

Which Language

RUBY JAVA PHP WTF...?
What language to use?

Factors

1. People cost
2. Speed of Implementation
3. Ease of Maintenance

CHOICE?

	PERFORMANCE	SPEED OF IMPLEMENTATION
Ruby	1	6
Python	2	5
PHP	3	4
Java	4	3
C++	5	2
C	6	1

Types

1. Open Source
2. Compiled
3. Scripted
4. Object Oriented

Paul Clifford

whichLanguage

Video Transcript:

Paul: Hi, it's Paul Clifford and welcome back to today's session, which is all about what language to use. What language should you actually use to build your online [SaaS app](#)? There's loads and loads of phrases and languages that you would have heard of and in fact when I last did a count there's like 309 different languages in technology which you could use.

So how do you know? Now, if you go to any developer, they will have their own opinion. If you look on the web, there will be tons of different opinions and it can be a mine field, so let me just look at the key things that you should be looking at.

Now, first of all, the factors to consider...

1. **People Cost**
2. **Speed of Implementation**
3. **Ease of Maintenance**

Number one, **the people cost**. How much is it going to cost you to build your application? How much are the people? And how much do you have to pay them per hour or per project if you get a fixed price? What is the **speed of implementation**, so how quickly can you actually get your build into operation? How quickly can it get up there and get customers using it? Of course **ease of maintenance**, once you built it and you got it out there, what about bug fixes, new features, customer requests? Everything that comes in, how quickly can you put those into action? So, it's ease of maintenance.

Now those are the three, but there is of course **performance**. How quickly the language functions in itself and that sometimes is worth considering with online apps. All the languages, [the key languages](#) I'm going to discuss today are pretty quick. Unless you're building like a rocket ship or the next shuttle or something like that, then don't worry about that too much.

Types of Languages...

1. **Open Source**
2. **Compiled**
3. **Scripted**
4. **Object Oriented**

The other thing to consider is what types of languages are they? First of all, are they **open source**? In other words, can your developers get using these straight away or do you have to buy licenses for them to actually code in these languages.

Secondly, are they **compiled** or are they **scripted**? A compiled language, once it's written, then needs to be turned into something that should run physically on the machine, and once it's compiled, it's very, very quick. When it's scripted and it runs in a scripted form, then it can run slower. But because it's scripted, it can be quicker and easier to make changes especially when you're working on the fly.

Lastly, is it **object-orientated**? In object-orientated, what's important about that is that it helps the ease of maintenance, so if it's well-structured and well-organized, then it's going to be easy to maintain moving forward.

Once you've understood those key things, then what are the choices? Well, I've just picked out the top six if you like. The top six that people will discuss if you talk to them in a conference or something like that or when you talk to a developer. I want to give you the tools and the languages and the technology understanding to know what you're talking about so you can have an informed and considered conversation.

Languages for consideration...

1. **Ruby**
2. **Python**
3. **PHP**
4. **Java**
5. **C++**
6. **C**

Now, the main ones considered are **Ruby, Python, PHP, Java, C++, and C**. Now generally, as a general rule of thumb, C and C++ right at the bottom here are more used for real time sort of applications. So like I said, building the shuttle, building something that needs to run really, really, really quickly. If I score all of these languages in terms of performance, speed of implementation, getting it out to your

customers, ease to maintain, ease of maintenance slope, cost to bug fix, et cetera, and availability of skill.

Then if I factored all those and I look at those, then first of all if I look at **performance**, so from the slowest to the fastest, Ruby is probably one of the slowest - Ruby and Python in between them, then comes PHP, then Java, C++ and C. As I said before, those two at the bottom, C++ and C are the fastest and that's why they're used when building missiles and controlling aircraft and things.

If you look at the **speed of implementation**, in other words, how quickly can a developer build something and get it out there? Putting aside the knowledge of course, then Ruby, Python are the fastest to build things, and then comes PHP, and then Java, and C++ and C which take longer to actually build applications with. Then, if you look at the **ease of maintenance**, again, because Ruby is quite quick in terms of implementation and it's well-structured in object-orientation, then Ruby sits at the top along with Python, and then PHP is probably the easiest to maintain, to get started, and then of course you got Java, C++ and C, which get harder.

This is probably the most important thing. If you consider all of those languages, **which one has the most programmers** and this is where it's really important to get, to make your decision based on your first factor here which is people cost, because basically, new technologies and fewer people mean more cost because the **new people with the new technologies they cost more**. That's the thing. If you've got something that's been out there for a long time that are stable and there's a big community of developers, then it's going to cost you less to actually put into action.

If you look at availability of skill, the most available is PHP because it's so common especially for web. It was designed for the web and if you do any searching on any outsourcing site, you'll immediately see

the most volume of developers in the community with those skills, and then it goes sort of PHP, then C, and then Java, C++, but Ruby and Python, not many of those around. They're quite new languages. I mean Python replaces PHP in many ways. It's a great, great language and developers would love it and the same with Ruby, but the people are in demand. There aren't many around, so when you look at all the factors, really **PHP is a great choice to go for**.

This is Paul Clifford. I hope you enjoyed that. Sign up below and get more hints and tips into your mailbox.

Are you ready to become a software entrepreneur?

Get everything you need at www.disruptware.com/.