# Frontend

## **Definition**

The frontend covers everything the user of a software or website can see, touch, and experience. On websites, the frontend includes content (posts, pages, media, and comments), design, and navigation. Another example would be the frontend of database software, where users can enter and display data.

## Leading Mobile and Web Front-End Development Technologies

### 1. React for a Responsive Web Front-End Development

React is an open-source front-end library licensed and released under MIT in 2013. React is probably the most used front-end technology and has greatly improved with every update.

The library is maintained and supported by the social media giant, Facebook.

React’s code is split into components to facilitate code reusability and fast debugging. The applications built on React are highly responsive and amounts for SEO friendliness. The data binding also ensures a solid code structure and a seamless app performance.

Airbnb, Dropbox, BBC, Facebook, New York Times, and Reddit are some of the prominent websites and web apps built with React.

### 2. Angular Front-End Development

AngularJS is a JavaScript front-end development framework originally developed in 2009. It was licensed under MIT as open-source and since that time, AngularJS has gained immense popularity for its advantages.

Angular is maintained by the search engine powerhouse, Google, and supported by a huge community of developers.

Angular uses MVVM architecture making it easy to maintain and refactor the codes. The component-based architecture further ensures a higher quality of code and eases testing.

The JavaScript front-end framework is consistent, readable, and its apps are performing well in various industries.

Netflix, Upwork, IBM, Goodfilms, and Freelancer are some of the renowned websites built with Angular. Gmail, Paypal, and The Guardian are some of the well-known applications built with Angular.

### Bootstrap CSS Framework

Bootstrap was released on 19th August 2011 under MIT license. It is an open-source CSS framework to build dynamic websites and web applications. It is built by Mark Otto at Twitter as a framework for consistency.

Bootstrap, as a front-end development framework, allows websites to be highly responsive with its fluid grid layout and gives a mobile-ready platform perfect for all the devices.

Bootstrap ensures front-end development consistency for developers and designers. The CSS framework has also got the support of a strong community.

### React Native Cross-Platform Mobile Front-End Development

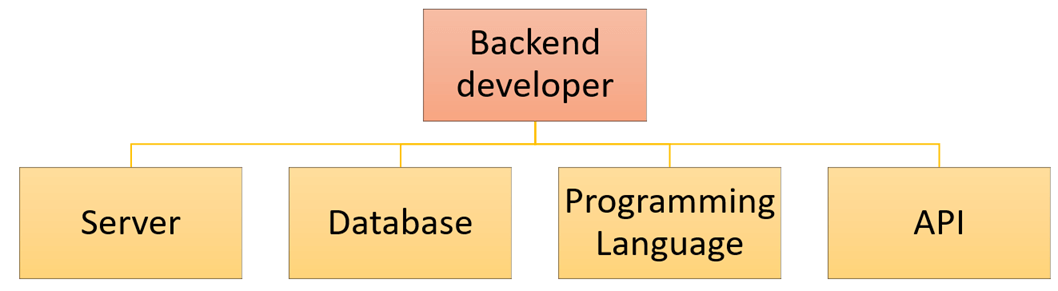
React Native was announced in February 2015 and was released as open-source under MIT License.

The front-end development framework was created by the social media giant, Facebook. It is the optimal cross-platform development framework delivering a superior app performance.

The component-driven front-end technology makes it easy for the developers with code reusability and testability Hot reloading further increases the speed of development by reducing the waiting time to view the changes.

React Native also has a huge community support and is updated regularly to add more features.

## **What is Backend Development?**

**Back-end Development** refers to the server-side development. It focuses on databases, scripting, website architecture. It contains behind-the-scene activities that occur when performing any action on a website. It can be an account login or making a purchase from an online store. Code written by back-end developers helps browsers to communicate with database information.

### **Web Development Languages:**

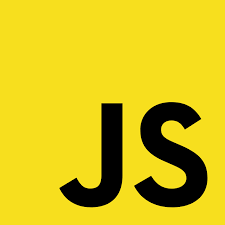
Backend engineer should know at least one server-side or Backend programming language like Java, Python, Ruby, . Net etc.

## **What is a Backend?**

[**Backend**](https://en.wikipedia.org/wiki/Front_end_and_back_end) is a technical term that is used by computer professionals and coders. Basically, the backend is the programmer’s code that deals with server-side operations, including CRUD functions with database and all server logic. The majority of operating syntax and data is backlogged and obtained in the backend of any application. Users can’t access the backend of any computer application.

Top Backend Technologies

## **JavaScript**

****

For the last eight years, [**JavaScript**](https://www.javascript.com/) has consecutively won the first position as the most commonly used programming technology in Stack [**Overflow Annual Surveys**](https://insights.stackoverflow.com/survey/). The 2020 Developer Survey of Stack Overflow also presents that 69.7% of professional developers (47,184 responders participated) voted in favor of JavaScript as the most popular technology.

With the assistance of Node.js, you can execute server-side operations under JavaScript programming technology. But it is also essential to understand that Node.js works as a platform and mostly uses Express.js as its server-side framework. In simple words, Express.js is an open-source backend framework for Node.js.

Express.js in conjunction with Node.js uses JavaScript as both front and backend language. It also develops application programming interfaces (APIs) for web, mobile, hybrid, single and multi-page applications.

### **JavaScript Features**

**Speedy Development:** JavaScript allows developers to use it as both front and backend technology. That’s why it is quick and easier to produce both mobile and web applications with this scripting language.

**Less Overhead:** This backend technology helps with its built-in features for DOM access and hoop etc. This leads to fewer scripting overheads and improves performance.

**Minimalist Backend Technology:** JavaScript’s backend platform like Express.js offers Express middleware modules to figure out different development challenges. These middleware packages consist of HTTP request logger, error handling, security headers and POST data functions. Moreover, JavaScript also an undoctrinaire framework which works without stringent regulations. Developers can use middleware of their choice without any restriction.

**Reduce Expenses:** JavaScript backend scripting performs in a very efficient manner. The same language is used for producing software’s front and server-side coding. That’s why it becomes easy to handle in terms of effort and budget.

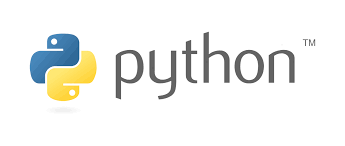
**Open-Source Community:** The scope of improvement becomes lesser without community contribution and feedback. Luckily, JavaScript is an open-source community and backend engineers always have the chance to be reviewed. Such reviews also lead to an improvement in your backend coding.

**I/O Request Handling:** Express.js in conjunction with Node.js, has enough power to handle thousands of notifications and I/O requests from application users.

### **JavaScript Limitations**

* It isn’t easy to understand the event-driven nature of JavaScript backends. Especially developers who work with other coding languages misread the callbacks and draft entire code in callbacks. However, this problem is rare in the latest versions.
* Programmers usually misinterpret the concept of middleware while using JavaScript server-side programming.
* It is also hard to host backend frameworks of JavaScript with MySQL databases.
* Few backend engineers also complain against the unopinionated nature of JavaScript backend technologies. According to them, standard designing is much better instead of this freedom.

## **Python**

****

[**Python**](https://www.python.org/) is a leading all-purpose programming language created by a Dutch programmer Guido van Rossum in 1991. With core competency of concise and readable codes, Python supports backend programmers to write rational and explicit scripts.

According to the 2020 Developer Survey of Stack Overflow, it is the most wanted back-end technology that developers want to learn. Yes, it is the fourth year in a row where developers want to learn Python as a priority. It maintains the third position among the most loved programming technologies as well.

### **Python Features**

**Easy to Understand & Code:** The core competency of Python is its readability feature. Actually, Python’s precise scripting makes it easy to understand and code for both newbies and programmers. Sometimes, it also seems like the English sentences.

**Extensive Libraries:** The backend programming technology downloads extensive libraries. These libraries include codes for unit-testing, databases, regular expressions, browsing, and email, etc. In short, with the support of extensive libraries, backend engineers don’t have to write each code manually.

**Affordable:** Python is open-source and free. That is why developers and companies have the chance to use free extensive libraries and other sources for their projects.

**IoT Opportunities:** With the help of Python’s modern programming features, you can build physical projects on Raspberry Pi.

**Embeddable:** It is easy to insert your Python code in the source code of languages like C++. However, the coder shouldn’t forget the Write Once Run Anywhere (WORA) rule here.

### **Python Limitations**

* Any interruption in Python’s coding can lead to slow execution. It can badly affect the speed of the project as well.
* As compared to other backend technologies, the database access layers of Python are not fully developed.
* Python also needs additional testing to detect bugs and other errors.
* It depends on third-party libraries and frameworks.

## **Ruby**

****

[**Ruby**](https://en.wikipedia.org/wiki/Ruby_(programming_language)) is another open-source backend technology that was designed by Japanese computer scientist Yukihiro Matsumoto in the 1990s. This scripting language has similarities with Python, Java, and Perl. Ruby is usually used for the development of web applications and is considered the favorite for prototyping.

