Standup Survey Bot

Anonymously Request Help

Jordan Atanassov
Computer Science
Virginia Tech
Blacksburg, Virginia
jordanatanassov@vt.edu

Reagan Gary
Computer Science
Virginia Tech
Blacksburg, Virginia
rbgary@vt.edu

Omer Abbas
Computer Science
Virginia Tech
Blacksburg, Virginia
omer18@vt.edu

Jake Lehoullier
Computer Science
Virginia Tech
Blacksburg, Virginia
jakelehou@vt.edu

Sebastain Bukvic
Computer Science
Virginia Tech
Blacksburg, Virginia
sebastianbukvic@vt.edu

Abstract

An inefficiency with IRAD (Internal Research and Development) projects is that it is not necessarily clear what a programmer should be working on, and that meetings are hectic. To

solve this solution, the Better Team bot uses the power of A.I. to listen in on meetings, create a legible transcript of the meeting for those that need to review them after the meeting, and allows for anonymous feedback in the form of surveys. This lets the user contact their project manager and gives them an estimation of their programmer's confidence on completing tasks given to them.

Introduction

According to Scacchi from the University of Southern California, productivity should be measured based on how the application is delivered and when it has been delivered compared to the time constraints given[1]. With the combination of how Scacchi views productivity, and the problem of un-defined problems in software development, new projects are very wasteful in their current states in terms of this productivity measurement.

IRAD's (Internal Research and Development) projects are conducted within a company to either create a new product or improve an existing product, without an external client guiding the project towards a vision. These projects are typically used in companies hoping to use grants given to them by the Department of Defense (DOD).[2] In order to do this, the ill-defined problems outlined in the grants are given to the software companies. With this, there are many issues with the interpretation of how to solve the issue. So, it can end up being a huge sinkhole for many companies financially.

With these inefficiencies, it can lead to logistical issues for programmers. In this case, the Better Team Bot is to be used to combat these issues. For example, if a Programmer attended a standup meeting, received a couple of tickets, and was unsure of why these tickets are tasked to their specific team of their project. They can interact with the Better Team Bot's transcript to make sure they did not mishear the tickets that were being created. And if they don't understand the tasks they were assigned still, they can send a low confidence level on their surveys, allowing for their project manager to be notified that someone is struggling with understanding the goals of the tickets.

Related Work/Ideas

Scrum meetings are used to address what to do and provide scrum master's an idea of how to help programmers solve issues. However, these meetings happen regularly and are frequently rated by programmers to be an issue. The more meetings a programmer has, the less time they must be productive and write lines of code or research solutions to their problems.

Microsoft Teams creates a seamless integration of communciation between different programmers on a team and their potential chains of command. Microsoft Teams groups can be helpful in establishing communication between struggling programmers and can be used to connect with a project manager whenever they are available.

Range is an application created to replace standups in newer projects[3]. It created a way for standups to be asynchronous, and project leads can manually assign goals and tasks to members through it, allowing tickets to be assigned based off to the discretion of the group manager.

Software Engineering Process

The software engineering process that this project would use would be the waterfall. This project doesn't have a lot of associated risks and there's precedent for all the components (client-service communication, databases, audio-to-speech). Since it's easy to see how components are ot be merged, we can assign the project into different phases. These phases would provide the bulk of the planning for the application, with a phase for designing, creating, testing, deploying, and rolling it out to the public.

High Level Design Approach

The high-level design for the better team bot and its approach will be using a client-server architecture. This is used because the clients being the programmers and project managers will be sending requests to either send data, receive data, format data, and/or display the data, while the server will *serve* these requests. The server will handle permissions, authentication, and establish a formal chain of privacy between the users in the company, having different levels of trust based off what position the user is in on the privacy system the administrator sets. This approach allows for the bot to be abstracted from the users, making it easy to use, and not complicated in terms of its interactions.

Testing Approach

The Better Team Bot will focus on using white box testing in the form of unit tests to ensure that specific network requests cannot trigger external functionality. This is significant, especially in a client-server architecture because attacks such as SQL injections and other malicious requests can be used to pry the server for its information. Tests will start with the most basic components, being authentication of users, grow into authenticated users and their requests they can perform. From there, the different responses and different forms of authentications will all be tested with an updating SQL database that is kept under a lock and key. This will allow for the different subsystems to be tested as the code is developed.

Deployment

We intend to divide deployment into 4 steps: staging, building, continuous integration and deployment, and test automation.

We will stage a build of the project and validate its candidacy for full-scale use. This will give us an opportunity to examine other alternative builds if issues arise. For example, if the server can't handle a particular workload will be made apparent here.

Next is the build step where we take the candidate and conduct additional tests among the development team. We will try to replicate the company's use of the bot and see if any issues arise. This is a controlled way of testing the software before it's in the hands of the users, so it's crucial to stress test the system.

We plan to use continuous integration to ensure that any changes in the system are responsively tested. This is important as our software has many working parts and plays an integral role in worker morale and company productivity. We will pay special attention to changes regarding privacy and security as we wish employers to have confidence that their survey response is kept confidential. This phase will also expose any components that, when merged, cause overall system failures.

And finally, we will use test automation. Through this technique, we intend on implementing fuzzing, mutation testing, and chaos engineering. We want the system to be resilient against all types of errors, and making sure that it recovers gracefully from unexpected inputs is the best way to do this. This level of testing should be harsher than how the bot will be used, so if the system has strong mutation coverage, for example, it likely means it's ready for clients. That's how we will know it is ready for deployment.

Maintaining

In the Maintenance phase of this application, it will continue to monitor activity and have outgoing support. Projects and Companies will be able to request additional basic features and functionality for the application. They will also have a portal to send questions and inquiries about these features that need to be either fixed or implemented. There will be security patches implemented and updates pushed to all the repositories as potential security threats and vulnerabilities arise.

Conclusion

The Better Team Bot is a simple application with many limitations. It does not have access to LLM's which makes it not as flexible as those models, not allowing complex queries and such. The bot also has a very UI which will need to be updated to appear better on other formats such as mobile and smaller screens. Another limitation is that it must be connected to the internet. With this, it does not have the specific isolation that government programs might need for clear work in which most of these IRAD's are completed under.

Future work to be done for the Better Team Bot, is to further increase the languages that the bot can transcribe, allowing for the bot to be involved in more companies, increasing its flexibility. On top of this, allowing for the bot to work purely off LAN, allowing for it to be specified on a single network and allowing for it to be its own application rather than a web application would allow it to work for Cleared Workspaces. This would allow all the data to remain inside the network, keeping it out of access from the outside, whether that be a programmer or a project manager that is not physically within the environment.

References

[1] “Async check-ins for teams: Replace your standup,” Range, https://www.range.co/product/check-ins (accessed Feb. 18, 2024).

[2] NDIA, Independent Research and Development, https://www.ndia.org/policy/issues/acquisition-reform/independent-research-and-development (accessed Feb. 18, 2024).

[3] W. Scacchi, “Understanding Software Productivity,” ics.uci.edu, https://ics.uci.edu/~wscacchi/Papers/Vintage/Software_Productivity.html.