Eclipse IDE vs. Visual Studio IDE

By: Amanda Sedgwick

**Table of Contents**

Abstract…………………………………………………………………………………………………………………………………..…3

Introduction……………………………………………………………………………………………………………………………….4

Decisions in Design Development…………………………………………………………………………………………………………………………..…4

IDEs and More……………………………………………………………………………………………………………………………5

Figure 1………………………………………………………………………………………………………………………………………6

Eclipse IDE………………………………………………………………………………………………………………………………….7

Visual Studio IDE………………………………………………………………………………………………………………………..8

Comparing the two IDEs…………………………………………………………………………………………………………….8

Figure 2……………………………………………………………………………………………………………………………………10

Conclusion……………………………………………………………………………………………………………………………….10

References……………………………………………………………………………………………………………………………….12

**Abstract**

I am still working on the abstract.

**Introduction**

Tools. Something most jobs require, whether it’s actual hardware tools, medical tools, cooking utensils, paintbrushes, or computer software. One of the most important tools for a web developer is an integrated development environment, aka an IDE. An IDE is a software application that provides a wide range of tools for a web developer. This report will analyze and compare the specific tools of 2 popular IDEs: Eclipse and Visual Studio. An examination of decisions in design development, IDEs and more, Vital Features of IDEs, specific features of both Eclipse and Visual Studio, as well as a comparison of the two IDEs will be conducted in this analytical comparative report. A variety of IDEs exist in today’s technology; however, many developers choose to work with Visual Studio or Eclipse. As a web developer, which IDE is better? Eclipse or Microsoft Visual Studio?

**Decisions in Design Development**

Similar to constructing an outline for an essay, Developers may create a design document to help maintain focus on the expectations of the program. “Developers work with design decisions to accomplish specific goals, both when initially choosing between alternatives to make a design decision as well as when returning to a previously made decision to understand how to follow it. For example, a developer may have a goal to understand the rationale behind a design decision and the reasons for choosing one alternative instead of others. Accomplishing this goal can build a correct mental model of a codebase and to ensure that future changes and decision-making follow the intended design” (Mehrpour, 2024, p.37:2). Before using IDEs and other developer tools, it is good practice to construct a design document, which consists of the major design decisions for the functionality of the program. In other words, plan out the logic of the program code, how to accomplish tasks the code should perform in the end. Planning design decisions can help a developer stay focused on the intended expectations of a program, as well as to evaluate the best way to solve problems in the code. Staying organized with a design document helps create structure for the program code.

**IDEs and more**

Software development requires extensive knowledge of coding processes. “Software developers usually use an integrated system of components such as code editors, compilers, and debuggers, to build a software product. During this process, developers often need to acquire external support such as code snippets, which can cause the developers to switch frequently between a development environment and knowledge environment, usually implemented through a browser” (Melo, 2020, p. 2). Code editors, compilers, and debuggers are all tools most IDEs offer. Combined with the IDE (development environment), software developers may also need to refer to knowledge environments provided on the web. Many IDEs offer IntelliSense or code completion to suggest solutions to the code. However, because Software Development requires such extensive knowledge, there are many websites for programmers to refer to as well, for code suggestions.

**Figure 1**

A screenshot of a computer program

Description automatically generated

The term IDE is short for Integrated Development Environment. An IDE is a software application that provides tools to aid in program development as both a code editor and compiler. The developer tools help programmers create working code more easily through debugging tools, code completion and other tools. IDEs offer Syntax highlighting that notifies users of typos, incorrect usage of language rules.

Code assistance helps with code completion, syntax highlighting and raises errors to be addressed.

IntelliSense is the functionality of code completion, giving the programmer suggestions and shortcuts to save time typing out lines of code.

Language Support is just as it sounds, typically a popular IDE can support over 50 different coding languages.

A compiler aids in execution of code by communicating the code interpretation to the computer.

Debugging tools assist in error resolution, providing suggestions and clues, as well as the ability to “Step Through” code to find the precise line of code halting successful execution.

Code Refactoring includes the ability to adjust code without affecting the desired output or functionality of the code. This helps increase efficiency and readability.

Compatibility just means that IDEs support a wide array of languages and extensions.

Version Control also supports efficiency and the tracking of changes to code (Pujara,2023).

**Eclipse IDE**

Eclipse is a well-known IDE used by millions of developers. The Eclipse Foundation was created in January of 2004, and it is a non-profit organization. The eclipse foundation offers 4 key services. These include IP Management, Ecosystem Development and marketing, Development Process, and finally IT Infrastructure. The foundation’s focal point is to produce successful open-source projects for open-source development. The eclipse project is used by millions of developers (Eclipse, n.d). Eclipse Theia 1.0 was released in 2020, it is an Eclipse IDE used by developers as a popular alternative to the Visual Studio IDE. “Eclipse Theia is an extensible platform to develop multi-language cloud and Desktop Integrated Development Environments (IDEs) with state-of-the-art web technologies that enable developers, organizations, and vendors to create new, extensible developer experiences” (Nichols, 2020). Eclipse Theia accommodates many customizations, it can run on desktop and cloud, it runs frontend and backend processes, and it allows multiple extensions or add- Ons to the IDE to remain compatible with a variety of coding languages. Eclipse has adjusted its IDE features to accommodate the everchanging world of technology. In conclusion, Eclipse IDE offers a developer many up to date tools to aid in the development process of a programmer.