Tech businesses like Esty, Airbnb, and Shopify are using Ruby as backend scripting technology due to its object-oriented, scalability, and flexible programming features.

### **Ruby Features**

**Productive:** Due to short readable code and 3rd party libraries’ availability, Ruby is a very product backend technology. Developers typically need fewer separate documentation with Ruby. It allows backend technologies to use already available projects.

**Metaprogramming:** Ruby also backs metaprogramming. In metaprogramming, the developer scripts code in Ruby, and the program presents the other code itself. With flexible tools like ghost methods, introspection, and macros, it is convenient to script in a short duration.

**Libraries:** Ruby offers 3rd party modules. You can find all of its libraries at RubyGems.

**Testing:** Ruby backend technology is also worth-using for the development of high-quality projects. These are testing automation features and valuable libraries of Ruby that make it happen.

**Fast & Stable:** Compared to other backend scripting languages, Ruby takes 40% less time in developing a code. It is also quick for making requests and further processing steps. In the same token, Ruby is also more resilient due to its object-oriented characteristics.

### **Ruby Limitations**

* The runtime speed of Ruby is slow as compared to other scripting languages.
* Although Ruby is an open-source community but unluckily, it doesn’t have enough libraries and sources.
* As compared to other programming languages, Ruby is new and functions with its own code. It is hard for other backend technologists to learn it.
* It is challenging to debug a code written in Ruby.

## **PHP**

****

[**PHP**](https://en.wikipedia.org/wiki/PHP) is a leading server-side scripting technology that was introduced by Rasmus Lerdorf in 1994. This open-source backend technology is commonly used for websites. Around 79.1% of websites on the internet use PHP as server-side technology, according to a recent survey of W3Techs.

This general-purpose scripting language is easy to use and emend information in the databases. The availability of several modern frameworks, massive community, robust codebase, and easy deployment also adds tremendous value to this technology. It is better to use PHP together with MySQL and Linux Apache.

### **PHP Features**

**Open Source & Platform Independent:** A variety of free PHP libraries are available on the web. Coders can use them for fast backend development. Though operating systems like Windows, Linux, etc., primarily support PHP, but you can also run PHP web applications on any platform.

**Cost-Effective:** With the presence of free and open-source communities, it is quite convenient to code with PHP. That is why it is inexpensive to hire a PHP backend developer. It also has the second last position among the top paying technologies in the United States as per the 2020 Stack Overflow Developer Survey.

**Simplifies Programming:** Instead of using lengthy SQL code, PHP frameworks allow developers to get the benefit of providing object-relational mapping (ORM) systems. Similarly, the use of model-view-controller (MVC) architecture also makes it easy to code server-side functions for developers.

**Best for Beginners:** PHP is suitable to learn for first-time backend engineers. Features like running sequence and no learning curve make PHP beginners’ choice to learn.

**Automate the Development Tasks:** With PHP scripting technology, it is easy to automate development tasks like session management, URL mapping, authentication, etc.

**Security Against Target Attacks:** No doubt, developers take PHP as an unsecured backend language. But with built-in security functions, you can avoid these threats.

### **PHP Limitations**

* The popularity of PHP is declining day-by-day. Developers rarely consider to include PHP in their skillset nowadays.
* PHP is not competing with modern backend technologies like Python and Ruby due to a lack of advanced libraries.
* Due to the open-source nature of PHP, it is accessible to misuse bugs in the coding.

## **Java**

****

[**Java**](https://www.java.com/en/) is one of the most powerful backend technologies, which has the second rank, according to TIOBE Index 2021. James Gosling originally developed this programming technology in 1991, but it was published in 1995 by Sun Microsystems.

Developers prefer to make feature-rich and adaptable web applications with Java for years. Howbeit, you can use Java for mobile devices, severe, and microcontroller software development as well.

### **Java Features**

**Simple & Highly Scalable:** Java EE is highly scalable because it permits numerous instances to server requests. Instant availability of Java components and unambiguous syntax technology also makes it simple for developers to use this backend language.

**Multi-Threading:** Java allows to handle all requests in independent threads due to a multi-threaded web server. With this multi-threading feature, Java functions very well for CPU-intensive applications.

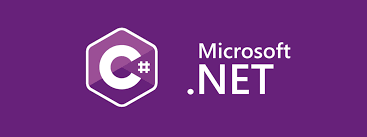
**Open-Source Libraries:** A variety of open-source libraries such JSON parsing, messaging, PDF, unit testing and excel libraries available for Java. Java developers can use these resources to expedite their server-side tasks.

**Security:** Java offers terrific features to overcome security risks. Likewise, Java Virtual Machines examine java bytecodes to detect and reduce the risk of viruses. Similarly, the security model of Java and testing of reusable codes help developers in avoiding security threats.

### **Java Limitations**

* This server-side programming consumes more time and memory.
* Java doesn’t offer command over garbage collection, and low-level programming support is also missing in Java.
* Due to high hardware cost, it may be expensive to use Java.
* Swing toolkit, which Java is using GUI applications is different as compared to trendy ones.

## **C# .NET**

****

[**C#**](https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/) known as C-sharp is one of the most famous backend programming languages preferred for automation in the Windows environment. C# is also used for web development in the ASP.net framework. It is one of the oldest programming languages and an extension to C++.

This backend technology is most commonly used for desktop applications and embedded systems in this era. The execution speed of C# is faster than most of the other programming languages, i.e., Python. In the modern world, C# is most widely used in game development using Unity like platforms. Backend engineers also use this backend technology to develop console applications.

### **C# Features**

**Object-Oriented Language:** C# is an object-oriented language that means you can structure your code using classes and relationships. It is helpful to implement the system with easy troubleshooting if something goes wrong.

**Cross-Platform:** The applications that use the C# backend can operate in different operating systems such as Windows, macOS, etc. Please read this article about [**cross platform frameworks**](https://blog.back4app.com/cross-platform-app-development-frameworks/) to know more.

**Compatibility:** C# applications allow interoperability with older legacy systems. This backend scripting technology is also useful for organizations that are not updated and using old programming frameworks.

**Automatic Garbage Collection:** C# has the ability to remove all the garbage on the system. This feature is beneficial for the quick execution of the program. This garbage collection does not let the system hang during the execution of the application.

### **C# Limitations**

* C# is a high-level language, so it does not let the programmer interact with the hardware directly.
* This backend technology is less flexible as compared to other scripting languages. Yes, it only runs in the .NET framework and can only be hosted on the Windows platform.

## **Perl**

****

[**Perl**](https://www.perl.org/) is another general-purpose backend technology that Larry Wall developed 32 years ago. According to the 2020 Stack Overflow Developer Survey, Perl is the topmost paying technology globally. By the way, it is significant to know that the same survey also ranks Perl among the first three dreaded languages.

Whereas Perl is pretty old-fashioned, but developers are still using Perl 5 for quick automation and prototyping.

### **Perl Features**

**Multi-Platform:** Perl can operate with distinct platforms, including Macintosh, Windows, and Most UNIX variants.

**Open-Source:** Like the other rival backend technologies, Perl also offers open-source and free libraries. However, in many cases, they could again ask for small copying charges.

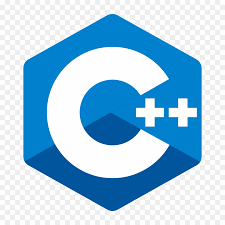
**Embeddable & Extensible:** Perl can easily embed in any of C++ and C applications. Though SWIG and XS, Perl also backs external libraries of C and C++.

**Text Processing:** Perl is a perfect scripting language for text processing. Luckily, Perl’s recent versions also maintain POSIX-compliant systems and process socket calls with other advanced features.

### **Perl Limitations**

* It is difficult to fix bugs and separate infectious data with Perl.
* Perl doesn’t provide high-performance processing as compared to other backend technologies.
* Perl libraries are not advanced enough.
* It agitates when code is extensive.
* Perl is not the right option if you want to improve the scalability and speed of your project.
* It is expensive to hire Perl backend engineers because it is slightly outdated, and developers don’t prefer to learn it.

## **C++**

****

[**C++**](https://en.wikipedia.org/wiki/C%2B%2B) is an extended version of C language. C++ was introduced with classes. This concept of classes or object-oriented programming was missing in C language. The idea of object-oriented programming is vital for any programming language in the modern world to write structured code using classes and by defining their relationships.

C++ is one of the oldest programming languages and is mostly used in system programming & embedded systems. C++ is a low-level language, and it can interact with the hardware resources. It is used for gaming applications, operating systems, database software, etc.

### **C++ Features**

**Portability:** Platform independence is one of the critical features of C++ applications. You can execute C++ applications on different platforms or operating systems.

**Object-Oriented Programming:** C language does not support object-oriented programming, but C++ fulfills this deficiency. This feature makes this language more powerful and easy to code with structured programming.

