

## Cloud-Based Lightweight Modern Integrated Development Environments (IDEs) and their Future

Abhishek Shukla

USA

### ABSTRACT

Integrated development environments (IDEs) have been considered one of the most important tools for developing software. Integrated development environments (IDEs) act as a toolkit designed to facilitate the whole process of designing and developing any software. However, there exists a significant difference between traditional IDEs and modern IDEs. This paper highlights the difference between traditional and modern IDEs and the relevance of Cloud-based IDEs in the present and future.

### \*Corresponding author

Abhishek Shukla, USA.

Received: January 04, 2024; Accepted: January 13, 2024; Published: January 22, 2024

**Keywords:** IDEs, Modern IDEs, Cloud-Based IDEs, Software Development

### Introduction

Integrated development environments (IDEs) have played a dynamic role in software development for many years now [1]. Integrated development environments (IDEs) offer different tools to perform different tasks considering development, including continuous compilation, automated testing, integrated debugging, and code refactoring to increase productivity [2]. Integrated development environments (IDEs) act as a toolkit specifically designed to facilitate the whole process of designing and developing of any software. However, a significant difference exists between the traditional IDEs and modern IDEs. Concerning the use and adoption of traditional IDEs, developers must work on a local machine to use all the tools that are needed for development because developing a software is an extensive and difficult process that demands time and significant effort.

In addition to the development of high-quality software, there exists the pressure of completing a project within the given timeline and budget, which can be difficult without the integration of IDEs [3]. For creating and developing a software, it demands a detailed set of developmental tools, including a code editor, a profiler, and a debugger, and all of these tools are offered by integrated development environments (IDEs). Over the years, some of the most popular programming languages, including Java or C++, have significantly evolved, and they have a quite complex syntax. Therefore, it demands considerable effort in order to develop a high-quality compiler. In the traditional context, integrated development environments do not support multiple programming languages via plugins. However, this feature is available in modern integrated development environments, where these plugins can be tightly coupled with integrated development environments. This highlight starts if m IDE supports n different languages, then it must be  $m * n$  to get the result. However, with that advancement, there are new versions of the language and updated versions

of Java and C++ for worldwide use of integrated development environments, including Eclipse CDT. With that, the updating process of compilers has become a significant, unbearable effort. Modern research works are focused on bringing a change in the traditional approach of plugins with a language server approach, which can help in reducing the process of implementations of different numbers [4].

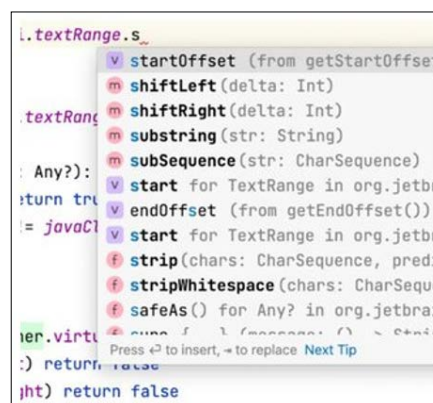


Figure 1: An Example of Coding using IDE

Not only this, as per research, traditional IDEs are restricted with platforms as some IDEs are compatible with Windows, and some only work with Linux and Mac iOS. The restriction of platforms tends to give rise to many problems for developers for developing software, specifically when it comes to collaboration between two operating systems. Some of the traditional IDEs are also resource intensive, which means they demand a substantial memory, and if a developer has lower-end hardware, then it can cause many problems. Traditionally, IDEs have also met criticism concerning handling larger databases [5]. These limitations highlight that there is a need for flexible software-integrated IDEs that are also focused on collaboration. In the light of these limitations, the modern IDEs are cloud-based IDEs that specifically address all these limitations

to offer an effective solution for all the developers.

### Literature Review

In the educational sector, traditional programming is focused on programming environments that are set on lab machines, which is a hectic process concerning the speed of developmental tools. On the other hand, the web is an appealing vision because moving integrated development environments to the web is not only about porting desktops but also a whole process of structured IDE to understand the full potential that the combinations of IDEs and web can utilize.

