DevelopIntelligence
www.developintelligence.com

What/How/Why Do Software Developers Want to Learn in 2020?

Our Developer Survey Data

Supporting Data From These Industry Leaders:

HackerRank stackoverflow

Intro and Context Setting

DevelopIntelligence has taught hands-on technical classes to more than 50,000 software developers since 2003. In that time span, the internet and technology industries have evolved significantly, which plays a key role in how training is perceived and conducted. For instance, several billion more people now use computers for work, play, and education. Most people in the developed world use their smartphones hundreds of times each day.

Given this massive increase in technology usage, where and how software developers study and learn has changed as well. For this reason, we regularly survey developers on how, why, what, and where they learn. Our clients – mid- and large-sized technology companies – need to understand these details in order to keep their developers productive, happy, and present, or, not poached away by competitors.

Other organizations like StackOverflow, HackerRank, JetBrains and Stripe have also been studying how developers learn. In this report, we've synthesized multiple research and data reports – including our own – in order to help organizations understand how developers learn, what training works best for this critical workforce cohort, and what technology needs they should anticipate in 2019. With this knowledge, we hope companies will create better work environments and active, business-oriented training plans to help developers learn and thrive.

The Main Reports and Data Sources We'll Cover in this Meta Report

→ DevelopIntelligence 2018 Developer Learning Survey Report

Sample/Methodology:
"A total of 858 developers completed the 20-minute online survey in early 2018."

→ HackerRank's Student Developer Report 2018 and 2019

Sample/Methodology:
"A total of 10,351 student developers completed the 10-minute online survey from October 16 to November 1, 2017."

→ 2019 Developer Skills Report by HackerRank / 2018 Developer Skills Report by HackerRank

Sample/Methodology:
"A total of 39,441 professional and student developers completed the online survey from October 16 to November 1, 2017."

→ 2019 Developer Survey by StackOverflow

Sample/Methodology:
"This year marks the ninth year we've published our annual Developer Survey results, and nearly 90,000 developers took the 20-minute survey earlier this year."

→ 2018 Developer Survey by StackOverflow

Sample/Methodology:
"Over 100,000 developers took the 30-minute survey this past January." We chose to incorporate StackOverflow's 2018 survey, as it had a number of good questions that the newer one didn't.

→ Developer Economics: State of the Developer Nation 16th Edition

Sample/Methodology:
"Developer Economics 15th edition reached an impressive 20,500+ respondents from 167 countries around the world."

What You Need to Know About Software Developers

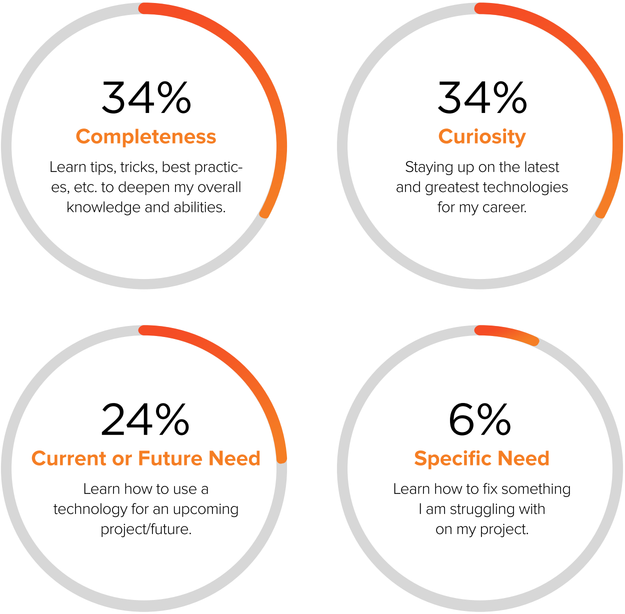
Developers are Constant Learners and Self-teachers.

Much of the research on developers currently on the market shares one critical finding: Developers are constantly learning and self-teaching. They love to learn new coding languages and tools. They habitually pick up new skills by reading, watching videos, and perusing docs.

Before trying to understand how and what developers are learning, it helps to explore 'why' they're doing it. For instance, the 2018 DI Developer Learning Survey Report found that developers learn primarily to satisfy curiosity, or to deepen their knowledge base.

DEVELOPER LEARNING MOTIVATION

Curiosity and skill improvement are not the only motivation



34%

Completeness

Learn tips, tricks, best practices, etc. to deepen my overall knowledge and abilities.

34%

Curiosity

Staying up on the latest and greatest technologies for my career.