**Memory Management:** Programmers get complete control over memory management using C++. It can help the programmer to manage memory effectively to execute the program.

**Low-Level Language:** C++ is a low-level language and close to the system. Therefore, most of the embedded systems are built in C++ because it can directly interact with the hardware resources.

### **C++ Limitations**

* There are some security issues in C++. With C++ backend technology, users can directly interact with the hardware using C++ as a low language.
* There is no garbage collection in C++ to filter out unnecessary data.

## **Kotlin**

****

[**Kotlin**](https://kotlinlang.org/) is a backend programming language that is being used for the development of android applications. It is taking over Java for the development of android applications, and its demand is increasing day-by-day. Over 60% of android app developers are using Kotlin at the backend. Kotlin interoperates completely with Java & JVM.

JetBrains introduced Kotlin as a backend programming language for Android applications in 2011. Since then, it has become one of the most famous programming languages in the world of computer science.

### **Kotlin Features**

**Concise Code:** Programmers can solve more significant problems by writing fewer code lines than in other languages.

**Easy to Maintain:** Due to concise code, Kotlin helps the programmers easily read and maintain the code.

**Interoperable with Java:** This programming technology is completely compatible with Java. Programmers do not need to change the whole project to switch to Kotlin. With all Java tools & technologies, you can add it to your Kotlin project to implement additional functionalities.

### **Kotlin Limitations**

* Kotlin has no primitive types for variables & functions. Programmers are used to working with primitive types in Java & other languages.
* It is challenging to manage variables & functions output with the absence of primitive types.
* Java is faster in compilation than Kotlin when it comes to clean builds of android applications.

## **Scala**

****

[**Scala**](https://www.scala-lang.org/) is a high-level language that combines object-oriented & functional programming to make it more concise. You can build high-performance systems accessing vast libraries of its JVM & JavaScript runtimes. Scala is interoperable with Java because it runs on JVM and can mix both languages’ code to give an ultimate solution.

The Scala compiler is helpful for static types. You do not need to work for types as the system will work for you about static types. It makes this backend programming language more helpful & useful.

### **Scala Features**

**Concise:** This backend programming language combines object-oriented & functional programming. That makes the code more concise & more comfortable to maintain.

**Interoperable with Java:** Scala runs on JVM, and it is entirely compatible with Java programming language. The code can be written in Java & Scala at the same time to get the desired results from the system. This seamless integration with Java is beneficial for Java developers to use this backend technology.

**Easy to Maintain:** The concise Scala code can easily maintain & programmers can read the code easily. It helps the programmers to understand the code quickly, significantly when a project is shifted from one programmer to another.

### **Scala Limitations**

* Scala presents an entirely different paradigm and different from traditional Java programming techniques.
* It is a bit harder for the programmers to understand the type of information as it is not done at the developer end. It is a combination of functional & object-oriented programming.

## **Conclusion**

This guide shared one of the leading backend technologies with readers. We considered TIOBE Index 2021, Stack Overflow Developer Survey 2020, and web technology survey of W3Techs to fetch statistical data regarding server-side technologies. Hopefully, the top backend technologies we explained above would help you in making the right decision.

I hope you liked this article! If so, you may also be interested to know more about the best backend frameworks. To know more, please read the article [**Top 10 Backend Frameworks**](https://blog.back4app.com/backend-frameworks/).

## **FAQ**

## **What is a backend?**

The backend is the programmer’s code that deals with server-side operations, including CRUD functions with database and all server logic.

## **What are the best backend technologies?**

– JavaScript

– Python

– Ruby

– PHP

– Java

– C#

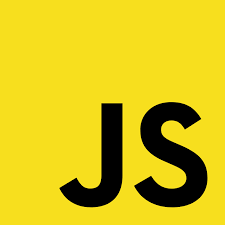
– C++

– Perl

– Scala

– Kotlin

## **JavaScript**

****

For the last eight years, [**JavaScript**](https://www.javascript.com/) has consecutively won the first position as the most commonly used programming technology in Stack [**Overflow Annual Surveys**](https://insights.stackoverflow.com/survey/). The 2020 Developer Survey of Stack Overflow also presents that 69.7% of professional developers (47,184 responders participated) voted in favor of JavaScript as the most popular technology.

With the assistance of Node.js, you can execute server-side operations under JavaScript programming technology. But it is also essential to understand that Node.js works as a platform and mostly uses Express.js as its server-side framework. In simple words, Express.js is an open-source backend framework for Node.js.

Express.js in conjunction with Node.js uses JavaScript as both front and backend language. It also develops application programming interfaces (APIs) for web, mobile, hybrid, single and multi-page applications.

### **JavaScript Features**

**Speedy Development:** JavaScript allows developers to use it as both front and backend technology. That’s why it is quick and easier to produce both mobile and web applications with this scripting language.

**Less Overhead:** This backend technology helps with its built-in features for DOM access and hoop etc. This leads to fewer scripting overheads and improves performance.

**Minimalist Backend Technology:** JavaScript’s backend platform like Express.js offers Express middleware modules to figure out different development challenges. These middleware packages consist of HTTP request logger, error handling, security headers and POST data functions. Moreover, JavaScript also an undoctrinaire framework which works without stringent regulations. Developers can use middleware of their choice without any restriction.

**Reduce Expenses:** JavaScript backend scripting performs in a very efficient manner. The same language is used for producing software’s front and server-side coding. That’s why it becomes easy to handle in terms of effort and budget.

**Open-Source Community:** The scope of improvement becomes lesser without community contribution and feedback. Luckily, JavaScript is an open-source community and backend engineers always have the chance to be reviewed. Such reviews also lead to an improvement in your backend coding.

**I/O Request Handling:** Express.js in conjunction with Node.js, has enough power to handle thousands of notifications and I/O requests from application users.

### **JavaScript Limitations**

* It isn’t easy to understand the event-driven nature of JavaScript backends. Especially developers who work with other coding languages misread the callbacks and draft entire code in callbacks. However, this problem is rare in the latest versions.
* Programmers usually misinterpret the concept of middleware while using JavaScript server-side programming.
* It is also hard to host backend frameworks of JavaScript with MySQL databases.
* Few backend engineers also complain against the unopinionated nature of JavaScript backend technologies. According to them, standard designing is much better instead of this freedom.

## **Python**

****

[**Python**](https://www.python.org/) is a leading all-purpose programming language created by a Dutch programmer Guido van Rossum in 1991. With core competency of concise and readable codes, Python supports backend programmers to write rational and explicit scripts.

According to the 2020 Developer Survey of Stack Overflow, it is the most wanted back-end technology that developers want to learn. Yes, it is the fourth year in a row where developers want to learn Python as a priority. It maintains the third position among the most loved programming technologies as well.

### **Python Features**

**Easy to Understand & Code:** The core competency of Python is its readability feature. Actually, Python’s precise scripting makes it easy to understand and code for both newbies and programmers. Sometimes, it also seems like the English sentences.

**Extensive Libraries:** The backend programming technology downloads extensive libraries. These libraries include codes for unit-testing, databases, regular expressions, browsing, and email, etc. In short, with the support of extensive libraries, backend engineers don’t have to write each code manually.

**Affordable:** Python is open-source and free. That is why developers and companies have the chance to use free extensive libraries and other sources for their projects.

**IoT Opportunities:** With the help of Python’s modern programming features, you can build physical projects on Raspberry Pi.

**Embeddable:** It is easy to insert your Python code in the source code of languages like C++. However, the coder shouldn’t forget the Write Once Run Anywhere (WORA) rule here.

### **Python Limitations**

* Any interruption in Python’s coding can lead to slow execution. It can badly affect the speed of the project as well.
* As compared to other backend technologies, the database access layers of Python are not fully developed.
* Python also needs additional testing to detect bugs and other errors.
* It depends on third-party libraries and frameworks.

## **Ruby**

****

[**Ruby**](https://en.wikipedia.org/wiki/Ruby_(programming_language)) is another open-source backend technology that was designed by Japanese computer scientist Yukihiro Matsumoto in the 1990s. This scripting language has similarities with Python, Java, and Perl. Ruby is usually used for the development of web applications and is considered the favorite for prototyping.

Tech businesses like Esty, Airbnb, and Shopify are using Ruby as backend scripting technology due to its object-oriented, scalability, and flexible programming features.

### **Ruby Features**

**Productive:** Due to short readable code and 3rd party libraries’ availability, Ruby is a very product backend technology. Developers typically need fewer separate documentation with Ruby. It allows backend technologies to use already available projects.

**Metaprogramming:** Ruby also backs metaprogramming. In metaprogramming, the developer scripts code in Ruby, and the program presents the other code itself. With flexible tools like ghost methods, introspection, and macros, it is convenient to script in a short duration.

**Libraries:** Ruby offers 3rd party modules. You can find all of its libraries at RubyGems.

**Testing:** Ruby backend technology is also worth-using for the development of high-quality projects. These are testing automation features and valuable libraries of Ruby that make it happen.

