Bagel Devs' To-Do List

MACEO CARDINALE KWIK, Virginia Tech, USA
ANNIE TRAN, Virginia Tech, USA
CONOR EDWARDS, Virginia Tech, USA
ARIK SCOFIELD, Virginia Tech, USA
MADDIE OEHLER, Virginia Tech, USA
ZACHARY LAHLOU, Virginia Tech, USA
ESTHER KIM, Virginia Tech, USA

1 ABSTRACT

This report outlines the development of a desktop application called Bagel-To-Do to support software engineers in their work. Key features will help software engineers track tasks, prioritize software projects, with future plans for users to stay focused via productivity timers, and enable better collaboration through the generation of notes for their standup meetings. An iterative approach to gathering user feedback will ensure the tool meets their needs. Risks around version control and team cohesion will be mitigated through communication plans.

Additional Key Words and Phrases: Software Engineering, Productivity, Software, Engineeering, Engineer, Software Engineer, Computer Science, Computers, Work

ACM Reference Format:
Maceo Cardinale Kwik, Annie Tran, Conor Edwards, Arik Scofield, Maddie Oehler, Zachary Lahlou, and Esther Kim. 2024. Bagel Devs' To-Do List. 1, 1 (May 2024), 5 pages. https://doi.org/XXXXXXX.XXXXXXX

2 INTRODUCTION

This report presents the development of a desktop application called Bagel-To-Do that is designed specifically to support software engineers as well as the general public to improve their productivity. The motivation stems from the challenges engineers face working remotely including staying organized, focused, and aligned with the direction of projects and teammates.

The Bagel-To-Do application aims to help engineers by providing key features like task/project tracking to stay organized, prioritized to-do lists to clarify work, and more. These capabilities directly target pain points in engineers' development workflow. Requirements and features were gathered directly from target users via interviews and prototypes with iterative feedback cycles. This engineer-centered approach guided the build-out of functionality, user experience,

Authors' addresses: Maceo Cardinale Kwik, Virginia Tech, Blacksburg, Virginia, USA, Maceo@vt.edu; Annie Tran, Virginia Tech, Blacksburg, Virginia, USA, anniettran@vt.edu; Conor Edwards, Virginia Tech, Blacksburg, Virginia, USA, conore@vt.edu; Arik Scofield, Virginia Tech, Blacksburg, Virginia, USA, ariks@vt.edu; Maddie Oehler, Virginia Tech, Blacksburg, Virginia, USA, mmo815@vt.edu; Zachary Lahlou, Virginia Tech, Blacksburg, Virginia, USA, lahlouzn@vt.edu; Esther Kim, Virginia Tech, Blacksburg, Virginia, USA, estherkdy@vt.edu.

Manuscript submitted to ACM

and overall capability of Bagel-To-Do to best aid engineers' needs. Risks associated with software collaboration like code conflicts and communication issues were mitigated through utilizing best practices in version control management and keeping transparency high across the remote team.

By focusing closely on engineers' needs, the goal is for Bagel-To-Do to meaningfully improve productivity on both personal and team software projects. We plan to help everyone with this flexible and versatile to-do app that does it all from allowing for simple organization of tasks, giving many options for specifics of tasks such as description, recurrence, custom tags for filtering Priority level, and allowing for tasks to be created for teams, giving shared access to tasks. This document will outline related work, high-level design decisions, implementation processes, testing approaches, and how the to-do app will be deployed and maintained.

3 MOTIVATING EXAMPLE

In today's working environment of complex project management, software engineers need to juggle multiple tasks, stay aligned with project goals, and collaborate efficiently with team members for maximum productivity. This need for focusing on project objectives and maintaining productivity is essential in the daily workflow of teamwork.

Bagel To-Do stands as an essential tool that assists teams with a platform that enhances productivity and streamlines workflow processes throughout every stage of development. Starting from creating tasks, organizing sprint schedules, monitoring progress, and adjusting details, Bagel-To-Do serves as a pillar of the project management framework that enhances effectiveness and efficiency among team members.

Consider a scenario where a team of software engineers is tasked with developing a mobile banking application for a leading financial institution. This application will require a comprehensive management system that allows customers to access their accounts, make online transactions, and allow secure management of finances from the user's smartphone. It is crucial to have a solution that meets the needs of banking customers with meticulous attention to detail, organization, and security all in a user-friendly interface.