24%

Current or Future Need

Learn how to use a technology for an upcoming project/future.

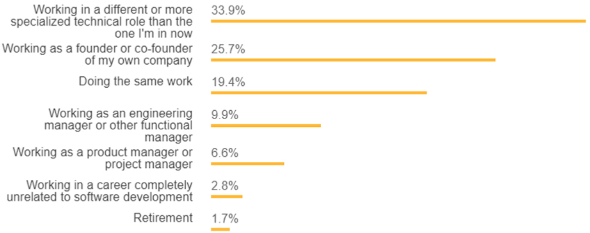
6%

Specific Need

Learn how to fix something I am struggling with on my project.

Source: DevelopIntelligence 2018 Developer Learning Survey Report

Many other surveys corroborate this finding. In a 2018 survey from StackOverflow had more than 100,000 developers around the world answer the question: “What Do You Hope To Be Doing in Five Years?” Here are the results:



Source: StackOverflow 2018 Developer Survey Results

More than 70% of respondents hope to either hold a more specialized role, found their own company, or move into management. Developers in the latter category know they’ll need more knowledge and hands-on experience with the right technologies in order to move up the career ladder and be effective in their new roles.

In the technology sector, developers are in high demand and are known to switch jobs when they want. Indeed, the Stack Overflow survey found that more than half (56%) have been in their current role for less than two years. Keeping their skills sharp and up to date allows them significant upward and lateral career mobility.

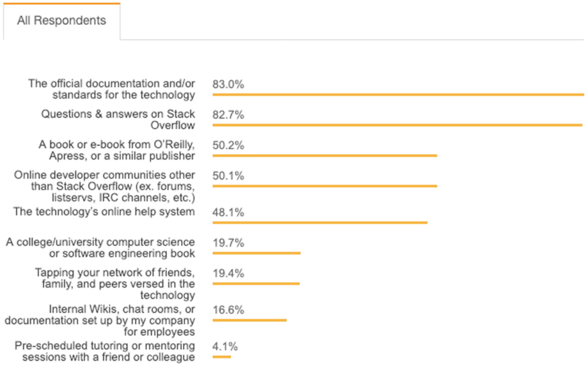
We now have some grasp on why developers learn, but how are they doing it in 2019? As DevelopIntelligence found in previous surveys, a mix of group and solo learning is common. In the Stack Overflow sample, 48% of developers have taken some sort of online video course or participated in a MOOC (Massive Open Online Course). MOOCs from Coursera, Udacity, or Stanford offer friendly video courses taught by top experts, and they’re supplemented by forums, group challenges, exercises, and other social features.

Formal learning is also popular. According to Stack Overflow, some 35% of respondents said they received on-the-job training similar to an expert-led DevelopIntelligence course. Other top responses include participating in coding competitions like HackerRank (24% of respondents tried it) or participating in Hackathons (26%).

DevelopIntelligence recommends combining all of the aforementioned learning modalities into a formal learning program for developers. Instructor-led courses are our bread and butter, but we’ve also helped many companies organize internal hackathons and coding competitions. Furthermore, our instructors regularly recommend the best MOOCs to students to ensure that learning doesn’t stop once a class is over. Learning happens in so many contexts and forms, companies should leverage the ones that developers love to promote knowledge retention and on-the-job application.

Developers also like to learn by reading. Surprising, right? You’d think it would be all about technology, but no. Stack Overflow data revealed that most developers teach themselves by reading documentation, Stack Overflow, books, or forums. It’s a logical off-shoot of their ability to search out and identify the best information at the moment of need.

Ways Developers Learn on Their Own



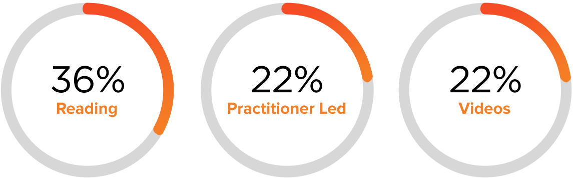
Source: StackOverflow 2018 Developer Survey Results

Therefore, it's helpful for companies and non-technical people to be aware of the popular online developer 'watering holes.' Developers often visit places like dev.to and r/programming to stay current on the latest technical developments, share their knowledge, and occasionally spar with their peers.

The DevelopIntelligence 2018 Developer Learning Survey Report found there is an interesting difference between how less experienced vs. more experienced developers learn. More experienced developers tend to prefer reading and instructor-led training, while less experienced developers tend to prefer reading, videos, and learning from peers more than ILT (Instructor Led Training).