**Fast & Stable:** Compared to other backend scripting languages, Ruby takes 40% less time in developing a code. It is also quick for making requests and further processing steps. In the same token, Ruby is also more resilient due to its object-oriented characteristics.

### **Ruby Limitations**

* The runtime speed of Ruby is slow as compared to other scripting languages.
* Although Ruby is an open-source community but unluckily, it doesn’t have enough libraries and sources.
* As compared to other programming languages, Ruby is new and functions with its own code. It is hard for other backend technologists to learn it.
* It is challenging to debug a code written in Ruby.

## **PHP**

****

[**PHP**](https://en.wikipedia.org/wiki/PHP) is a leading server-side scripting technology that was introduced by Rasmus Lerdorf in 1994. This open-source backend technology is commonly used for websites. Around 79.1% of websites on the internet use PHP as server-side technology, according to a recent survey of W3Techs.

This general-purpose scripting language is easy to use and emend information in the databases. The availability of several modern frameworks, massive community, robust codebase, and easy deployment also adds tremendous value to this technology. It is better to use PHP together with MySQL and Linux Apache.

### **PHP Features**

**Open Source & Platform Independent:** A variety of free PHP libraries are available on the web. Coders can use them for fast backend development. Though operating systems like Windows, Linux, etc., primarily support PHP, but you can also run PHP web applications on any platform.

**Cost-Effective:** With the presence of free and open-source communities, it is quite convenient to code with PHP. That is why it is inexpensive to hire a PHP backend developer. It also has the second last position among the top paying technologies in the United States as per the 2020 Stack Overflow Developer Survey.

**Simplifies Programming:** Instead of using lengthy SQL code, PHP frameworks allow developers to get the benefit of providing object-relational mapping (ORM) systems. Similarly, the use of model-view-controller (MVC) architecture also makes it easy to code server-side functions for developers.

**Best for Beginners:** PHP is suitable to learn for first-time backend engineers. Features like running sequence and no learning curve make PHP beginners’ choice to learn.

**Automate the Development Tasks:** With PHP scripting technology, it is easy to automate development tasks like session management, URL mapping, authentication, etc.

**Security Against Target Attacks:** No doubt, developers take PHP as an unsecured backend language. But with built-in security functions, you can avoid these threats.

### **PHP Limitations**

* The popularity of PHP is declining day-by-day. Developers rarely consider to include PHP in their skillset nowadays.
* PHP is not competing with modern backend technologies like Python and Ruby due to a lack of advanced libraries.
* Due to the open-source nature of PHP, it is accessible to misuse bugs in the coding.

## **Java**

****

[**Java**](https://www.java.com/en/) is one of the most powerful backend technologies, which has the second rank, according to TIOBE Index 2021. James Gosling originally developed this programming technology in 1991, but it was published in 1995 by Sun Microsystems.

Developers prefer to make feature-rich and adaptable web applications with Java for years. Howbeit, you can use Java for mobile devices, severe, and microcontroller software development as well.

### **Java Features**

**Simple & Highly Scalable:** Java EE is highly scalable because it permits numerous instances to server requests. Instant availability of Java components and unambiguous syntax technology also makes it simple for developers to use this backend language.

**Multi-Threading:** Java allows to handle all requests in independent threads due to a multi-threaded web server. With this multi-threading feature, Java functions very well for CPU-intensive applications.

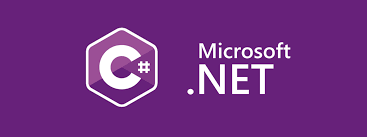
**Open-Source Libraries:** A variety of open-source libraries such JSON parsing, messaging, PDF, unit testing and excel libraries available for Java. Java developers can use these resources to expedite their server-side tasks.

**Security:** Java offers terrific features to overcome security risks. Likewise, Java Virtual Machines examine java bytecodes to detect and reduce the risk of viruses. Similarly, the security model of Java and testing of reusable codes help developers in avoiding security threats.

### **Java Limitations**

* This server-side programming consumes more time and memory.
* Java doesn’t offer command over garbage collection, and low-level programming support is also missing in Java.
* Due to high hardware cost, it may be expensive to use Java.
* Swing toolkit, which Java is using GUI applications is different as compared to trendy ones.

## **C# .NET**

****

[**C#**](https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/) known as C-sharp is one of the most famous backend programming languages preferred for automation in the Windows environment. C# is also used for web development in the ASP.net framework. It is one of the oldest programming languages and an extension to C++.

This backend technology is most commonly used for desktop applications and embedded systems in this era. The execution speed of C# is faster than most of the other programming languages, i.e., Python. In the modern world, C# is most widely used in game development using Unity like platforms. Backend engineers also use this backend technology to develop console applications.

### **C# Features**

**Object-Oriented Language:** C# is an object-oriented language that means you can structure your code using classes and relationships. It is helpful to implement the system with easy troubleshooting if something goes wrong.

**Cross-Platform:** The applications that use the C# backend can operate in different operating systems such as Windows, macOS, etc. Please read this article about [**cross platform frameworks**](https://blog.back4app.com/cross-platform-app-development-frameworks/) to know more.

**Compatibility:** C# applications allow interoperability with older legacy systems. This backend scripting technology is also useful for organizations that are not updated and using old programming frameworks.

**Automatic Garbage Collection:** C# has the ability to remove all the garbage on the system. This feature is beneficial for the quick execution of the program. This garbage collection does not let the system hang during the execution of the application.

### **C# Limitations**

* C# is a high-level language, so it does not let the programmer interact with the hardware directly.
* This backend technology is less flexible as compared to other scripting languages. Yes, it only runs in the .NET framework and can only be hosted on the Windows platform.

## **Perl**

****

[**Perl**](https://www.perl.org/) is another general-purpose backend technology that Larry Wall developed 32 years ago. According to the 2020 Stack Overflow Developer Survey, Perl is the topmost paying technology globally. By the way, it is significant to know that the same survey also ranks Perl among the first three dreaded languages.

Whereas Perl is pretty old-fashioned, but developers are still using Perl 5 for quick automation and prototyping.

### **Perl Features**

**Multi-Platform:** Perl can operate with distinct platforms, including Macintosh, Windows, and Most UNIX variants.

**Open-Source:** Like the other rival backend technologies, Perl also offers open-source and free libraries. However, in many cases, they could again ask for small copying charges.

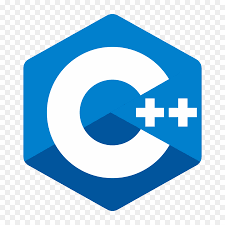
**Embeddable & Extensible:** Perl can easily embed in any of C++ and C applications. Though SWIG and XS, Perl also backs external libraries of C and C++.

**Text Processing:** Perl is a perfect scripting language for text processing. Luckily, Perl’s recent versions also maintain POSIX-compliant systems and process socket calls with other advanced features.

### **Perl Limitations**

* It is difficult to fix bugs and separate infectious data with Perl.
* Perl doesn’t provide high-performance processing as compared to other backend technologies.
* Perl libraries are not advanced enough.
* It agitates when code is extensive.
* Perl is not the right option if you want to improve the scalability and speed of your project.
* It is expensive to hire Perl backend engineers because it is slightly outdated, and developers don’t prefer to learn it.

## **C++**

****

[**C++**](https://en.wikipedia.org/wiki/C%2B%2B) is an extended version of C language. C++ was introduced with classes. This concept of classes or object-oriented programming was missing in C language. The idea of object-oriented programming is vital for any programming language in the modern world to write structured code using classes and by defining their relationships.

C++ is one of the oldest programming languages and is mostly used in system programming & embedded systems. C++ is a low-level language, and it can interact with the hardware resources. It is used for gaming applications, operating systems, database software, etc.

### **C++ Features**

**Portability:** Platform independence is one of the critical features of C++ applications. You can execute C++ applications on different platforms or operating systems.

**Object-Oriented Programming:** C language does not support object-oriented programming, but C++ fulfills this deficiency. This feature makes this language more powerful and easy to code with structured programming.

**Memory Management:** Programmers get complete control over memory management using C++. It can help the programmer to manage memory effectively to execute the program.

**Low-Level Language:** C++ is a low-level language and close to the system. Therefore, most of the embedded systems are built in C++ because it can directly interact with the hardware resources.

### **C++ Limitations**

* There are some security issues in C++. With C++ backend technology, users can directly interact with the hardware using C++ as a low language.
* There is no garbage collection in C++ to filter out unnecessary data.

## **Kotlin**

****

[**Kotlin**](https://kotlinlang.org/) is a backend programming language that is being used for the development of android applications. It is taking over Java for the development of android applications, and its demand is increasing day-by-day. Over 60% of android app developers are using Kotlin at the backend. Kotlin interoperates completely with Java & JVM.

