Trends in Software Technologies: the Developer's Viewpoint

Alexis Koster
Department of Management Information Systems
San Diego State University
San Diego, CA, USA

Abstract

This paper analyzes data provided as a result of a large survey by stackoverflow.com, a website where developers post their programming problems and other developers post answers and solutions. A large number of software technologies are indicated, including operating systems, frameworks, integrated development environments and other software tools. The most often mentioned technologies are computer languages. The three most popular languages are Javascript, SQL, and Java. The survey also indicates a renewed interest in the paradigm of functional programming. Finally developers are some of the most satisfied of all workers. They are well-paid and like many aspects of their work.

Introduction

Before the advent of the internet, software technologies consisted mostly of programming languages, operating systems, and system software (such as compilers). The pervasive use of the Internethas opened the floodgates to the creation and use of a large number of software technologies. The widespread use of smartphones, the growth of cloud-computing, and the processing of "big data" have magnified this phenomenon and amplified the need for qualified personnel. It is not surprising that computer science is now a popular major, for example the top major at Stanford University [8].

Following the terminology of the survey analyzed here, the word "developer" is used. It should be noted that this is not a standard name in academia. Although they are not completely synonymous, developers are also called software engineers or programmers. The Bureau of Labor Statistics classifies them as "computer and information technology" workers [3]. Typically many develop computer applications, some create and maintain software technologies. In most cases, they perform their work by writing computer programs.

The Top Software Technologies

Close to 50,000 developers provided information about their use of software technologies. Operating systems, frameworks, integrated development environments and other software tools, many of them fairly recent, are among those technologies. The most often mentioned technologies are computer languages. Here, computer languages consist of programming languages (example: Java), scripting languages (example: JavaScript), and database languages (example: SQL). Programming languages are typically used for complex tasks. Writing a program in a programming language is time-consuming, as it requires a heavy setup as well as the use of some complex features. Scripting language programs consists of scripts, “small” programs that automate the execution of tasks in a run-time environment. Javascript, for example, is used by web browsers. Scripts are much easier to write than other types of programs, hence they are very popular. Each script instruction is executed from the source program at run-time (the script is “interpreted”). This makes the testing and debugging of the script easy. Note that the distinction between those two types of languages is blurry. Python can be viewed as a programming language or a scripting language. For the third type of computer language, namely database language, there is a de facto standard language: SQL. A SQL command can be used stand alone to process a database (create a table, retrieve data, update data) or can be embedded in an application program written in another language.

The three most popular languages are Javascript (used by 55% of the respondents), SQL (49%), and Java (36%). It is not surprising that JavaScript tops the list, as most of the survey respondentprogram web applications.

The place of databases at the center of information systems of most organizations and of most applicationsexplains the popularity of SQL. SQL runs on databases systems as different as Oracle, Microsoft SQL Server, Microsoft Access, SAP, and open source MySQL. The SQL language is an exception among computer languages, as it is quasi-standard. Initially developed at IBM in the 1970’s, it has been popularized by Oracle, the dominant provider of database systems. It survived very strong criticisms by Edgar Codd, the inventor of the relational database technology [4] and by other well-known database researchers.

Java was released in 1995 by Sun Microsystems (now part of Oracle). It is used in many areas of software development, such as website development (frontend and backend), mobile development such as Android, and large corporate programs. Unlike C++ (its initial competitor), it has a unity of design around objects and classes, and was the first language to provide an extensive library of classes, in particular for graphical applications. It is often the first programming language taught in computer science departments [7].

Many programming languages are enduring. The C language, which started to be used circa 1972, ranks eighth on the list of most popular languages, with 15.5% of users. Even Fortran, born about 1960, appears on the list (11th among languages, with 0.14% of users).

Developers often use combinations of technologies. The top double combination consists of JavaScript and SQL, used by 30% of respondents. This is explained by the fact that many web applications require access to a database. The combination “Java, JavaScript” ranks second in this category, and “Java, SQL’ ranks fifth.

