Robin: A Voice Controlled Virtual Teammate for Software Developers and Teams

Bruno da Silva, Chloe Hebert, Abhishu Rawka, Siriwan Sereesathien
Computer Science and Software Engineering Department
California Polytechnic State University
San Luis Obispo, California
Email: {bcdasilv, chhebert, arawka, ssereesa}@calpoly.edu

Abstract—Software developers typically collaborate while relying on source code management tools and platforms such as Git and GitHub, which involve process workflow and issue-tracking, for development and maintenance of software-intensive solutions. Developers may lose concentration and productivity when having to navigate through numerous screens to perform daily tasks, which are oftentimes too repetitive, such as applying labels and assigning members to issues, finding who has open pull requests (and how many), closing issues/branches/pull requests, finding open issues, to name a few. Therefore, we hypothesize that current tools need to be improved to increase the developer and team experience and productivity in software projects. Consequently, we have developed Robin, a voice-controlled virtual teammate, to assist developers and teams on projects that strongly rely on source code control and issue tracking. In this short paper, we provide an overview of related work, some design decisions we have made, and insights for the road ahead.

I. INTRODUCTION

Software development is strongly dependent on source code management tools and platforms such as Git and GitHub. Software teams rely on issue-tracking systems, such as Jira or GitHub Issues, to manage tasks during the development and maintenance of software applications. Besides individual coding and testing, software developers usually dedicate a good amount of time working on collaborative activities, thus dealing with version control configurations; creating, editing, assigning, and labeling issues; reviewing code and making comments on pull requests; merging branches; to name a few.

However, software repositories are only partially explored by services such as GitHub and Jira. Developer tools around software repositories are usually provided through web-based user interfaces, frequently accessible through multiple clicks, switching to different screens, thus leading to poor developer experience [1]. Also, these interfaces are not always convenient for teams when they have to make quick decisions and actions during meetings.

Therefore, we propose Robin, a voice controlled virtual teammate, to assist software developers and teams. Robin can listen to voice commands, answer and ask follow-up questions by voice, to perform read/write operations on software repositories and issue-tracking systems, thus supporting multiple scenarios. Robin has been designed for developers either working individually or collaboratively during team meetings. We have gathered requirements for Robin based on literature review, competitive analysis, our own development experiences, and an ongoing developer survey. In the next section, we summarize similar initiatives we have encountered in published papers and in well-known marketplaces for voice apps.

II. RELATED WORK

There are a few attempts to develop voice assistants for developers. Our search on Amazon’s and Google’s marketplaces found eight apps available by the time we write this paper. Lab Assistant1 is the closest to our approach. This app is an Amazon Alexa skill that works for GitLab repositories supporting questions around issue-tracking, todos, merge requests, and CI/CD pipelines. We could not find any documentation related to this app other than their GitHub repository.

Git Commands2 is another app also as an Alexa skill. It works as a general reference guide on how to use Git. Daily GitHub and Repo Head both support GitHub user-related questions. They work as an Alexa assistant for an individual developer supporting high-level commands such as “read my GitHub notifications” or “create/list my repos.” Also, there are particular voice apps for answering simple questions related to GitHub followers (GitHubFollowers app) or trending GitHub repositories (GitHub Trending app), both for Google Assistant.

Finally, there are two educational voice apps (“Git and GitHub Quiz” “GitHub Quiz”) on Google’s marketplace that help you test your knowledge on Git/GitHub.

Additionally, we found three peer-reviewed research papers that somewhat relate to our work. In [2], the authors proposed an automated, on-demand, expert support for software developers. They aim at specific situations individual developers face while having to solve a programming task in isolation, such as “I forgot how to use the syntax for the script HTML tag” or “how to make an AJAX get call from JavaScript?” Their supported queries do not aim at software project collaboration such as questions related to issue tracking, pull requests and code review, or other general repository related aspects that assume teamwork. Also, they have not provided voice support.