JetBrains introduced Kotlin as a backend programming language for Android applications in 2011. Since then, it has become one of the most famous programming languages in the world of computer science.

### **Kotlin Features**

**Concise Code:** Programmers can solve more significant problems by writing fewer code lines than in other languages.

**Easy to Maintain:** Due to concise code, Kotlin helps the programmers easily read and maintain the code.

**Interoperable with Java:** This programming technology is completely compatible with Java. Programmers do not need to change the whole project to switch to Kotlin. With all Java tools & technologies, you can add it to your Kotlin project to implement additional functionalities.

### **Kotlin Limitations**

* Kotlin has no primitive types for variables & functions. Programmers are used to working with primitive types in Java & other languages.
* It is challenging to manage variables & functions output with the absence of primitive types.
* Java is faster in compilation than Kotlin when it comes to clean builds of android applications.

## **Scala**

****

[**Scala**](https://www.scala-lang.org/) is a high-level language that combines object-oriented & functional programming to make it more concise. You can build high-performance systems accessing vast libraries of its JVM & JavaScript runtimes. Scala is interoperable with Java because it runs on JVM and can mix both languages’ code to give an ultimate solution.

The Scala compiler is helpful for static types. You do not need to work for types as the system will work for you about static types. It makes this backend programming language more helpful & useful.

### **Scala Features**

**Concise:** This backend programming language combines object-oriented & functional programming. That makes the code more concise & more comfortable to maintain.

**Interoperable with Java:** Scala runs on JVM, and it is entirely compatible with Java programming language. The code can be written in Java & Scala at the same time to get the desired results from the system. This seamless integration with Java is beneficial for Java developers to use this backend technology.

**Easy to Maintain:** The concise Scala code can easily maintain & programmers can read the code easily. It helps the programmers to understand the code quickly, significantly when a project is shifted from one programmer to another.

### **Scala Limitations**

* Scala presents an entirely different paradigm and different from traditional Java programming techniques.
* It is a bit harder for the programmers to understand the type of information as it is not done at the developer end. It is a combination of functional & object-oriented programming.

## **Conclusion**

This guide shared one of the leading backend technologies with readers. We considered TIOBE Index 2021, Stack Overflow Developer Survey 2020, and web technology survey of W3Techs to fetch statistical data regarding server-side technologies. Hopefully, the top backend technologies we explained above would help you in making the right decision.

I hope you liked this article! If so, you may also be interested to know more about the best backend frameworks. To know more, please read the article [**Top 10 Backend Frameworks**](https://blog.back4app.com/backend-frameworks/).

## **FAQ**

## **What is a backend?**

The backend is the programmer’s code that deals with server-side operations, including CRUD functions with database and all server logic.

## **What are the best backend technologies?**

– JavaScript

– Python

– Ruby

– PHP

– Java

– C#

– C++

– Perl

– Scala

– Kotlin

## What is DevOps?DevOps Definition - What is DevOps - Edureka

* The term DevOps is a combination of two words namely Development and Operations. DevOps is a practice that allows a single team to manage the entire application development life cycle, that is, development, testing, deployment, and monitoring.
* The ultimate goal of DevOps is to decrease the duration of the system’s development life cycle while delivering features, fixes, and updates frequently in close synchronization with business objectives.
* DevOps is a software development approach with the help of which you can develop superior quality software quickly and with more reliability. It consists of various stages such as continuous development, continuous integration, continuous testing, continuous deployment, and continuous monitoring.

So since what is DevOps, let us have a look at the history of DevOps.

## History of DevOps

Before DevOps, We had two approaches for software development namely the Waterfall and the Agile.

### Waterfall Model

* The waterfall model is a software development model that is pretty straight forward and linear. This model follows a top-down approach.
* This model has various starting with Requirements gathering and analysis. This is the phase where you get the requirements from the client for developing an application. After this, you try to analyze these requirements.

### waterfall model - what is devops - Edureka

* The next phase is the Design phase where you prepare a blueprint of the software. Here, you think about how the software is actually going to look like.
* Once the design is ready, you move further with the Implementation phase where you begin with the coding for the application. The team of developers works together on various components of the application.
* Once you complete the application development, you test it in the Verification phase. There are various tests conducted on the application such as unit testing, integration testing, performance testing, etc.
* After all the tests on the application are completed, it is deployed onto the production servers.
* At last, comes the Maintenance phase. In this phase, the application is monitored for performance. Any issues related to the performance of the application are resolved in this phase.

#### Advantages of the Waterfall Model:

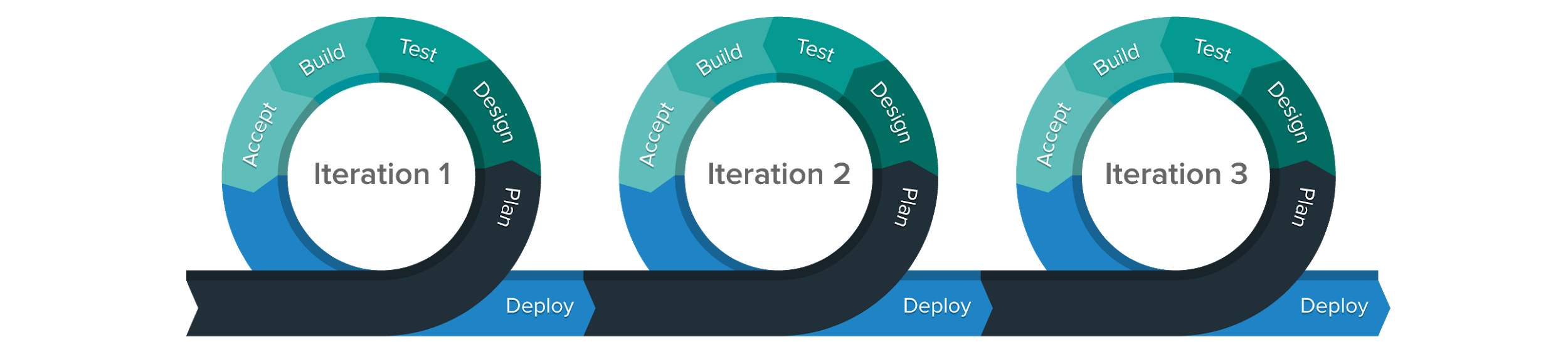
* Simple to understand and use
* Allows for easy testing and analysis
* Saves a significant amount of time and money
* Good for small projects if all requirements are clearly defined
* Allows for departmentalization & managerial control

#### Disadvantages of Waterfall Model:

* Risky and uncertain
* Lack of visibility of the current progress
* Not suitable when the requirements keep changing
* Difficult to make changes to the product when it is in the testing phase
* The end product is available only at the end of the cycle
* Not suitable for large and complex projects

### Agile Methodology

Agile Methodology is an iterative based software development approach where the software project is broken down into various iterations or sprints. Each iteration has phases like the waterfall model such as Requirements Gathering, Design, Development, Testing, and Maintenance. The duration of each iteration is generally 2-8 weeks.



#### Agile Process

* In Agile, a company releases the application with some high priority features in the first iteration.
* After its release, the end-users or the customers give you feedback about the performance of the application.
* Then you make the necessary changes into the application along with some new features and the application is again released which is the second iteration.
* You repeat this entire procedure until you achieve the desired software quality.

#### Advantages of Agile Model

* It adaptively responds to requirement changes favorably
* Fixing errors early in the development process makes this process more cost-effective
* Improves the quality of the product and makes it highly error-free
* Allows for direct communication between people involved in software project
* Highly suitable for large & long-term projects
* Minimum resource requirements & very easy to manage

#### Disadvantages of Agile Model

* Highly dependent on clear customer requirements
* Quite Difficult to predict time and effort for larger projects
* Not suitable for complex projects
* Lacks documentation efficiency
* Increased maintainability risks

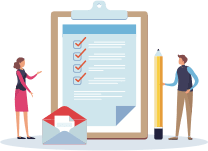
Now let us move on and discuss the DevOps stages and tools.

## DevOps Stages and Tools

As mentioned earlier, the various stages such as continuous development, continuous integration, continuous testing, continuous deployment, and continuous monitoring constitute the DevOps Life cycle. Now let us have a look at each of the stages of DevOps life cycle one by one.

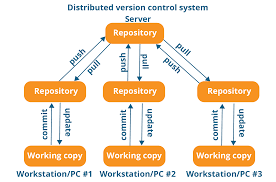
Stage – 1: Continuous Development

Tools Used: Git, SVN, Mercurial, CVS



* *Access*

Process Flow:

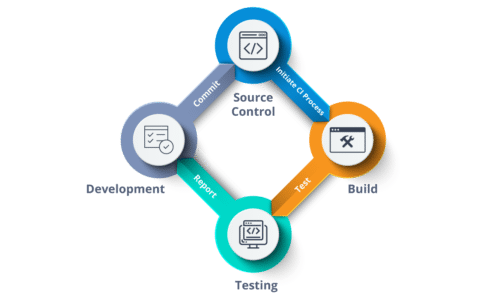


* This is the phase that involves ‘planning‘ and ‘coding‘ of the software. You decide the project vision during the planning phase and the developers begin developing the code for the application.
* There are no [*DevOps tools*](https://www.edureka.co/blog/devops-tools) that are required for planning, but there are a number of tools for maintaining the code.
* The code can be in any language, but you maintain it by using Version Control tools. This process of maintaining the code is known as Source Code Management.
* After the code is developed, then you move to the Continuous Integration phase.

