

# Technical Debt Tool Comparisons

Basil Roy

School of Enterprise Computing and Digital Transformation, TU Dublin, Ireland

X00193213@myTUDublin.ie

## Introduction

Technical Debt can be sometimes known as tech debt or code debt. Technical debt basically describes what results when development teams take certain actions towards technical implementations to speed up the delivery of a certain functionality of the project to meet business deadlines set which will need refactoring in the future. Briefly Technical Debt can be put forward as the outcome of prioritizing speedy delivery of code over perfect quality code. If you are an individual working in the software industry this is where technical debt creeps up mostly. To tackle this debt there are various technical debt tools available in the market. With Technical debt there are a few risks associated with it. With risks there comes impact. These factors are to be considered as they are very crucial in the long run and on the sustainability of the applications in question. With the usage of technical debt tools, we can certainly maintain or erase technical debt altogether from our environment, but a thing to look out for is that there are a lot of tools available in the market. How can we determine which is the correct one for our team / organization. That is the challenge we are facing.

## Research Question

**RQ1** : What factors mainly influence a user when deciding to Onboard a new tool ?

**RQ2** : Do users fall into the trap of selecting the most popular tool in the market and not consider the requirements at stake ?

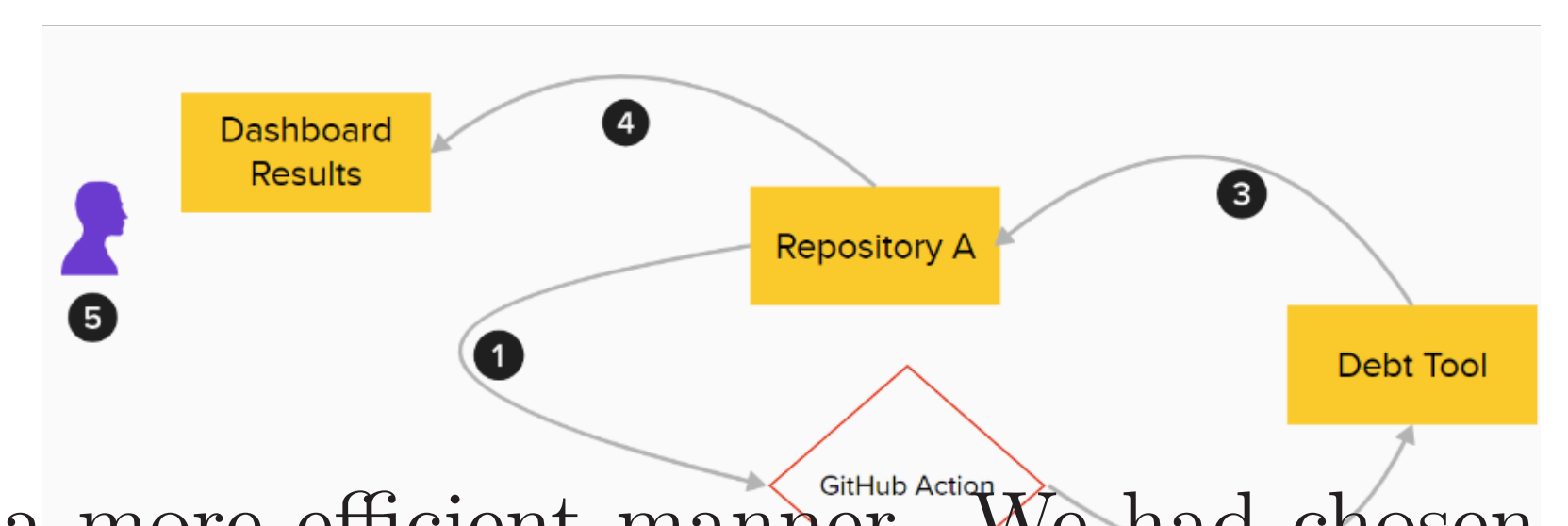
## Weighted Factor Results

Factors	Weighted Score of 10
Cost	7
Features	8
Adaptability	9
Usability	8
Support	10