In [3], the authors proposed a text-based bot to support software developers in answering questions related to software

1https://gitlab.com/nfriend/lab-assistant
2This and other voice assistant apps cited in this paper can be found on Amazon’s or Google’s marketplace depending on your country.
repositories. Although their work did not attempt to create a voice interface, it shares some similarities with our approach. The set of questions they support are related to project repositories assuming developers engaging in collaborative work involving bug fixing, issue tracking, and multiple commits from different contributors in a project. They provided a list of 15 supported questions deemed relevant based on the software engineering research literature. Their work also included a case study with 12 participants.

In [4], the authors proposed Devy, a voice controlled application, implemented using the Amazon Alexa platform, to help developers offload low-level actions to an automated assistant. Devy’s approach supports context-aware actions, assuming the developer is sitting on their machines with a task associated with a particular context in their workstations (e.g., changing a file on the IDE, checking out or merging a branch). To date, Devy is not available either on GitHub or on Amazon marketplace.

Our work aims at providing a voice controlled application rather than a text-bot as in [3]. We use the teammate metaphor to assist developers associated with software repositories. Moreover, instead of focusing mostly on questions around bug-tracking as in [3], we intend to support scenarios where developers are individually carrying out tasks or collectively discussing, making decisions, and managing project repositories, branches, and pull requests, during team meetings or collaborating asynchronously (not necessarily focusing on bug-tracking). Besides implementing Robin on Amazon’s platform as in [4], we are also implementing Robin using Google Assistant. These are the two most popular platforms to date. Beyond supporting questions as in [3], we have also covered commands that write on the desired repositories, which is also somewhat implemented by Devy [4]. However, Robin focuses on questions/commands that might be useful for developers and teams while collaborating synchronously or asynchronously without looking at context data loaded on the developers’ machine. In contrast, Devy assumes developers will be sitting on their computers, providing contextual data based on the files they have open on their IDEs or repositories and branches they have checked out.

III. Robin’s Architectural Overview

On Figure 1, we show an overview of Robin’s architecture. The depicted structure accounts for both Alexa and Google Assistant implementations. The first box from left to right represents the interaction layer with end-users. Both platforms provide native support for voice interaction: input and output via built-in speech-to-text and text-to-speech conversions, respectively. The voice commands and questions are then mapped to intents that are part of a dialog model in both platforms (middle layer box). Given the mapped intent, the assistant then works to take one or more actions for the user (e.g., answering a question or performing a write command). We fully implement these actions in our fulfillment backend component (outer box on the right-hand side). We divide this component into three parts: the webhook, the business logic subcomponent, and the GitHub API. We implement the webhook service, which is hosted on the cloud, and it receives requests from Alexa and Google Assistant mapped intents. The webhook subcomponent works as a dispatcher that delegates calls to our backend subcomponent that implements all the business logic involved in the intent fulfillment. The business logic eventually triggers one or more calls to the GitHub API to gather data from or to perform write operations on repositories.

We have implemented automated test cases for all the components we have control over in the architecture. Future work will involve user studies with developers. In this first iteration, our focus has been on conversations around issue-tracking and pull requests, since these elements are heavily used in highly collaborative projects, where a voice controlled teammate would be most useful.

IV. Road Ahead

Developers generate a massive amount of data around software repositories beyond source code, such as issue tracking, code review comments, and process workflow. Moreover, developers, either working individually or collaboratively in teams, take a lot of actions navigating through multiple screens, tools, and context switches. With recent advancements in voice assistant technologies, researchers and practitioners have started to build voice assistant solutions for various developer needs. We have implemented Robin, a voice controlled virtual teammate, for developers and teams. To date, we have supported questions and commands around repositories, pull requests, and issue tracking. Robin has been developed on Alexa and Google Assistant platforms and is compatible with GitHub. We have an ongoing online survey to collect developers’ opinions and insights about the usage of such a voice assistant for different scenarios. That will feed our process of choosing and implementing the right commands and questions in future iterations. Moving forward, we plan to keep investing in questions and commands around collaborative aspects in software development, that could better support developers working on co-located and geographically distributed teams. We have designed Robin based on literature research and empirical investigations, such as our ongoing online developer survey. In the future, we will perform a user study with developers using the very first release of Robin.
REFERENCES