Stage – 2: Continuous Integration

Tools: Jenkins, TeamCity, Travis

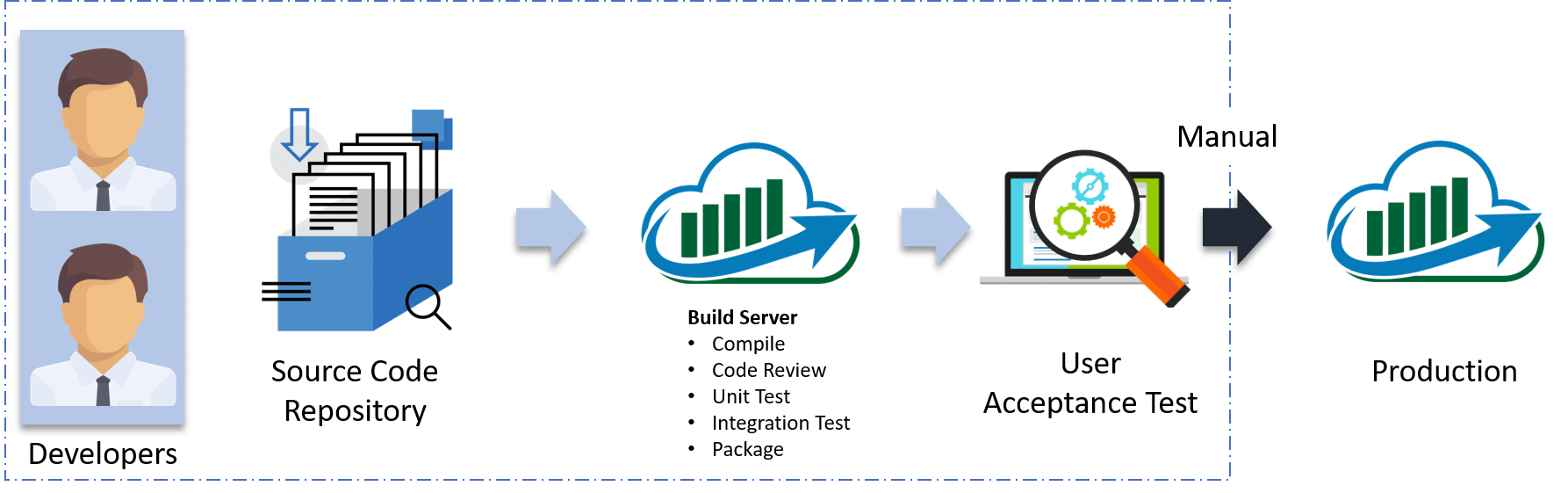
Process Flow:

* This stage is the core of the entire DevOps life cycle. It is a practice in which the developers require to commit changes to the source code more frequently. This may be either on a daily or weekly basis.
* You then build every commit and this allows early detection of problems if they are present. Building code not only involves compilation but it also includes code review, unit testing, integration testing, and packaging.  
  
* The code supporting new functionality is [*continuously integrated*](https://www.edureka.co/blog/continuous-integration/) with the existing code. Since there is a continuous development of software, you need to integrate the updated code continuously as well as smoothly with the systems to reflect changes to the end-users.
* In this stage, you use the tools for building/ packaging the code into an executable file so that you can forward it to the next phases.

Stage – 3: Continuous Testing

Tools: Jenkins, Selenium TestNG, JUnit

Process Flow:



* This is the stage where you test the developed software continuously for bugs using automation testing tools. These tools allow QAs to test multiple code-bases thoroughly in parallel to ensure that there are no flaws in the functionality. In this phase, you can use Docker Containers for simulating the test environment.
* [*Selenium*](https://selenium.dev/) is used for automation testing, and the reports are generated by [*TestNG*](https://www.edureka.co/blog/selenium-webdriver-tutorial). You can automate this entire testing phase with the help of a Continuous Integration tool called Jenkins.
* Suppose you have written a selenium code in Java to test your application. Now you can build this code using ant or maven. Once you build the code, you then test it for User Acceptance Testing (UAT). This entire process can be automated using [*Jenkins*](https://www.edureka.co/blog/jenkins-tutorial/).

Stage – 4: Continuous Deployment

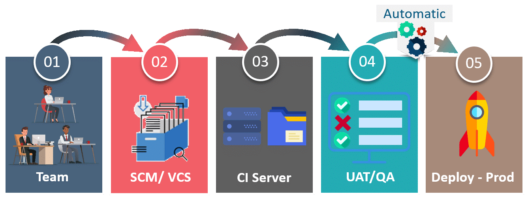
Next

Configuration Management – Chef, Puppet, Ansible

Containerization – Docker, Vagrant

Process Flow:

* This is the stage where you deploy the code on the production servers. It is also important to ensure that you correctly deploy the code on all the servers. Before moving on, let us try to understand a few things about Configuration management and [*Containerization tools*](https://www.edureka.co/blog/docker-tutorial). These set of tools here help in achieving Continuous Deployment (CD).



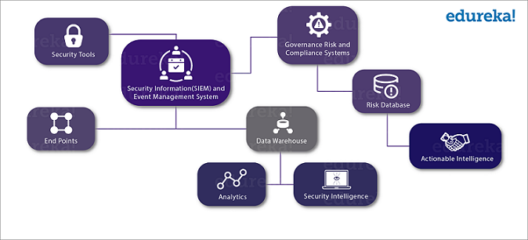
* [*Configuration Management*](https://www.edureka.co/blog/what-is-puppet/) is the act of establishing and maintaining consistency in an application’s functional requirements and performance. Let me put this in easier words, it is the act of releasing deployments to servers, scheduling updates on all servers and most importantly keeping the configurations consistent across all the servers.
* Containerization tools also play an equally crucial role in the deployment stage. The containerization tools help produce consistency across Development, Test, Staging as well as Production environments. Besides this, they also help in scaling-up and scaling-down of instances swiftly.

Stage – 5: Continuous Monitoring

Tools Used: Splunk, ELK Stack, Nagios, New Relic

Process Flow:

* This is a very critical stage of the DevOps life cycle where you continuously monitor the performance of your application. Here you record vital information about the use of the software. You then process this information to check the proper functionality of the application. You resolve system errors such as low memory, server not reachable, etc in this phase.



* This practice involves the participation of the Operations team who will monitor the user activity for bugs or any improper behavior of the system. The Continuous Monitoring tools help you monitor the application’s performance and the servers closely and also enable you to check the health of the system proactively.

## Lastly, we will discuss who exactly is a DevOps Engineer.

## Who is a DevOps Engineer?

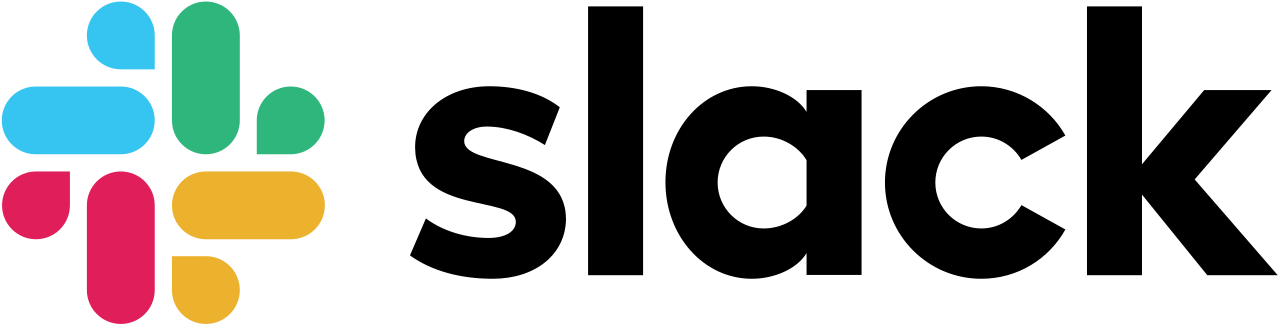
DevOps Engineer is somebody who understands the Software Development Lifecycle and has the outright understanding of various automation tools for developing digital pipelines (CI/ CD pipelines).

DevOps Engineer works with developers and the IT staff to oversee the code releases. They are either developers who get interested in deployment and network operations or sysadmins who have a passion for scripting and coding and move into the development side where they can improve the planning of test and deployment.

