**DevOps IDE Investigation Report**

# Introduction

As part of the pipeline required for the Rogue Event project, the Agile team must establish and agreed set of application software and plugins to be used in completion of code. During the initial Scrum meeting, it was mentioned that the task of garnering enough information about the popular integrated development environments (IDEs) available on the market to make a voted choice on which to utilise going forward.

Below will be a list of three of the most popular platforms globally to be chosen from by the team. For the completion of this specific pipeline, the options were voted to be free to use by the creator as to not create extra cost where it would be considered unnecessary for the project.

# Choices

## Visual Studio Code



Figure - Visual Studio Code logo

The first choice is Visual Studio Code (VSC) coming from Microsoft Studios. As many team members already make use of this IDE during their years at the establishment, it was an immediate option that was discussed at the initial Scrum meeting. Although it does not natively support, nor was it created specifically for Java programming, there is a vast array of extensions that will allow for the platform to be used as such. Not all team members are within the same course, so it will be down to a vote as to what the final choice will be. Below will be a list of pros and cons for this development environment.

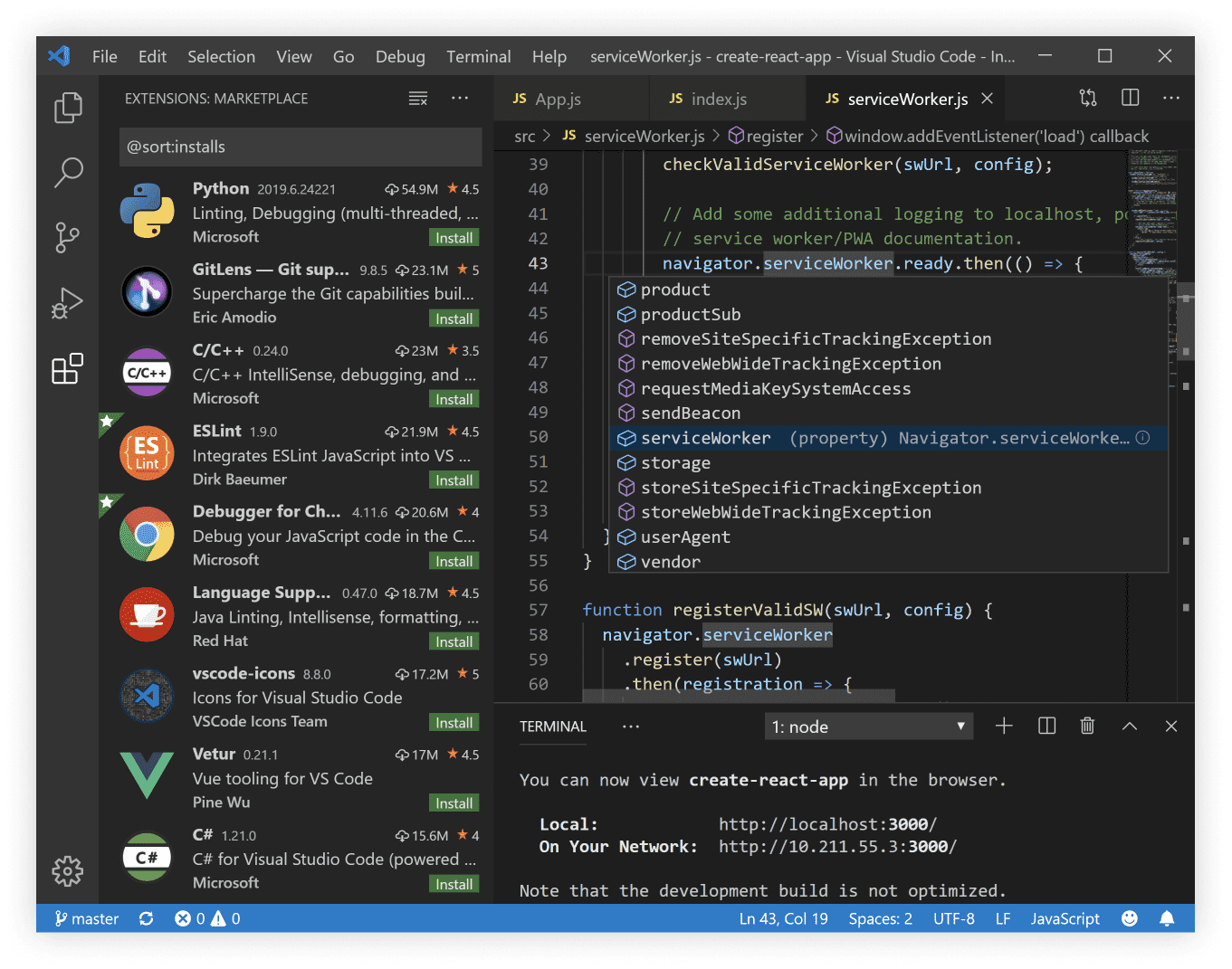


Figure - Example of the environment showing code and potential plugins/extensions to use.

### Pros & Cons

+ Lightweight and highly adaptable environment.

+ Open-source application, free to use.

+ Access to collaborative extensions for multiple users to code simultaneously.

+ Can install extensions to allow for debugging, testing and integrated Git control.

* Environment not designed for use with Java inherently, requires extra installation steps.
* Requires further information on necessary extensions for debugging, “IntelliSense” etc.
* Other options that were designed specifically for Java development are freely available.
* Not all team members have prior use, or knowledge in this environment.

## Eclipse



Figure - Eclipse logo

The next product comes from the Eclipse Foundation, and as of the current year (2022) holds around forty eight percent of the market share globally for usage. This also comes with a four-point eight user rating and a ninety-two-user satisfaction review.

Only around half of the team are familiar with this development environment from the prior “Software Implementation” and “Secure Programming” modules which focuses on introducing unit testing created software. The modules also focused on the introduction of plugins for resolving errors, highlighting security issues within the code, and effectively documenting the features being created.

The learning curve for this specific platform may be a little higher as compared to alternatives and the current pipeline is already time sensitive. Again, the pros and cons will be listed below.