Over the years, programming has evolved significantly with the advent of new technologies and innovations. programming and integrated development environments are significantly important as they allow the creation of software, mobile apps, websites, and other digital products that have made lives easier and faster. With regard to integrated development environments, it is important to understand the types of programming languages, which are broadly divided into three groups, including low-level languages, high-level languages, and scripting languages. Low-level languages are closely related to machine learning, and they are basically used for writing operating systems, firmware, and device drivers such as assembly language, C, and C++. On the other hand, high-level languages are quite easy to learn, and they are mostly adopted to write games, applications, and websites, which include Java, Python, and Ruby. Scripting languages are mostly used for the automation of repetitive tasks such as web development and administration of systems [6].

In today's world, software development is an important activity. However, programming has significantly evolved with the evolution of computers. With the evolution, some of the tasks that include editing, automation, and debugging have become more difficult, and the ultimate solution for these existing problems is integrated development environments. Integrated development environments are known as lifesavers for programmers. However, there are some disadvantages associated with them as IDEs are installed on the system, and one has to be restricted to the computer for using all

the features of IDE to develop software. As mentioned above, the integrated development environments demand language-specific integrated environments such as Visual Studio so that they can be downloaded and configured. If a programmer is working on another machine, the whole system and development kit must be installed on the new system, which is a hectic process. In that regard, cloud-based integrated development environments offer significant solutions. Cloud-based integrated development environments are flexible, and a programmer can have access to the project on the web. The only requirement is to connect with the web IDE. Programmers can even write programs on mobile phones with Android applications with web cloud-based IDE. However, there are some compilers that are not compatible with Android, and the solution is being offered by cloud-based IDE [7]. The onset of cloud-based integrated development environments has changed software development dynamics and made it much easier than before. Programmers rely on cloud-based integrated development environments instead of traditional integrated development environments.

### Future of Cloud-Based Integrated Development Environments (IDEs)

The advancement in technology has made features of cloud-based lightweight modern integrated development environments quite transformative, as it aligns with the changing needs of the software industry. With that, the future is promising as cloud-based IDEs seamlessly run on browsers that do not require an open package installation. This feature has allowed customizable workspaces, which can also optimize collaboration. In the present era, the remote work has become the new normal, and web developers all across the world aim to access their code and integrated development environments from anywhere in the world without restriction of time. Traditional and local IDEs are prone to security breaches and data loss. On the other hand, cloud IDEs are specifically designed while keeping these vulnerabilities in mind, and cloud IDEs have inbuilt backups and encryption along with other security features. These features have improved the quality of collaboration among developers to work on a database without any interference simultaneously [8].

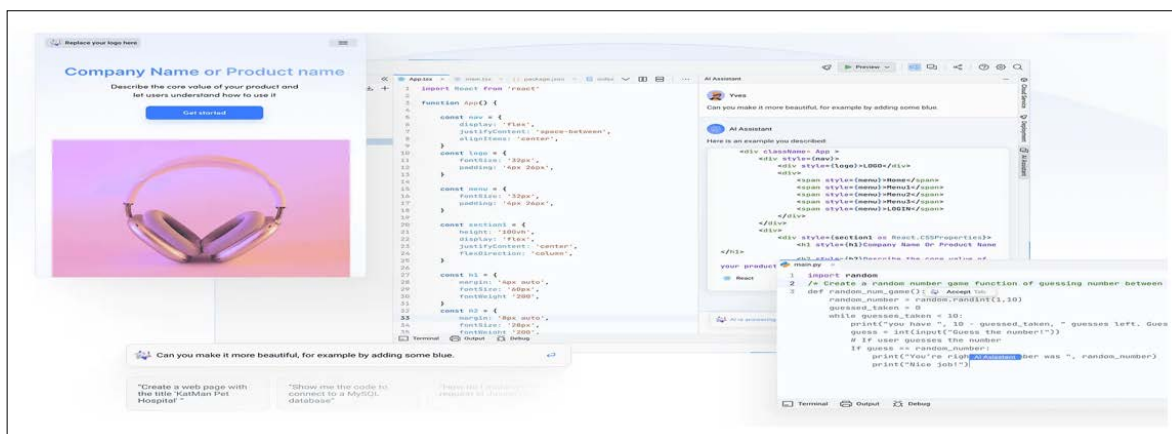


Figure 2: Benefits of Cloud-based IDEs

Cloud-based IDEs are beneficial from an economist's perspective as well as a significantly helping reduce the cost that is being spent on maintaining and developing the infrastructure. The developer or business is responsible for paying only for the resources used to maintain cloud-based IDEs. Given this, it doesn't come as a surprise that the cloud-based IDE market is witnessing significant growth. As per data from market research future, the global cloud IDE market is predicted to grow at a CAGR of 17% from 2018 to 23.