## Top 10 DevOps Tools

As technology advances, numerous DevOps tools have been developed to make collaboration and development easier. To help you refine your DevOps strategy, we mention 10 top DevOps tools which you should use in the year 2020.

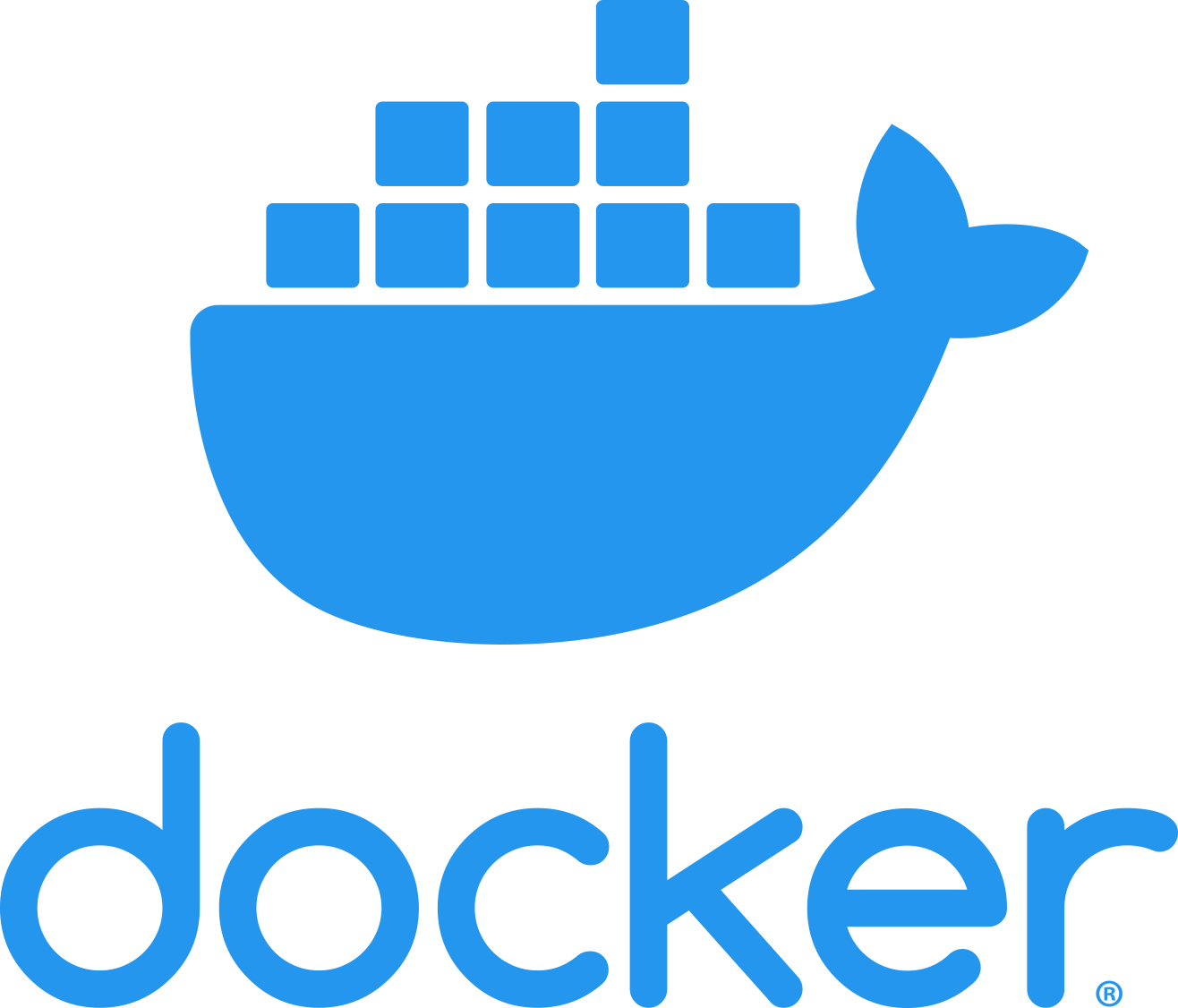
### 1. Slack

Launched in the year 2013, Slack is still one of the top communication tools used by teams for effective collaboration on projects. This tool in the DevOps arsenal is used by technical organizations across the globe to tear down barriers and offer all team members a clear insight into the workflow. One exciting feature about Slack is that it allows developers to collaborate using toolchains in the same environment they are communicating with other maintenance and service members.

### 2. Jenkins

An open-source continuous integration server, Jenkins automates the complete build cycle of a software project. The USP of this tool is the Pipeline feature it offers, which can be utilized by the developers to automatically commit code into the repository, run test cases, as well as fetch reports obtained after testing. This highly customizable tool provides instant feedback and hence will warn you if a particular sprint is leading to a broken build or is harming the same. Most of the tasks and tools involved in [SDLC](https://hackr.io/blog/sdlc-methodologies) can be automated using Jenkins, allowing team members to increase their throughput.

### 3. Docker

Docker is a tool which is at the centre of containerization, a trend which is quickly gaining momentum in the IT world. Docker allows secure packaging, deploying and running of applications irrespective of the running environment. Every application container contains the source code, supporting files, run time, system config files, etc. responsible for application execution. Using the Docker Engine, the containers can be accessed which in turn can execute applications in a remote environment. The app has allowed organizations to reduce infrastructure costs. According to a report, 2 out of 3 companies who have tried this application have adopted it within 30 days of using it.

### 4. Phantom

Security of software is one of the prime concerns of any DevOps team. As such, the Phantom tool comes as a great help to developers who wish to build a defensible infrastructure from the very beginning of SDLC. Using the phantom tool, you can collaborate in a centralized environment on an incident and be aware of the rising security threats at the same time. The tool further gives DevOps professionals an option to mitigate such risks instantaneously using techniques such as file detonation, device quarantine, etc.

### 5. Nagios

Similar to Phantom, Nagios is also a monitoring tool which tends to keep tabs on the applications, servers as well as your overall business infrastructure. The tool comes in as a great help for large organizations which have a countless number of circuitry (routers, servers, switches, etc.) in the backend. It alerts the users in case a particular fault occurs on the backend or any device fails. It also regularly maintains a performance chart and monitors trends to alert the user of a possible failure which may occur.

### 6. Vagrant

A vagrant is a tool for managing and working with virtual machines in a single workflow. Using Vagrant, team members can share software running environment and can test applications faster without wasting time on setting up configurations. The tool ensures that the environment for a particular project remains the same across every developer’s machine, and the excuse of ‘runs-on-my-system’ can be thrown out of the window.

### 7. Ansible

Ansible is one of the most simple yet effective IT orchestration and configuration management tools available in the market. Compared to its competitors such as Puppet and Chef, which are loaded with features, Ansible offers a softer outlook and doesn’t hog on your device’s resources in the background. This tool is primarily utilized for pushing new changes within the existing system, as well as configuring newly deployed machines. Lowering the costs of infrastructure and increasing the replication speed of scalability are just two of the reasons which have made this an absolute favourite amongst IT companies.

### 8. GitHub

Launched in the year 2000, GitHub remains as one of the top DevOps tools for easy collaboration. Using this tool, developers can make rapid iterations to the code, the notification of which is sent instantaneously to other team members. In case of any error or fallout, immediate rollbacks can be done to the previous version within seconds, thanks to the branched history of changes which are stored contiguously within the tool.

### 9. Sentry

Sentry, a tool used by companies such as Uber and Microsoft, is one of the best DevOps tools for error or bug detection. This free tool supports languages such as Ruby, IOS, JavaScript, etc. and further has inbuilt SDKs which can be customized for supporting most languages and frameworks. The tool continuously scans lines of code across the entire system and sends notifications if it finds an error or problem. Not only does it highlight the problem, but offers a choice of possible solutions which can be incorporated with a single click.

### 10. BitBucket

Similar to GitHub, BitBucket is also a tool which helps manage project code throughout the software development cycle. While GitHub still ranks as the top repository, people are shifting to BitBucket due to its reduced cost as well as the feature of the private repository (a feature only available in the paid variant of GitHub). While the core functionality of BitBucket resembles that of GitHub, features such as easy integration with Jira and Trello, inbuilt CI/CD functionality tend to give this tool by Atlassian an edge.

Testing:**Software testing** is an investigation conducted to provide stakeholders with information about the [quality](https://en.wikipedia.org/wiki/Software_quality) of the [software](https://en.wikipedia.org/wiki/Software) product or service under test.[[1]](https://en.wikipedia.org/wiki/Software_testing#cite_note-Kaner_1-1) Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding failures,[[2]](https://en.wikipedia.org/wiki/Software_testing#cite_note-IEEEglossary-2):31 and verifying that the software product is fit for use.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

* meets the requirements that guided its design and development,
* responds correctly to all kinds of inputs,
* performs its functions within an acceptable time,
* is sufficiently usable,
* can be installed and run in its intended [environments](https://en.wikipedia.org/wiki/Operating_environment)
* achieves the general result its stakeholders desire.