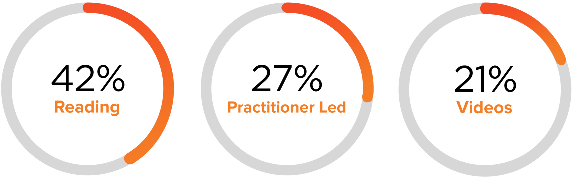
MOST DESIRED FORMS OF TRAINING PER ROLE

Senior Developers (5+ years)



Source: DevelopIntelligence 2018 Developer Learning Survey Report

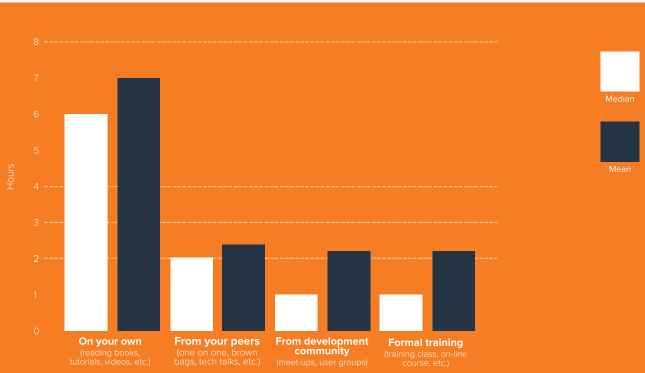
Junior Developers



Source: DevelopIntelligence 2018 Developer Learning Survey Report

A note on self-teaching

As much as we love the amount of self-teaching developers do, it's important to contextualize that learning through an opportunity cost lens. When developers are self-teaching and reading docs, they are not programming and building products. Our 2018 Developer Learning Survey Report found that developers study an average of seven hours per week.



Number of hours spent learning via different methodologies. (Source: DevelopIntelligence Developer Survey Report)

Do the math: Multiply those hours by the high salaries developers command, and you're looking at least $18,000 or more per year in lost productivity, per developer. One hour in an expert-led, hands-on course is worth more in learning dollars than many hours reading docs alone.

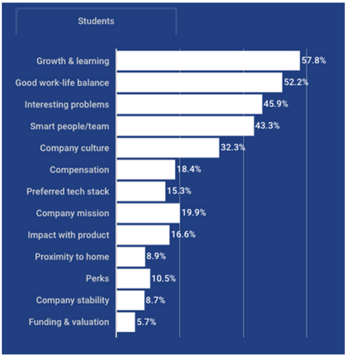
Academia confirms this. There have been some very interesting results when comparing self-paced vs. group learning. The research paper Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching has been cited almost 6,000 times by other papers, according to Google Scholar. We don't want to go too far down the research rabbit-hole here, but one of the key ideas in the paper is that "novice learners….lack proper schemas to integrate the new information with their prior knowledge."

Schemas are like a web that allows different pieces of knowledge to connect to one another. An expert instructor/developer has a fully-built schema around a technology like Python data science, Java 11.0.2, or functional reactive programming such as React. A great instructor will start a class by presenting this web/schema to their students, and then help them navigate it as they learn new material.

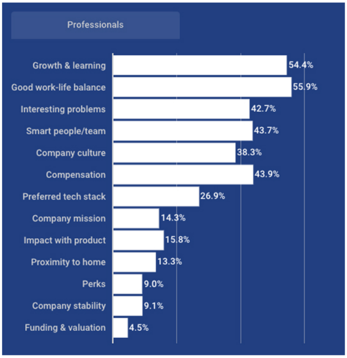
This prevents students from floundering due to lack of information or potentially latching onto a poor or ineffective solution and going too far down the wrong path. The paper states that when students learn "with pure-discovery methods and minimal feedback, they often become lost and frustrated." Minimizing that frustration and providing direction is the instructors' primary role in developer learning. MOOCs and reading docs is helpful, but effective, well-designed instructor-led training will help students learn in a more efficient and holistic way.

Developers are looking for learning-first work environments.

A large percentage of developers choose employers and jobs based on how much they expect to learn. The HackerRank's Student Developer Report 2018 found that the opportunity to grow and learn is the number one quality most university students look for in a job.