For example, one member of the team that is responsible for the security protocols of the mobile banking application utilizes Bagel-To-Do to organize system architecture and database schemas. Another member of the team who is responsible for front-end development utilizes Bagel-To-Do to manage tasks related to UI/UX design that includes all steps of the development process. As the project progresses, Bagel-To-Do becomes the application that motivates collaboration among team members. Integration with version control systems allows developers to keep track of code changes and effectively minimizes conflicts with code updates.

Ultimately, the Bagel-To-Do application encourages software engineers to successfully deliver the mobile banking application on schedule with all requirements met. The application's task filtering, version control management, and flexibility in the user-friendly design empower software engineers to navigate complex projects in a collaborative setting.

4 RELATED WORK

The section highlights the significance of task management tools in enhancing productivity for software engineers, drawing insights from research papers on team collaboration and task prioritization to inform Bagel-To-Do's development and features.

4.1 Task Management Tools

In the context of developing Bagel-To-Do, task management tools play a pivotal role. The insights drawn from the research paper "A study on task management system" inform the selection and understanding of these tools [3]. Todoist offers comprehensive task management functionalities, addressing the challenges of remote work with synchronized task tracking and collaboration features. Similarly, Trello's Kanban system aligns with the proposal's objective of improving task organization, providing a visual platform for managing tasks. Wrike also provides a drag-and-drop interface and strong collaboration features.

4.2 Insights from Task Management Research

The proposed Bagel-To-Do app aims to boost software engineers' productivity, drawing insights from the research paper "Enabling team collaboration with task management tools" [2]. The paper explores how task management tools aid collaboration by organizing tasks and tracking progress among team members. Furthermore, the research paper "Prioritizing tasks in software development: A systematic literature review" intersects with Bagel-To-Do's goals [1]. It underscores the importance of task prioritization in software development, ensuring essential goals stay focused. These findings align with our objectives, such as providing features like task tracking and prioritized lists to enhance collaboration, organization, productivity. Bagel-To-Do aims to bridge research insights with practical implementation.

4.3 Novel Features of Bagel-To-Do

However, while existing tools offer valuable features, Bagel-To-Do provides novel functionalities such as customizable task templates, allowing users to streamline task creation for common workflows. With Bagel-To-Do's template feature, users can predefine task details, including title, description, priority, due date, and recurrence settings, saving time and ensuring consistency across tasks.

5 IMPLEMENTATION

Our team used the Prototyping software development process as we recognized the importance of iterating quickly and delivering a functional product. For a relatively small application, this model allowed for rapid development and continuous refinement. Under this approach, individual group members focused on developing specific features, contributing to the evolving prototype. This iterative development process enabled us to gather feedback early and often, ensuring that the application meets user needs and expectations.

As part of our prototyping process, we also employed storyboarding techniques to visualize user interactions and system workflows. Storyboarding provided a structured framework for mapping out the user journey, from task creation to completion, allowing us to anticipate user needs and refine the user interface. By integrating storyboarding into our prototyping workflow, we ensured that Bagel-To-Do's design and functionality were aligned with user expectations.

As for our implementation process, our team leveraged Figma for prototyping, enabling collaborative design iterations and feedback gathering. This platform facilitated the visualization of the application flow, ensuring a user-centric design approach. Additionally, although our prototype did not involve code, testing our user interface involved validation of each feature's functionality with our team members, ensuring a seamless user experience across our different use cases.

Furthermore, we plan to use unit testing, allowing us to validate individual components in isolation, ensuring they functioned as intended. Additionally, we created black box test cases to examine the system's functionality from an external perspective, ensuring that inputs produced the expected outputs without knowledge of the internal workings. This comprehensive approach to testing bolstered the reliability of our application.

For the design decisions of Bagel-To-Do, our team adopted the Model-View-Controller (MVC) architecture, which facilitates abstraction levels and ensures a clear separation of concerns. The model component encompasses classes and methods handling tasks, categories, recurrence settings, and system interactions. Meanwhile, the views component focuses on the visual aspects of the interface, distinct from underlying data management. Acting as an intermediary, the controller manages user actions and application flow, facilitating features like task completion marking, setting recurring tasks, and task filters. This architectural choice guided our implementation process, providing a structured framework for development.

In terms of low-level design, we opted for the Creational design pattern family, which aligns well with the task creation and storage focus of Bagel-To-Do. This pattern facilitates the creation of various task types, such as team tasks and individual tasks, ensuring flexibility and scalability in our application. Overall, through these implementation design decisions, processes, and testing approaches, we created a user-friendly and efficient task management solution for users.