As the number of possible tests for even simple software components is practically infinite, all software testing uses some strategy to select tests that are feasible for the available time and resources. As a result, software testing typically (but not exclusively) attempts to execute a program or application with the intent of finding failures[[2]](https://en.wikipedia.org/wiki/Software_testing#cite_note-IEEEglossary-2):31 due to software faults.[[2]](https://en.wikipedia.org/wiki/Software_testing#cite_note-IEEEglossary-2):31 The job of testing is an iterative process as when one fault is fixed, it can illuminate other failures due to deeper faults, or can even create new ones.

Software testing can provide objective, independent information about the quality of software and risk of its failure to users or sponsors.[[1]](https://en.wikipedia.org/wiki/Software_testing#cite_note-Kaner_1-1)

Software testing can be conducted as soon as executable software (even if partially complete) exists. The [overall approach to software development](https://en.wikipedia.org/wiki/Software_development_process) often determines when and how testing is conducted. For example, in a phased process, most testing occurs after system requirements have been defined and then implemented in testable programs. In contrast, under an [agile approach](https://en.wikipedia.org/wiki/Agile_software_development), requirements, programming, and testing are often done concurrently.

These tools help organize the end to end test Cycle

### **1)** [**TestRail**](https://bit.ly/2KlmTfP)

[TestRail](https://bit.ly/2KlmTfP) is your source for scalable, customizable, web-based test case management. Set up in just minutes with our cloud-based/SaaS solution, or install on TestRail your own server.



* Efficiently manage manual and automated test cases, plans, and runs.
* Get real-time insights into testing progress with informative dashboards, metrics, and activity reports.
* Boost efficiency with milestones, personal to-do lists, and email notifications.
* Document test cases with screenshots and expected results. Use the flexible built-in templates or create your own custom templates.
* Integrate with tools in your CI/CD/DevOps pipeline including JIRA, Bugzilla, Jenkins, TFS and more.
* Enterprise edition designed for large teams & mission-critical projects.
* Support for Docker containers.



### **2)** [**Testpad**](https://bit.ly/2CpRy6u)

[Testpad](https://bit.ly/2CpRy6u) is a simpler and more accessible manual test tool that prioritises pragmatism over process. Instead of managing cases one at a time, it uses checklist-inspired test plans that can be adapted to a wide range of styles including Exploratory testing, the manual side of Agile, syntax highlighted BDD, and even traditional test case management.



**Key features:**

* Guest testers, invited by email, who don’t need accounts
* Simple enough to use by non-testers; get everyone to help at release time
* Keyboard-driven editor with a javascript-powered (i.e. responsive) UI
* Drag'n'drop organisation of test plans
* Add new tests during testing, as you think of new ideas
* Lightweight integration with issue trackers, including JIRA

### **3)** [**Xray**](https://bit.ly/2oV4nm5)

****

[Xray](https://bit.ly/2oV4nm5) is the #1 Manual & Automated Test Management App for QA. It’s a full-featured tool that lives inside and seamlessly integrates with Jira. Its aim is to help companies improve the quality of their products through effective and efficient testing.

**Features:**

* Traceability between requirements, tests, defects, executions
* Define reusable preconditions and associate to tests
* Organize tests in folders and test sets
* Test plans for tracking progress
* Test environments
* BDD - Write Cucumber scenarios in Jira
* Integrates with test automation frameworks (Selenium, JUnit, Nunit, Robot, ...)
* Built-in REST API
* CI integrations (Bamboo, Jenkins)
* Built-in reports

### **4)** [**Practitest**](https://bit.ly/2B4MFzK)

[PractiTest](https://bit.ly/2B4MFzK) is an end-to-end test management tool. A common meeting ground for all QA stakeholders, it enables full visibility into the testing process and a deeper broader understanding of testing results.



**Features:**

* A vast array of third-party integrations with common bug trackers, automation tools, and robust API for the rest.
* Fully customizable & flexible for the ever-changing needs of QA teams: customize fields, views, permissions, issue workflows and more
* Reuse tests and correlate results across different releases and products.
* Unique hierarchical filter trees - organize everything and find anything quickly.
* Never work twice - with anti-bug duplicates, permutations, step parameters and call to test
* Visualize data with advanced dashboards and reports
* Fast professional and methodological support

### **5)** [**Zephyr Scale**](https://bit.ly/2LNgvS0)

[Zephyr Scale](https://bit.ly/2LNgvS0) is a scalable, performant test management solution inside Jira, with advanced test planning, reporting, and reusability features.



**Features:**

* Reuse tests and eliminate duplication: cross-project hierarchical test libraries, parameters, test data, and shared steps
* Improve visibility, data analysis, and collaboration: Access over 70 cross-project reports, 60 gadgets, and 60 Confluence macros
* Easier to audit and trace: Detailed change history, test case versioning, end-to-end traceability with Jira Issues and Confluence pages
* Test automation and DevOps ready: Empower teams with BDD, CI/CD and automation integration using the built-in, free REST API. Publish automated test-execution results from Jenkins, Bamboo, and other tools

### **6)** [**SpiraTest**](https://bit.ly/3iXun8w)

[SpiraTest](https://bit.ly/3iXun8w) is a state-of-the-art solution for managing requirements, plans, tests, bugs, tasks, and code in a single environment. It is designed to be fully tailored to your needs, methodology, workflow, toolchain with the industry best practices baked-in.



**Features:**

* seamlessly generate tests from requirements, and bugs from tests
* easily manage test cases, sets, and runs in manual and exploratory testing
* create tests based on parameters with end-to-end traceability throughout
* executive dashboards for a top-down view of your projects, visualization, and business analytics
* integrates with over 60 apps on the market
* works out-of-the-box with minimal configuration
* flexible concurrent user pricing with unlimited products, projects, sprints, tests, API calls included in the price.

### **7)** [**TestMonitor**](https://bit.ly/2JvtS5a)

[TestMonitor](https://bit.ly/2JvtS5a) is an end-to-end test management tool for every organization. A simple, intuitive approach to testing. Whether you’re implementing enterprise software, need QA, building a quality app or just need a helping hand in your test project, TestMonitor has you covered.



**Features**

* Requirement and risk-based testing.
* Advanced test case design capable of supporting thousands of cases.
* Robust planning tools with multi-tester runs and milestone cloning.
* Comprehensive result tracking.
* Integrated issue management.
* Smart reporting with many filter and visualization options.
* Revolutionary simple UI.
* Third-party integrations featuring Jira, DevOps, and Slack. REST API included.
* Professional support with quick response time.

## **Automated Testing Tools**

This category of tools helps automate functional and [Regression Testing](https://www.guru99.com/regression-testing.html) of your application under test.

### **8)** [**Kobiton**](https://bit.ly/3gAGWp8)

[Kobiton](https://bit.ly/3gAGWp8) empowers testing and development teams to automate Functional, Performance, Visual, and Compatibility testing across real mobile and IoT devices.



**Solutions:**

* Access to Real devices in a Public or Private Cloud
* "Cloudify" local devices for shared remote access
* On-premises solutions
* Scriptless/and or scripted automated Functional, Performance, Visual, and Compatibility Testing
* Support for leading open-source automation tooling and frameworks
* Generate 100% open-standard Appium code with every scriptless test
* AI-assisted remediation of Functional and Visual issues
* Unlimited user policy
* In-depth test session exploration
* Access and share analytics around app quality
* Access to real devices within your IDE for rapid debugging

### **9)** [**Ranorex**](https://bit.ly/2Y1VNRf)

Over 14,000 users worldwide accelerate testing with [Ranorex Studio](https://bit.ly/2Y1VNRf), an all-in-one tool for cross-platform test automation. Ranorex is easy for beginners with a codeless click-and-go interface, but powerful for automation experts with a full IDE.



**Features include:**

* Robust object identification for reliable tests and reduced maintenance
* Shareable object repository and reusable code modules
* Cross-browser testing
* Automates challenging interfaces like SAP, ERP, Delphi and legacy applications
* Run tests in parallel or distribute on a Selenium Grid
* Built-in reporting

Ranorex integrates with solutions for BDD, CI/CD, source control, test management, defect-tracking, and more for a complete test automation toolchain

How to write a TEST CASE Software Testing Tutorial

### **10) Selenium:**

****

Selenium is one of the most popular software testing tools. It specifically designed to support [Automation Testing](https://www.guru99.com/automation-testing.html) of functional aspects of web based applications, wide range of platforms and browsers.

**Features:**

* It is one of the best qa tools which offers the support for parallel test execution that reduce the time taken in executing parallel tests.
* Selenium needs very lesser resources when compared to other manual testing tools.
* Test cases prepared using this testing tool can be executed on any OS
* It supports the many known programming languages like Java, Python, C#, Perl, PHP, and JavaScript.