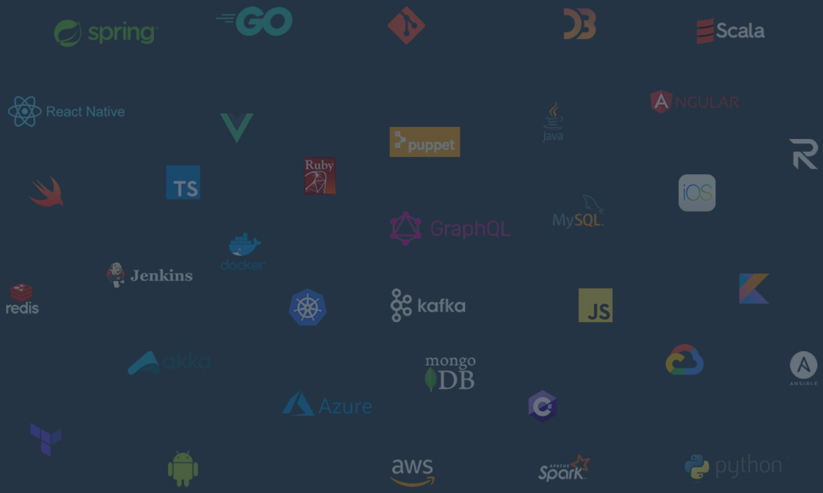


Source: HackerRank's Student Developer Report 2018



Source: HackerRank's Student Developer Report 2018

This data suggests a few things: Most developers prioritize learning on the job. Younger people place an even higher premium on learning. Companies competing for young talent should use learning/growth opportunities as a key marketing asset. Instead of trying to out recruit the FAANG companies for talent based on pay or stock options, companies should use their developer learning and development programs as a talent lure.

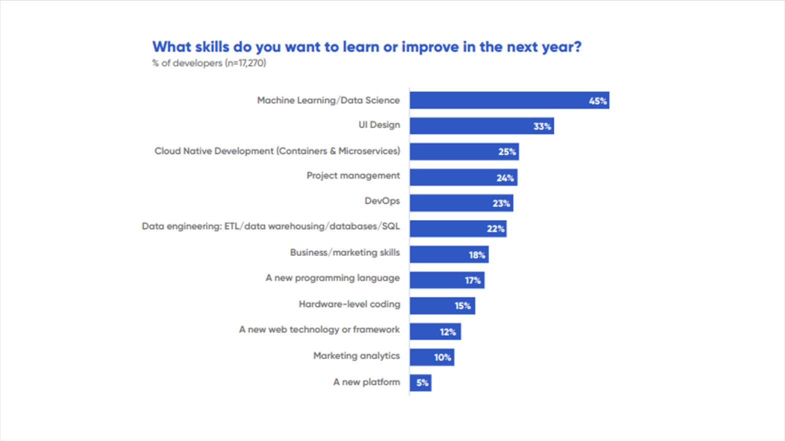


Which technologies do developers want to learn now?

Data and analysis from the following survey reports explore the specific technologies that developers currently use, love, or would like to learn in the future.

Data Science and Machine Learning

Data science and machine learning are changing and redefining many industries, such as finance, energy, and the pharmaceutical industry. New tools and use cases for these technologies pop up every other day within mainstream media, niche blogs, and developer forums. This could mean something ominous like China using these tools to track citizens, or more lighthearted, fun examples like I forced a bot to watch over 1,000 hours of Hallmark Christmas movies and then asked it to write a Hallmark Christmas movie of its own. Developers are aware that data science and machine learning will continue to grow, and many want to get on this wave. Developer Nation's 2018 state of the union report of 40,000 developers found that data science and machine learning were the top technologies developers want to learn by a wide margin.



Source: Developer Economics: State of the Developer Nation 16th Edition

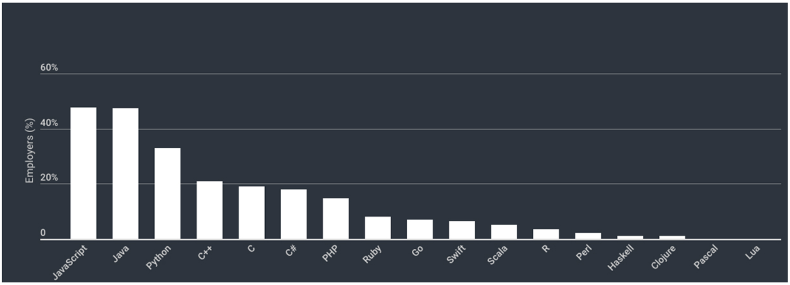
We recommend that companies add machine learning and data science courses to their technical training curriculum. Even if they're not actively using them in production, chances are they will be soon, and developers will enjoy diving into this new programming world.

