Code Analyzer

Control+Alt+Defeat

Trevor Jarzynka
Computer Science
Virginia Tech
Blacksburg, VA, USA
trevorj04@vt.edu

Zekun Li
Computer Science
Virginia Tech
Blacksburg, VA, USA
zekunli@vt.edu

Cooper Robson
Computer Science
Virginia Tech
Blacksburg, VA, USA
cwrobson10@vt.edu

Ben Fielder
Mechanical Engineering
Virginia Tech
Blacksburg, VA, USA
benf02@vt.edu

ABSTRACT

In software engineering there are two major problems holding developers back, time spent understanding code and adding onto existing code. Companies currently try to mitigate these problems by holding team meetings, code reviews, and constant communication. Our solution is an algorithm that finds similar coding styles amongst company teams. It will provide a visual representation within a website that shows who codes similarly and what style techniques they most commonly use. In addition, our program will allow for database searches within a company's code to allow software developers to search for solutions to their problems before going to the internet or other sources. We believe that this program will optimize software developers' time and more innovation will come from it.

INTRODUCTION

Within software development a major problem lies in when teams expand and it is, coding together. This is an issue companies try to mitigate since team development leads to larger and higher quality innovation when compared to singular development. Techniques such as team meetings, code reviews, and constant communication are used to help team members combine coding knowledge to produce an innovative product in a small amount of time . These current techniques are good at mitigating this issue but there is still a lot of time wasted understanding others' code which is difficult to do. Our program will provide an online dashboard showing team members coding styles. This will allow project managers to better understand their developers and create more efficient teams within their company. In addition, this can be implemented in the hiring process to assist in finding the best fit for the job.

The way in which we will accomplish this is by tapping into a company's code database like git or similar technologies to analyze all code and who commits the code. After this process, our natural language processing algorithm will upload profiles for each employee to the online dashboard. The profiles will include styles for major coding techniques which are customized by the employer. For example, commenting style, how they utilize if statements, for loops and what algorithms are commonly used like Dijkstra's or Bellman-Ford. In addition, we will implement a code database that stores small snippets of code written by people within the company to mitigate copying online code which can lead to cyber security issues.

With all of these features we believe that companies will want to use this tool as it benefits code development without requiring drastic change to existing code development. The cyber security aspect is another part of this program that companies are in need for solutions as cyber security is one of the largest issues within software engineering. With that said our program will help mitigate two major problems within software engineering and companies that use our product will produce innovative products quicker and more secure.

Related Work

Research Studies:

Besides conducting our own research we found a research paper titled *Today was a Good Day: The Daily Life of Software Developers* to further highlight the issue our program will solve. In this study about 32% of a software engineer's day at Microsoft was spent actually coding with a sample size of 5,000 employees. With the same sample size it was also found that 35% of their time was spent reading, reviewing and understanding code written by other team members. This highlights the importance of this issue since more time is spent understanding code then writing it at one of the largest software companies in the world Microsoft.

Existing Programs:

Current programs don’t currently exist that do what our program will do. Similar software programs are Git, SonarQube and JetBrains Qodana. Our program has a unique focus that SonarQube and JetBrains Qodana programs don’t currently have which is company software development and Git does not currently have code analysis for large companies as well. We believe this highlights an opportunity for a program like ours to fill a need for companies.

Software Engineering Process

The software engineering process chosen was the Agile method. This method allows for projects to be broken down into phases or sprints. This means that increments to be made and it is the best for working between client and the team. After every sprint the team should be able to look back and see what needs to be changed or improved. This helps them adjust their strategy for the next sprint.

We chose this method because of the many benefits that an interactive and incremental approach can provide to development. With a project like ours, there are many unknowns at the start of development. Questions like whether we will use API’s or our own algorithm to group teams and suggest changes, what type of programming language we will use, and whether we will add any additional features later on, are all things that we need to think about and answer at some point in the development process. One of the biggest benefits of an iterative approach is that we don’t have to have all of those questions answered at the start of development. Using an Agile method, we can determine the best solutions to our problems along the way. Other methods like spiral or waterfall wouldn’t help us in any major ways. We have no huge risks that we are dealing with and there are no components of our problem where it is critical that we get it right the first time and not make mistakes.

In conclusion, although there are many great software engineering processes, the Agile method is the best choice for use because of the iterative nature of the method, the lack of risk in our project, and the vast array of unknowns that are facing us right now.

References

"Asana." Agile Methodology: What Is It and How It Works,Asana, https://asana.com/resources/agile-methodology.
Accessed 27 Sept. 2024.

Meyer, André N., et al. Today Was a Good Day: The Daily Life of Software Developers.Microsoft Research, Apr.2019, https://www.microsoft.com/en-us/research/uploads/prod/2019/04/devtime-preprint-TSE19.pdf. Accessed 27 Sept. 2024.