Joshua Milian

10/9/19

1

New programming languages are continuously being introduced as the years passes by. Hundreds of languages exist already and the reason why new language are getting created is because there is a big demand in this field. The reason why there is a big demand is because of people out of college stops pursuing this career path because of its difficulty. As a result, newer languages are getting created to make things more easier and simplified for new beginners. With creating a more simple language for newcomers, this will create more people to fill up the demand in this field. It also make it a bit easier for software engineers to code at work. New languages would most likely be created in the time span of my own career and the way I would keep up with these new developments is to simply try and learn any new languages. If I prefer the new language, I would simply adapt to this new language. When there's a new releases in a language I'm already fluent in, I would also test it out to see if i'm more comfortable with that version. If I am, I would shift over to this new one.

2

When being a software engineer, there are other tasks besides coding. There are 3 duties that software engineers perform that are not the activity of programming. One of the duties would be documentation. A software engineer would prepare documents for the code so it would be easy to maintain the code by other coworkers or even fix a problem by looking at the documentation to interpret their code. Another duty would be unit testing, where developers would need to design a unit test case to code so it can be error free. The third duty would be to call a Scrum meetings, which is when a team would meet up together to update each other on their progress. For example, they can discuss how much progress they made the day before, what their plans are for the day and If there are any impediments in the way. These duties are significant because this is what leads towards improvement and more enhanced communication skills for a team/yourself. This would provide more experience to the developer to become more proficient at what they do at work.

3

The software development life cycle (SDLC) has several stages: define, design, develop, deploy and maintain. The standard stages would be planning & requirement gathering, system analysis, design & development (which includes coding), testing, and maintenance. The "Planning & Requirement Gathering" phase is all about the blue print on what you’re planning to create and the materials you are going to need to make it happen. The "System Analysis" phase is to study and interpret the blue print to determine how to accomplish this design. The "Design & Development" phase is when they start creating the program, which includes the coding to make it function. The "Testing" phase would be when you get volunteers to test out your design, so you’re able to receive feedback on what should be implemented or improved. Lastly, the final phase would be the "Maintenance" phase, which is when the person oversees and makes repairs, corrects problems, etc. A model that encourages customer involvement and feedback would be the Evolutionary Prototype Model. The steps for this model are to initialize the concept, design & implement initial prototype, refine prototype until it’s acceptable, then lastly complete and release the prototype. This model is discovered by customer feedback on each release so when customer get asked "What do you think of this version?", it allows them to receive feedback to help improve their program. This helps encourage customer involvement in the software development because the customer can get what they ask for based on their feedback and developers know what to implement based on their customer’s responses.

4

If I was in charge of forming a productive software development team of 5 people, one of the qualifications I would look for in an individual is someone who works well with others in a team environment. A person with great communication skills, experienced, open to new ideas, ambitious and stress resistant. Some strategies for forming a productive team would be knowing their strengths and weaknesses, setting goals, giving the team ownership, building a strong and trusting relationship, and providing support to the team. During the hiring phase, one of the things I would take into consideration is how would this individual fit in with the team. I would assess their strengths and weaknesses and determine how it would work overall with the team. I would ask questions about their working style, describe a time when they had to overcome an obstacle and if they have any professional achievements.

5

New programming languages are continuously being introduced as the years passes by. Hundreds of languages exist already and the reason why new language are getting created is because there is a big demand in this field. The reason why there is a big demand is because of people out of college stops pursuing this career path because of its difficulty. As a result, newer languages are getting created to make things more easier and simplified for new beginners. With creating a more simple language for newcomers, this will create more people to fill up the demand in this field. It also make it a bit easier for software engineers to code at work. New languages would most likely be created in the time span of my own career and the way I would keep up with these new developments is to simply try and learn any new languages. If I prefer the new language, I would simply adapt to this new language. When there's a new releases in a language I'm already fluent in, I would also test it out to see if I’m more comfortable with that version. If I am, I would shift over to this new one.

6

In 100 trillion years into the future, if aliens find a code snippets in an Object Oriented Programming language, I would assume that the alien can decipher the program languages. I would assume that the aliens would decipher the Python program language because in my opinion, it was way more simple than java and C++. The reason why is because there’s much less memorization when coding in Python. For example, there’s no need to initialize the data type (Int, Double, String and Char). In python, you begin coding on a blank page, but in java, you need to start with "public static void main(String []){}" or in C++ where you need "#include<string> using namespace std; int main(){}" to able to start coding. An alien would probably wonder what those text are for. In Python, you don't need any of those text to start coding. In order to print out to the console in python, you must type in " print("some text") ", but in java, you need to write out "System.out.println("some text")" and in C++ " cout << "some text" ". A functional programming language an alien would most likely be able to decipher would be Scheme because it is very simplified and similar to python.

7

The performance metrics I would consider when analyzing and comparing different database product options is a database that would carters to every ones needs. A user friendly database where it wouldn’t be difficult to use for the users. Some metrics I would take into consideration is how efficient each database is and to view if the information being stored would be small or big. Also we would have to consider if the database can grow because if it reaches its limit, then you can’t not add any more data to the database. That’s why it’s important to find a database that can grow by itself once the limit has been reached to be able to add more space to the database. A database I would consider using would be SQL database, the user can quickly and efficiently retrieve a large amount of records from a database and it doesn't require a large amount of code to manage the SQL.

8  
A forward declaration is a declaration of an identifier which tells the compiler that an identifier exists before it is actually defined such as denoting a type, a variable, a constant, or a function. For which the programmer hasn't yet given a complete definition. In Python or Java you do not need a forward declaration because identifiers are recognized automatically from the source files. I would prefer programming languages to not require forward declarations because it's more stuff to memorize while coding. Without forward declaration there is less writing when it comes to coding. In scheme programming language you are not required to use to forward declaration just like Java/Python.

9

An example I would use for recursion would be the factorial example. Recursion is a common method of dividing the problem into sub problems until it hits its base case. Iteration repeats parts of a code where recursive repeats the entire function itself. The advantages for recursion is that it can reduce time complexity and the time needed to write and debug code. People use recursion only when it’s too complex to write iterative code or for tree traversal techniques like preorder, post order.

10

The sample queries that would be performed would be to grab the data from the box office website. I would choose the “Top 2019 Movies” where it would retrieve the top movies from the year 2019 and the generated revenue for those movies. It would be listed from descending order of the generated revenue. It would show the top movies up from 1 to 100 for that current year. Also it would show the ranking, release date, closing date and the total amount of theaters the movie was showed at. The way I would show the stats to the manager would be through a csv files. It would be easier for manager to view the stats in csv file then paper work because they can always have the data on their phone or on a computer. Also there is less paper work to handle, where sometimes paperwork can get lost with other paperwork’s.