In addition, the onset of COVID-19 boosted the concept of remote working, allowing more and more organizations to move towards cloud-based development tools rather than traditional ones.

Many organizations all across the world have significantly benefited by moving from traditional development environments to cloud-based integrated development environments. One of those organizations includes Autodesk, which is considered a leader in 3D design, engineering, and software related to entertainment. For enhancing the productivity and reducing the cost and setup time, Autodesk adopted the services of an AWS' Cloud 9 IDE for their developers. In the same way, Red Hat employed Eclipse Che, also a cloud IDE, to create a scalable and productive development environment for all its developers. This transition significantly reduced to development time and enhanced the speed of innovation [9].

The rise of cloud-based integrated development environments has changed the dynamics of the technological world as almost all small and big organizations are adopting the services for benefits. The demand for cloud IDE is quite high as with remote work, more and more companies are adopting the cloud development environments. This trend is more likely to continue because of the need for collaborative development environments. In addition to this, the integration of cloud IDEs with DevOps has become quite common as it offers a seamless environment for writing codes, testing, and deployment, which significantly increases the software delivery process. The previous versions of cloud IDEs were focused on specific programming languages. Still, now they have become language-agnostic platforms that allow developers to work with different languages within the same environment without changing interfaces, thus increasing flexibility and productivity. The modern cloud IDEs can also be customized per the developers' needs and environment, and project requirements. Artificial intelligence and machine learning are also having an impact on the evolution of cloud IDEs as they can enhance the different aspects of the software development process. Artificial intelligence and machine learning in cloud IDE can be employed for the automation of routine coding tasks by relying on the previous task and coding patterns, and AI can also focus on generating a code, which can increase the productivity of the developer [10].

The integration of AI and ML can also help in the personalization of coding and testing, along with highlighting specific aspects and setting the layout.

## Conclusion

To sum up, based on the discussion, the future of a cloud-based integrated development environment is quite productive and efficient. With the advancement of AI and ML, more and more people are adopting cloud-based IDEs to enhance productivity and to cut down the maintenance cost of the system.

## References

1. Snipes W, Murphy-Hill E, Fritz T, Vakilian M, Damevski K, et al. (2015) Chapter 5 - A Practical Guide to Analyzing IDE Usage Data. ScienceDirect 85-138.
2. Bibaev V, Kalina A, Lomshakov V, Golubev Y, Bezzubov A, et al. (2022) All you need is logs: improving code completion by learning from anonymous IDE usage logs. Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering 1269-1279.
3. Brown AW, Wallnau KC, Feiler PH (1993) Understanding integration in a software development environment: Issues and illustrations. Journal of Systems Integration 3: 303-329.
4. Chen Y, Piao K, Ezzati-jivan N, Dagenais MR (2021) Distributed Architecture for an Integrated Development Environment, Large Trace Analysis, and Visualization Sensors 21: 5560.
5. Kline RB, Seffah A (2005) Evaluation of integrated software development environments: Challenges and results from three empirical studies. International Journal of Human-Computer Studies 63: 607-627.
6. Mohindroo SK (2023) The Evolution of Programming Languages: Past, Present, and Future [www.linkedin.com](http://www.linkedin.com).
7. Manjunath R, Arunjith C (2018) Cloud based Integrated Development Environment for Android Devices. International Journal of Engineering Research & Technology 3: 1-5.
8. Zakieh AS, Enrique GG, Hadi G, Sara RG, Jaume J, et al. (2021) Modern Integrated Development Environment (IDEs). Sustainable Smart Cities and Territories 274-288.
9. Teamcode (2023) Discussing Cloud Development Environment and its Future. Medium <https://medium.com/@teamcode2023/discussing-cloud-development-environment-and-its-future-df2f78f8f1c2>.
10. Luo L, Schäf M, Sanchez D, Bodden E (2023) IDE Support for Cloud-Based Static Analyses ACM Reference Format 21: 23-28.

**Copyright:** ©2024 Abhishek Shukla. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.