6 DEPLOYMENT

Moving forward, it is important to note how we play to deploy and maintain our project. Our to-do app is unique because it will be used by software developers, meaning that it is crucial that we create a high-functioning application with constant updates. As a result, we would use a rolling deployment strategy, as reviewed in class. This strategy would allow for gradual updates, minimized risks, and also provide opportunities for real-world testing. We would start by deploying the initial version of our application to a subset of servers, or about 10

After initial testing, we will gradually increase the percentage of servers running the new version while still monitoring for any issues. Each server will be subject to health checks to ensure that the new version is functioning properly. If any server fails, it will automatically be removed from the rotation and we will use that information to improve the application on other servers. At the same time, real users will interact with the new version, providing valuable feedback and uncovering any unforeseen issues. Throughout our rolling deployment, we will increment updates. As we gain more experience and confidence with our application, we will deploy to more servers until all are running our updated code.

Our rolling deployment process will allow us to discover issues with our application and easily roll back by directing traffic to old versions of our application. This is specifically helpful for our target audience because of the fact that our audience is software engineers, so they will have high expectations for this application. As a result, it is crucial that we deliver a well-designed application and that we catch any problems quickly and can fix them efficiently.

7 DISCUSSION

However, there are still many limitations to consider for this application. First, software development teams often are consistent with what tools they use, but every team might use different tools. For example, each team might use a different application program interface, or API, throughout their process. As a result, our to-do application would need to be able to integrate with a large and unpredictable number of tools and features. This could prove to be challenging, and may also introduce room for error. Similarly, trying to include too many features in one application may overwhelm users and complicate development. As a result, it is crucial to focus on being specialized rather than general. The goal of our application should be to solve specific problems efficiently.

Going forward, there are many features that would enhance our application. To start, incorporating AI into our program would assist many features. Specifically, an AI task summarizer for standup meetings would make our application more functional. This feature would analyze completed tasks and generate concise summaries for standup meetings. The benefits of this feature include saving time during daily standups, ensuring everyone is well-informed, and highlighting key accomplishments.

Additionally, using AI, we could order tasks based on priority automatically rather than relying on user input on priority levels. This would help users focus on critical tasks first, improving time management and productivity. Similarly, using AI, we could implement a feature that integrates with users' calendars to recommend tasks aligned with scheduled events. Seamlessly aligning tasks planning with real-world commitments would reduce conflict and ensure that all members can attend meetings and other scheduled events.

Overall, these features would benefit the functionality of our application, but they are not necessary. As a result, these features would be implemented after we know that the basic implementation of our application has been perfected.

8 CONCLUSION

In conclusion, this report has outlined the development of Bagel-To-Do, a desktop application designed to enhance productivity for software engineers and the general public alike. Motivated by the challenges in staying organized, focused, and aligned with project goals and team members, especially in remote work settings, faced by engineers, Bagel-To-Do was created to address these pain points through features such as task and project tracking through prioritized to-do lists.

By employing an engineer-centered approach and leveraging insights from related research, Bagel-To-Do was developed to meet the specific needs of its target audience. Moving forward, the deployment strategy for Bagel-To-Do involves a rolling deployment approach to ensure gradual updates, minimize risks, and gather real-world feedback. While the application aims to streamline task management and improve productivity, there are considerations for future enhancements, including the incorporation of AI-driven features to further enhance functionality.

Overall, Bagel-To-Do represents a solution to the challenges faced by software engineers, offering a comprehensive toolset to support their work processes and ultimately enhance productivity and efficiency.

REFERENCES

[1] Yegor Bugayenko, Ayomide Bakare, Arina Cheverda, Mirko Farina, Artem Kruglov, Yaroslav Plaksin, Witold Pedrycz, and Giancarlo Succi. 2023. Prioritizing tasks in software development: A systematic literature review. *PLOS ONE* 18, 4 (2023), e0283838. https://doi.org/10.1371/journal.pone.0283838

[2] Dimitra Chasanidou, Brian Elvesæter, and Arne-Jørgen Berre. 2016. Enabling team collaboration with task management tools. In *Proceedings of the 12th International Symposium on Open Collaboration (OpenSym '16)*. 1–9. https://doi.org/10.1145/2957792.2957799

[3] N. S. Jyothi and A. Parkavi. 2016. A study on task management system. In *Proceedings of the 2016 International Conference on Research Advances in Integrated Navigation Systems (RAINS)*. 1–6. https://doi.org/10.1109/RAINS.2016.7764421

Received 2 May 2024