**Visual Studio**

Microsoft Visual Studio in comparison to Eclipse IDE is equally as popular and used by many developers. Visual Studio is productive, modern, and innovative. Visual Studio now also has the option to code in the cloud, meaning that developers can code anywhere on any device. Like many other IDEs, Visual Studio includes an option to code with IntelliSense (code suggestions). When a developer is coding, the IDE may give suggestions of code based on the previous code in the program. It attempts to understand what you are trying to accomplish. Integrated debugging allows developers to “step through” code, allowing them to examine the output of each statement or line of code. This will help understand where the problem in the code is occurring. Visual Studio also allows “Live Share”, this is a feature that allows teams of developers to view and share code. A variety of tools are included with Visual Studio to help improve the code faster (Visual Studio, 2024). Visual Studio IDE offers many great developer tools to make coding more efficient. Visual Studio has implemented many new features over time to improve the coding experience of the developer.

**Comparing the two IDEs**

When comparing Eclipse vs. Visual Studio at a user standpoint, both seem to offer similar features for programmers. Eclipse IDE has 180 reviews and a 4.34/5-star rating overall. Microsoft Visual Studio has 2885 reviews and a 4.62/5-star rating overall. Eclipse’s ease-of-use rating is 4/5, while Visual Studio has a 4.5/5 rating. Both IDEs scored 4.5/5-star rating for Value for money. Both also scored 4/5 for customer support. In addition, both IDEs received the same rating for Functionality at 4.5/5 stars. Some listed Pros of Eclipse are as follows. Easy to use GUI (Graphical User Interface), which is basically the interaction between the user and the computer. Another Pro of Eclipse is that it has reliable integration and multiple options for “Plug ins” or extensions to utilize additional features. However, some Cons listed include the following. Upgrading Versions may be difficult, and it is a little outdated. It is also mentioned that if developers are not familiar with using this IDE, it might be difficult to navigate the project explorer. On the other hand, some Pros of Microsoft Visual Studio are as follows. It is easy to use and easy to start/create projects. It has also helped developers improve the quality of software through code analysis. Some Cons of Visual studio are as follows. It may be a boring coding process because it doesn’t have “Plug ins”. Some developers have also complained of Visual Studio lagging, resulting in having to restart the software to resolve (Eclipse IDE vs Microsoft Visual Studio, 2024). In general, the rating for the Eclipse IDE is slightly lower than the rating for Visual Studio. However, Visual Studio has been around for longer than the Eclipse IDE, but it compares closely to Visual Studio in the features it offers developers.

**Figure 2**

A table with text and words

Description automatically generated with medium confidence

Eclipse IDE includes a variety of features and plugins and is compatible with a wide range of programming languages. Eclipse is a popular open-source IDE with integrated tools for debugging, code refactoring and auto completion.

Visual Studio is also a very popular open-source IDE that supports various extensions. It is also compatible with a variety of programming languages. This IDE is known to be lightweight, fast and offers IntelliSense to aid in code completion.

Eclipse and Visual Studio are very similar in the features they provide. So, it really comes down to the developer’s preference in the layout structure and ease of use for that particular individual (Obregon, 2023).

**Conclusion**

In conclusion, design decisions are vital to prepare web developers for creating successful code. IDEs offer an integrated system of components, such as code editors, compilers and debuggers that assist developers in building a software product. Vital features of an IDE are syntax highlighting, language support, code refactoring, code assistance, a compiler, IntelliSense, compatibility, debugging tools, and version control. Eclipse IDE offers a variety of extensions, allows for customizations, can run on both desktop and cloud, and is compatible with a wide array of coding languages. On the other hand, Visual Studio is productive, easy to use, modern, innovative and it offers a feature called “Live Share”, as well as a variety of other standard IDE components. Both IDEs offer many of the same features for developers and both scored very similar ratings by programmers that use them. Visual Studio has progressed so much over time and Eclipse has caught up in a shorter amount of time. Which is better? Well, that is in the eye of the beholder, both are excellent sources for any developer.

**References**

APA Citations

Eclipse (n.d). Eclipse Foundation AISBL.

<https://www.eclipse.org/org/>

Eclipse IDE vs Microsoft Visual Studio.(2024). *Software Advice*.

<https://www.softwareadvice.com/ide/eclipse-ide-profile/vs/microsoft-visual-studio/>

Mehrpour, S., & Latoza, T. D. (2024). A Survey of Tool Support for Working with Design

Decisions in Code. *ACM Computing Surveys*, *56*(2), 1–37.

<https://doi-org.ezproxy.wctc.edu/10.1145/3607868>

Melo, G., Oliveira, T., Alencar, P., & Cowan, D. (2020). Knowledge reuse in software projects:

Retrieving software development Q&A posts based on project task similarity. *PLoS ONE*, *15*(12), 1–27.

<https://doi-org.ezproxy.wctc.edu/10.1371/journal.pone.0243852>

Nichols, J. (2020, March 31). The Eclipse Foundation Releases Eclipse Theia 1.0, a True Open

Source Alternative to Visual Studio Code – Leading open source adopters for Eclipse Theia include ARM, Arduino, EclipseSource, Ericsson, Gitpod, Google Cloud, IBM, Red Hat, SAP, and TypeFox. *GlobeNewswire(USA).*

<https://infoweb-newsbank-com.ezproxy.wctc.edu/apps/news/document-view?p=AMNP&docref=news%2F17A108B15623D4D8>

Obregon, A. (2023, May 1). Eclipse IDE vs. Other Popular IDEs: A Comparative Analysis for Beginners.

*Medium.*

<https://medium.com/@AlexanderObregon/eclipse-ide-vs-other-popular-ides-a-comparative-analysis-for-beginners-6bb692a76535>

Pujara, K. (2023, November 9). What is an IDE?- Guide to Integrated Development Environment.

*MultiQoS.*

<https://multiqos.com/blogs/guide-to-integrated-development-environment/>

Visual Studio(2024). Visual Studio.Microsoft.com.

<https://visualstudio.microsoft.com/vs/>