How to Choose a Programming Language For a Project?

unsetLast Updated : 26 Sep, 2020

If you're a **non-technical** person than here are some questions for you…

Would you use Axe or Chainsaw? What do you choose?

The first question you might be thinking is…***for what?***

If we answer stitching cloth…you will say **NO** and if we answer chopping wood then you may respond with **YES.**

Now let's come to another question…

If you need to build a house from scratch what would you choose? hammer or drill?

Before jumping to any answer, don't you think that it's important to know how big you want to build your house? How many rooms do you want? On which land you want your house to be built? It would build of timber or brick?

Now think about the below if you're a **technical person** or you're in software development…

- How would you choose a language for an iPhone, if there are only two of them?
- How would you choose a language for Android, if there is only "one of them"?

In software development, if we need to choose a language for a project, we need to ask a few questions to ourselves before we make any decision. For example what kind of project is it? scalability of the application, the complexity of the application, development budget, development time limit, application security, available resources, etc. The project team always wants the application to stay for the long haul and fulfilling the needs of the customer even if the business changes occur later on.

Image Source: Stackoverflow Survey

Some Quick Tips

- Choosing a language for a project in the corporate IT sector is one of the big issues technology managers struggle with. While choosing programming languages the most important thing you should consider is the ecosystem, the community, and the availability of programmers.
- A lot of developers make mistake and choose programming languages just because it is more popular, trendy, and cool. If you chose Lisp for a project just because it is a very clean and beautiful functional language then later it may turn into a wrong decision. So it's good to avoid this mistake.
- Programming language for a project should be based on your business needs not just because it has some syntactic sugar or it is hyped up. Maybe you feel that since you're the developer it should be totally up to you, your choice to implement any language. Only you have all the freedom to choose the technology whatever you want but this doesn't work in an IT organization and it may backfire you.
- As a technical manager, firstly you need to pay attention to all the moving pieces of your project. You should know all the components for a better view and this will help you to choose a specific programming language. A good view at the beginning of your project helps in choosing a sensible programming language and this leads to less time spent in maintaining the project, scaling up the project, and securing the project later on.
- In programming, if you can write good software in Java, C#, Python, PHP, or any other languages then you can also write bad software using these languages. No single language is the best choice for any software. Some languages and frameworks are a better fit for projects than others. Consider the example of Java. It wasn't a good language the moment it was created. It was just more convenient than competitors. While choosing a language you need to think in this way.

Now let's discuss some questions that you should ask while choosing a programming language for a project…

Questions to Ask When Choosing a Programming Language

- Does the language have proper ecosystem support? Is it going to work for the long haul? Is the vendor support available for the language?
- What is the environment for the project (web, mobile, etc)?
- Do we need to consider some infrastructure such as new hardware? What kind of deployment do we need?
- What's the preference of the client?
- Any specific requirement of libraries, features, and tools for the programming language?
- Is the developer available to code in this language or do we need to hire new developers? Are they experienced and comfortable in working with this language, or they need to learn the language quickly?
- What are some important constraints of this project? Time, budget, resources?
- What's the performance consideration and is the languages suitable to accommodate this performance?
- What's the security consideration and do we need to use any third-party tool?

A lot of questions given above clear out the confusion of choosing a programming language for your project.

Now in programming suppose you have a task that is very heavy on the number crunching. In this case, you need a language that supports some libraries to performs complex maths operation for you and can handle the numbers easily. Another example you can consider is processing a lot of text. In this case, you need a language whose features set allows you to easily slice, dice, and search textual data.

To choose the right programming language for your project, you need to consider different factors such as performance, type of application, security, etc which we have discussed in the section of question we need to ask. Now let's discuss these factors in detail to use an appropriate language for a project but remember that there's always some trade-off.

1. Type of Application

You should have a clear purpose or goal in your mind before choosing a programming language for a project. The choice of language heavily depends on the type of application. What kind of application you are going to build? Is it a web application or mobile application or it's embedded firmware etc. This is the first and one of the biggest factors you should consider before you choose any programming language for a project.

Also, you need to consider the complexity or size of the application. Small projects such as web application form, portfolio presentation, simple marketing website, or simple personal blog can be built with some CMS systems like WordPress (requires PHP knowledge) or Umbraco (requires C# knowledge). Java or **C#** can be used for medium size projects such as e-commerce sites, internal enterprise applications, and IoT solutions.

