requirements. Using this feedback, we can refine prototypes to better meet with user needs while ensuring the product meets the general design. Another significant advantage of the prototyping model is its support for experimentation and innovation. By quickly testing new features or designs, we can avoid committing significant time and cost to full-scale development. This flexibility enables us to experiment with different UI/UX or task management functionalities, and make informed decisions based on real feedback. This approach ensures a more refined final product.

**REFERENCES**

[1] Bugayenko Y, Bakare A, Cheverda A, Farina M, Kruglov A, Plaksin Y, Pedrycz W, and Succi G. 2023. Prioritizing tasks in software development: A systematic literature review. PubMed Central. Retrieved September 27, 2024 from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10079059/

[2] Taylor Bruneaux. 2023. How Google measures developer productivity. DX. Retrieved September 27, 2024 from https://getdx.com/blog/how-google-measures-developer-productivity/

[3] How to Develop a Task Management Software? (April 2023). Retrieved from <https://www.matellio.com/blog/task-management-software-development/>

[4] 2023. How to Develop a Task Management Software? (April 2023). Retrieved from https://www.matellio.com/blog/task-management-software-development/