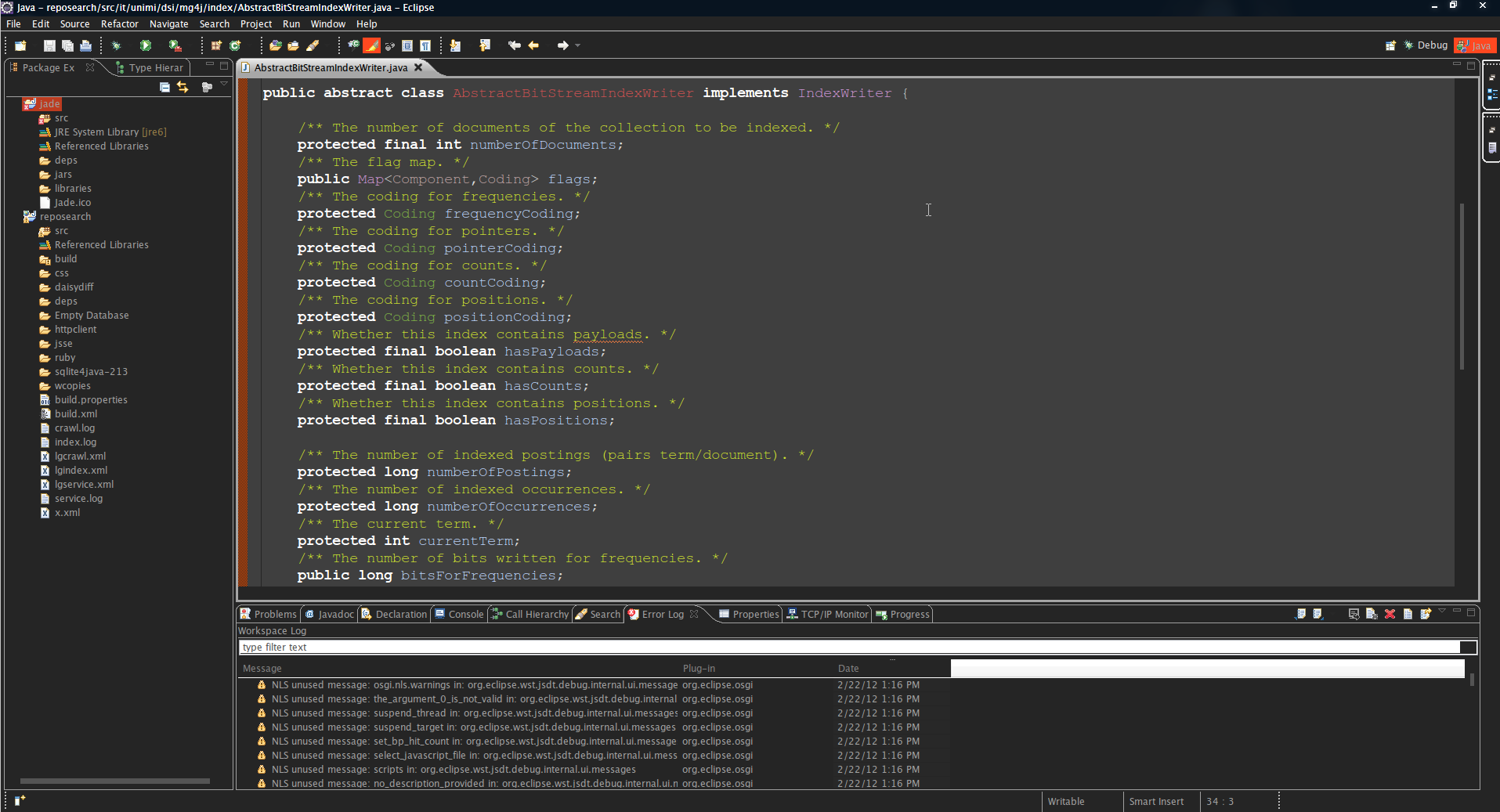


Figure - Example of Eclipse development evironment showing directory tree.

### Pros & Cons

+ Designed from the ground up for Java based programming with an array of built-in features (IntelliSense, Code Formatting, Debugging, Error Handling, Refactoring).

+ Eclipse has a solid plugin system that enables developers to create unique features with built-in marketplace.

+ Ability to work cross platform where necessary for team members that could be on Linux or Mac OS.

+ Free to all users, stated as open source based on online information.

* Not as modern in design/UI as compared to alternative choices.
* Setup can be quite difficult, not as straightforward with directories and components.
* Garnered a reputation for running at a slower pace, with higher CPU usage allocation.
* Can frequently crash depending on project volume.

## IntelliJ IDEA

A picture containing text, sign, businesscard, vector graphics

Description automatically generated

Figure - IntelliJ logo

The final option comes from the company “Jetbrains”, with the IDE named IntelliJ IDEA. By 2022, the software now has over thirty three percent of the market share and is considered the most used development environment of the current year. It also retains around eighty-nine percent user satisfaction based on survey reviews. The IDE contains unique resources such as a built-in version control system, frameworks and allowing for support of multiple programming languages outside of Java.

Text

Description automatically generated

Figure - IntelliJ main screen showing file structuring and plugin list

### Pros & Cons

+ Strong customization features.

+ Built-in version control support.

+ Strong plugin and integration support.

+ Powerful compiler.

+ Grade build system.

* Ultimate version can be costly outside of community edition.
* Can be heavy on system resources.
* Steeper learning curve compared to simpler IDE choices.
* Documentation of tools could be improved.

# Conclusion

As the team met for the first Scrum meeting and deliberated over the various choices, it had become more prevalent of each courses actual time spent with certain IDE platforms. Once running through the more apparent options, it was decided that IntelliJ IDEA from Jetbrains would be the forerunner due to the ease of use. Also, it’s available plugins marketplace, and focus on the Java language as compared to something like Visual Studio Code which could take longer to adapt to with the required plugin support to be installed prior to initial coding being done. The team had a vote and IntelliJ was chosen as the winning candidate.