For example, Object-oriented languages are the best suitable for web applications. For Android application, Java or Kotlin is good, for system programming middle level languages such as C can be used.

- Front End Development: JavaScript, HTML, and CSS
- Backend Development: Python, C# or JavaScript
- 2D Game development: JavaScript or C#
- 3D Game Development: C# or C++
- Data Science/Machine Learning/Analytics: Python, R, Clojure, Julia
- Math & Scientific Computing: Matlab, FORTRAN, ALGOL, APL, Julia, R, C++
- Big Data: Java, Python, R, Scala, Clojure
- Operating Systems: C, C++
- Distributed System: C, Go, Rust

2. Targeted Platform

Another factor you need to consider is the targeted platform where you want to run your program. Let's say you have two languages Java and C. If you have written a program in C and you want to run that on Windows and Linux. In this case, you need platform compilers and two different executables.

In the case of Java language, the bytecode will be generated, and to run that on any machine you should have Java Virtual Machine installed. A similar thing goes with the websites. It should look and work the same across all browsers. If you're using CSS3 and HTML5 tags then your website may look and behave differently across different browsers.

3. Maintainability

Whatever technology and language we choose, we get the ecosystem of libraries and vendor support. We need to pay attention to the maintainability of the application and this is the reason we should always look at the latest release of the language or technology stack. Make sure that whatever we pick is current and stay current for a long time.

While building an application you need to think about engaging with your clients for the long term and at some point, you will have to transition your codebase to another team. You will have to hire the internal developers in your area depending on the technology you will be working on and you will have to pay the maintenance cost for that.

This maintenance cost will be affected by the availability and proficiency of the development team in your area. For example, In the Midwest, it's easy to hire C# and Java developer than Ruby, Python, and PHP developer. The whole point is…you need to think about the ease of maintenance and the long term survival of your application while choosing a programming language for your project.

4. Scalability and Performance

The performance of your application heavily depends on the choice of the programming language. It matters a lot when the development environment doesn't offer much scope for scaling. To squeeze out so much performance from your application you need to choose the right programming language.

How many users your application can handle and what will happen in the future if the numbers of users will increase? Is the growing number of users going to slow down the performance of your application or is it going to crash your application? Think about the scalability of your application. Is your application flexible enough to accept more add-ons without hampering the performance of your application?

A lot of studies have been conducted measuring the performance of the application. However, the results were not the concrete assessments of the performance of programming languages. One example of this kind of environment is handheld devices. Some popular tech stack that has good scalability is Ruby on Rails (RoR), NET, Java Spring, LAMP and MEAN.

5. Security

You also need to think about the security of your application while choosing a programming language for your project. Every application has different kinds of security requirements. For example, healthcare-related applications need HIPAA compliance and financial application need PCI compliance. Your application will become more robust if you keep your eye on the security of your application.

You also need to protect your application from cyber threats and hacking. You need to follow the security guidelines before choosing any programming language for your application.

6. Community Support

During the development of any application you will face many issues and it can create a big problem if there is no enough community support available. While building an application you may get stuck at any moment and you may have to take the help of some resources or some other developers. This is the reason you should pay attention to this factor as well.

If you are choosing a language that doesn't have many tutorials or the articles to learn then it will be tough for you to adapt the language and work on that. Before choosing any programming language it's good to check out the video tutorials, articles, forums, and some community support such as GitHub, Reddit, or StackOverflow.

7. Development Time Limit

Every organization wants to release the application as soon as possible. Business stakeholders also want new features or the application to be ready as soon as possible. This is the reason you should pay attention to this factor that is…development time.

You should choose a language that can help you to build the application as quickly as possible. If enough library support, packages, or modules are available for a language then you won't have to build everything from scratch and you will save the development time for your application.

Here you need to pay attention to the experience in your chosen language. The more familiar you are with the language the less time you will take for the development. In other words, if a language is easy to learn, then you won't take much time to learn it, produce the code and make your application live on the server.

Final Thought

With a list of questions to ask, a list of programming languages available, and considering the different factors, you can make a sensible decision regarding the programming language that you use. It's good to compare the languages based on the above factors to choose the right one for your project.