## Testing Methods

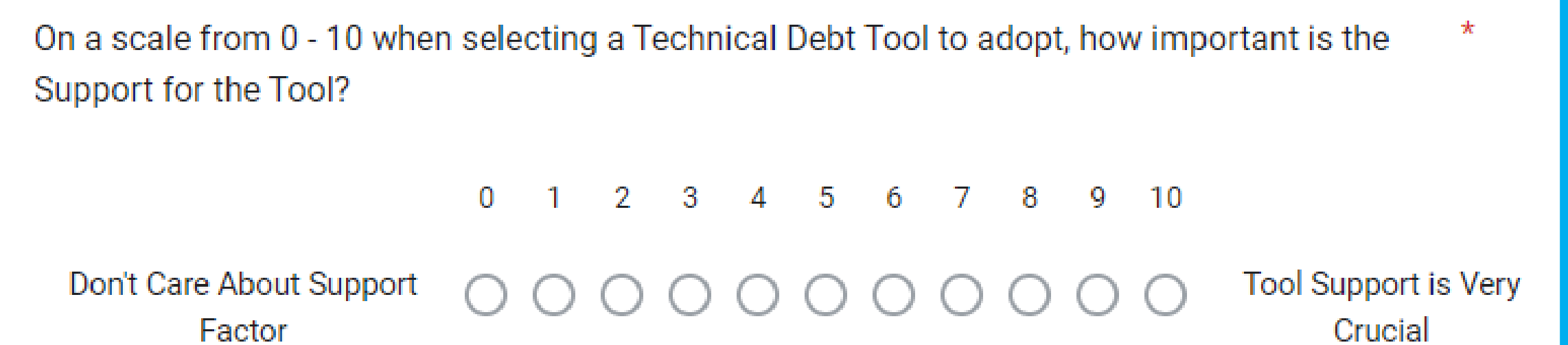
### 1. Testing Design Flow:

This is the testing design followed for the experiments. It is a simple flow which is repeatable for the 4 repositories that was chosen for this test. The base of the flow starts from our repositories as that is our primary location of code base to scan. We employed Github Action feature which GitHub provides to aid us to setup up scanning in a more efficient manner. We had chosen 3 Tech debt tools for this comparison which were Sonar Cloud, Synk And CodeEql. With the use of Github Actions we can use a template yml file which has predefined steps for specific tech debt scan. But we had to modify those files to meet our requirements. Since this was a yml file there are options to trigger the running of the file on commit or manual basis. With this option it allows us to achieve Continuous Integration for running the scan on each commit user do to maintain a healthy tech debt backlog. Once this setup is complete we focus on any configuration needed to make with the repository itself to allow to scan the repository. As there will be permission required and addition of secrets/tokens which is a way the Tool communicates with the Repository. Once the Scan is done we head to the Tool Dashboard for each individual tool and here the users can analyse the scanning results. There maybe stats available which can aid users to see the overall health of the application.