The Surge of Functional Programming

The paradigm of functional programming seems to be surging. Functional programming is a style of programming that differs from procedural programming in several aspects. Functional programming-based programs specify what actions to perform rather than how to perform them. They avoid the use of imperative features such as assignments to variables and looping constructs. One of their advantages is expressive power. For a given task, it is often simpler and much shorter to write a functional program than a procedural program. A second advantage is that functional programming provides for parallel execution of many instructions whereas other languages require sequential execution. John Backus, winner of the Turing award in 1977, made a strong case for functional programming [1], but without success.For traditional computers, functional programming was too slow, and the potential for parallelism could not be exploited.

Moore’s Law affirms the increase of computing power by doubling the number of transistors on a chip every two years. It is coming to an end [9]. Therefore, parallel processing will be one way to remediate the end of Moore’s Law. Already, some computer systems permit parallel execution of code, for example with multi-core processorsandwith cloud computing. The efficient processing of “big data” wit technologies such as Hadoop [6] also entails parallel processing. Not surprisingly, functional programming is gaining in popularity. Release 8 of Java includes a functional programming component called “lambda” [5]. In the survey of stackoverflow.com, three functional programming languages, “F#”, “Scala” [10] and “Clojure”, rank third, fourth, and sixth respectively as “most loved” technologies, and second, fourth, and ninth as “top paying technologies.” In the area of big data, Scala ranks first as top paying technology. It is worth noting that Clojure is an extension of Lisp, one of the first functional programming languages (mostly used in artificial intelligence).

Developers and Job Satisfaction

According to the survey, developers seem to be some of the most satisfied of all workers. They are well-paid. Unemployment seems fairly rare (only 1.8% of respondents list themselves as unemployed); 15% only are actively looking for a new job. This is corroborated by the Bureau of Labor Statistics , which projects that the number of computer and information technology workers in the USA will grow from 3.9 million in 2014 to 4.4 million in 2024 [3]. The Bureau also indicates that developers earned a median annual salary of $81,000 in 2015, compared to $36,000 for all occupations.

If 63% of respondents list salary as their top priority, many list other work features they appreciate such as flexible hours, work-life balance, and company culture. Technology companies like Google are known to provide an environment where workers are neither constrained by a strict work schedule, nor by strict product development milestones and deadlines. Other positive features of work include "building significant things"and "learning new technologies".This confirms Brookswho, in his seminal software engineering book, speaks of "the joy of the craft" and mentions "the sheer joy of making things" and "the joy of always learning" [2]. Finally, developers are more often working remotely (12% do it full-time).

Conclusion

JavaScript, Java, and SQL, three computer languages, top the list of the most used software technologies. Among newer software technologies, the surging trend toward a larger usage of functional programming languages is noticeable. The ever increasing demand for software developers makes that profession a desirable one and offers them the choice of using established technologies or learning new ones. Software developers rate their work as very satisfying.

References

1. Backus, J., 1978, "Can Programming be Liberated from the von Neumann Style? A Functional Style and its Algebra of Programs." Communications of the ACM, 21(8).

2. Brooks, F., 1975, "The Mythical Man-Month", Addison-Wesley

3. Bureau of Labor Statistics, 2015, "Occupational Outlook Handbook", http://www.bls.gov/ooh/computer-and-information-technology, December

4. Codd, E., 1988, "Fatal Flaws in SQL", Datamation, September

5. Deitel, P and L. Deitel, 2015, "Java How to Program", 10th Edition, Wiley

6. Jurney, R., 2014, "Agile Data Science: Building Data Analytics Applications with Hadoop", Amazon Kindle

7. Koster, A., 2010, “Are Academic Programs Adequate for the Software Profession?”, American Journal of Business Education, 3(3).

8. Meyer, R., 2012, “Stanford’s Top Major is Now Computer Science”, http://www.www.theAtlantic.com,June 29.

9. Mock, R., 2016, “The Chips are Down for Moore’s Law”, Communications of the ACM, 59(2).

10. Odesky, M. and Rompf, T., 2014, “Unifying Functional and Object-Oriented Programming with Scala”, Communications of the ACM, 57(4).