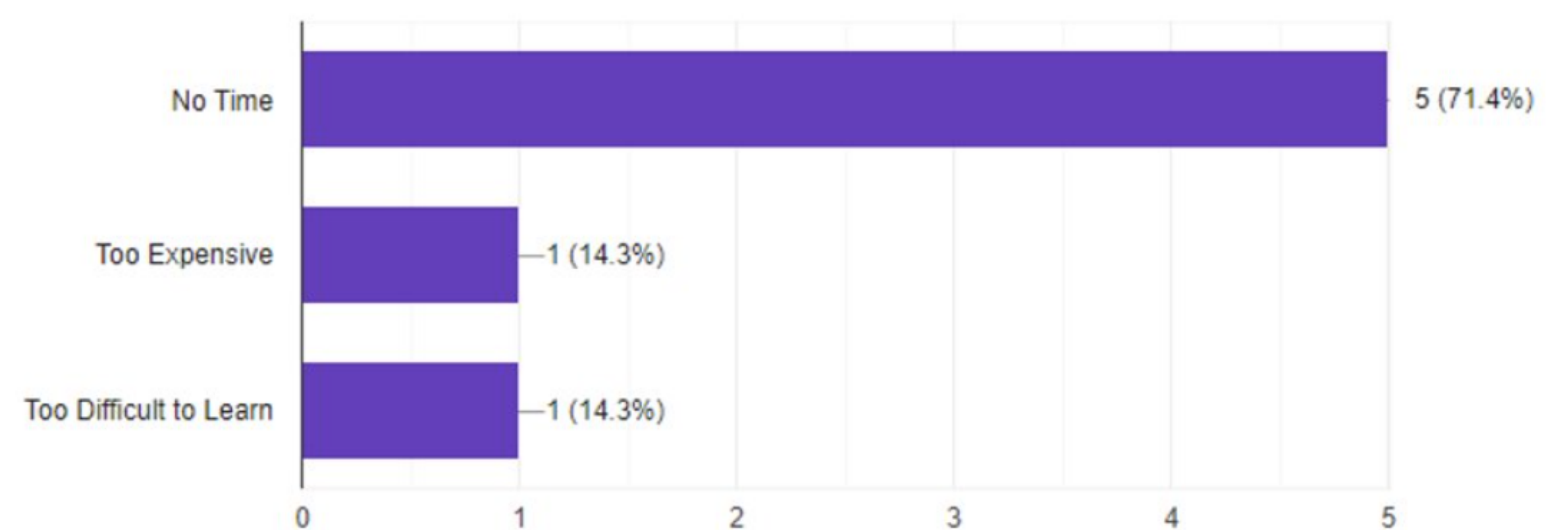
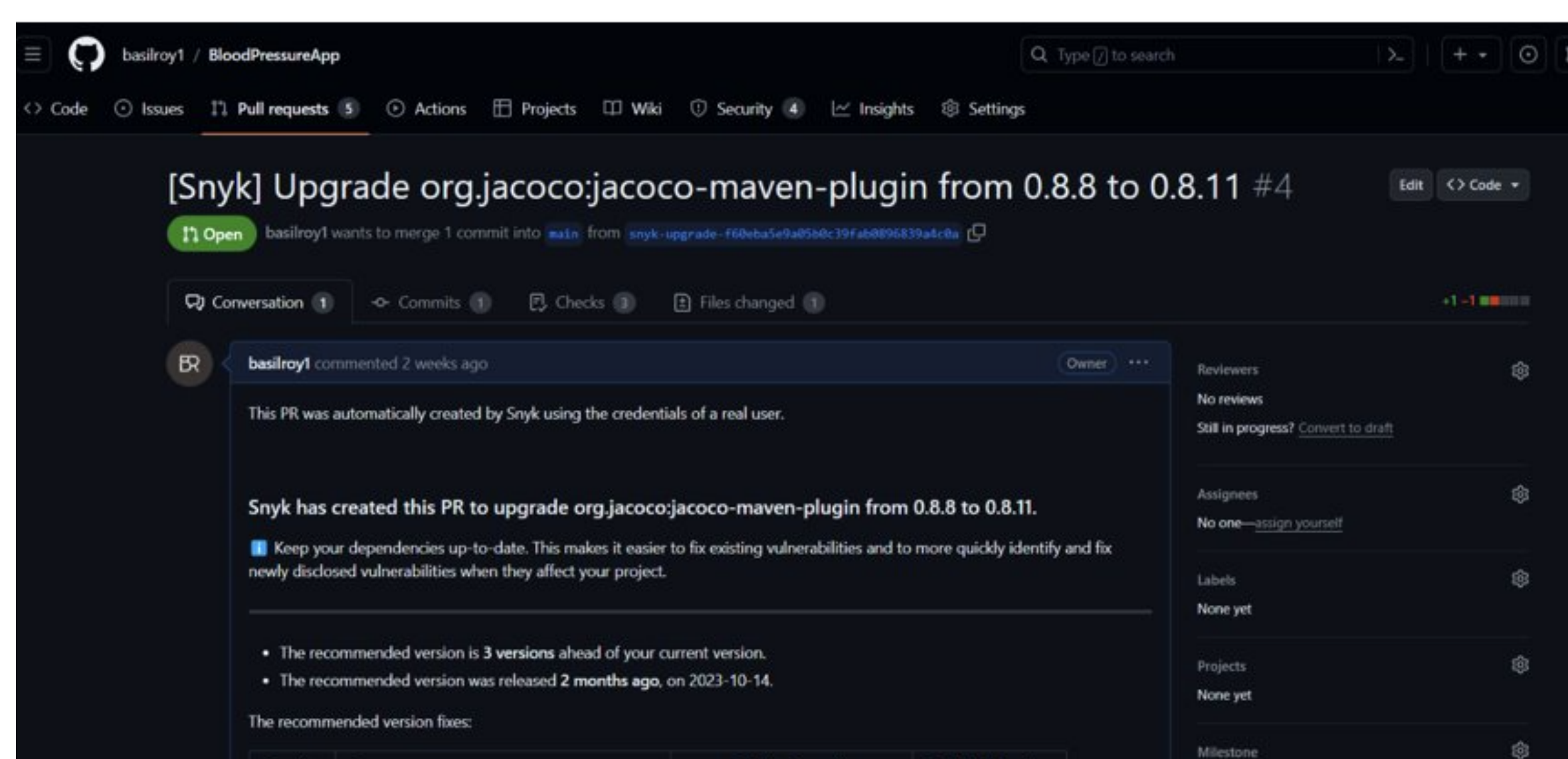


### 2. Questionnaire:

Survey was an important factor to the data collection for our experiments. For conducting surveys a Sample of users for participating in the survey. Simple random technique was used to collect our sample of 23 users for our survey.



## Key Research Findings



## Conclusions and Future Work

This research had few limitations such as repository code languages used and only the free trial versions of the tools that's only available to be used for the testing which may have limited features and other reports facility available. In some cases, for other tech debt tools even to get access to the free versions we would need to request access by filling a form in which we have no definite time defined for when the expected response would be, which also limits the usage of other tools in the market.

One area I felt from the research and existing tools is with current boost in the AI sector. So as an area of future works can be what impact does AI (Artificial Intelligence) bring to the current technical debt tools. AI is picking up pace extremely fast in the past year or two. AI is certainly influencing all aspects of the software development cycle. There are AI plugins for the IDE used for development. ChatGPT being the one of the contenders which basically writes code from scratch which is a huge win over the productivity aspects with reduced lead time. Similarly, there are certainly interesting aspects which AI will change the current way of doing things for scanning repositories.

## QR Code for Recording