The solid five: JavaScript, Python, Java, C, & C#

There are a variety of indexes out there that track the popularity of programming languages. TIOBE, Redmonk, and PYPL are all good indices, and they generally track close to each other. Our research revealed that JavaScript, Python, Java, C, and C# are the most popular languages among software developers.

HackerRank 2018 Developer Skills Report found that:

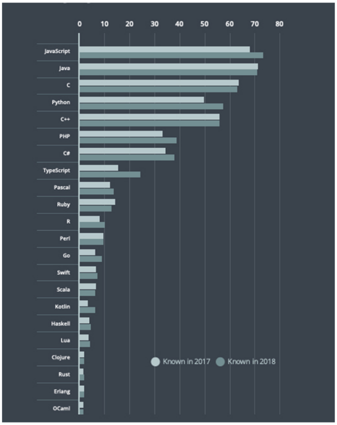
Which language do employers look for by industry?



Source: 2018 Developer Skills Report by HackerRank

JavaScript continued to gain momentum in 2018:

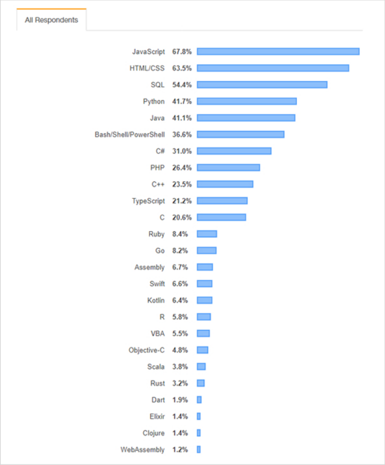
Languages known in 2017 vs 2018



Source: 2019 Developer Skills Report by HackerRank

Here's the StackOverflow's results:

Programming, Scripting, and Markup Languages

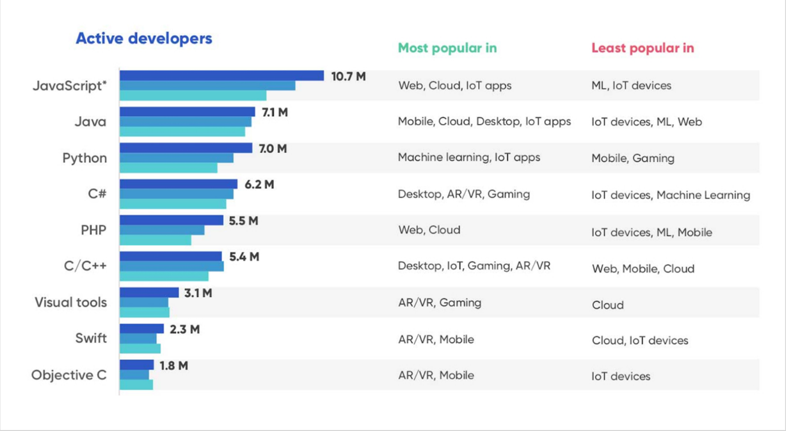


Source: StackOverflow 2019 Developer Survey Results

Developer Nation's report found that:

JavaScript, Python & PHP are growing the fastest

Number of active software developers, globally in millions, (Q2 2018 n=13,652)

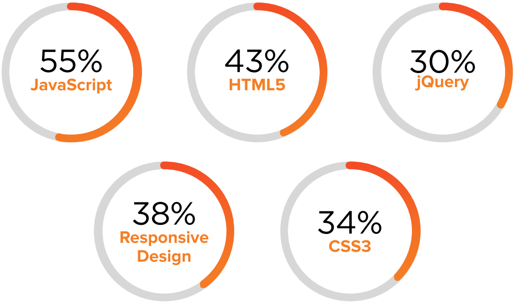


	Active developers	Most popular in	Least popular in
JavaScript*	10.7 M	Web, Cloud, IoT apps	ML, IoT devices
Java	7.1 M	Mobile, Cloud, Desktop, IoT apps	IoT devices, ML, Web
Python	7.0 M	Machine learning, IoT apps	Mobile, Gaming
C#	6.2 M	Desktop, AR/VR, Gaming	IoT devices, Machine Learning
PHP	5.5 M	Web, Cloud	IoT devices, ML, Mobile
C/C++	5.4 M	Desktop, IoT, Gaming, AR/VR	Web, Mobile, Cloud
Visual tools	3.1 M	AR/VR, Gaming	Cloud
Swift	2.3 M	AR/VR, Mobile	Cloud, IoT devices
Objective C	1.8 M	AR/VR, Mobile	IoT devices

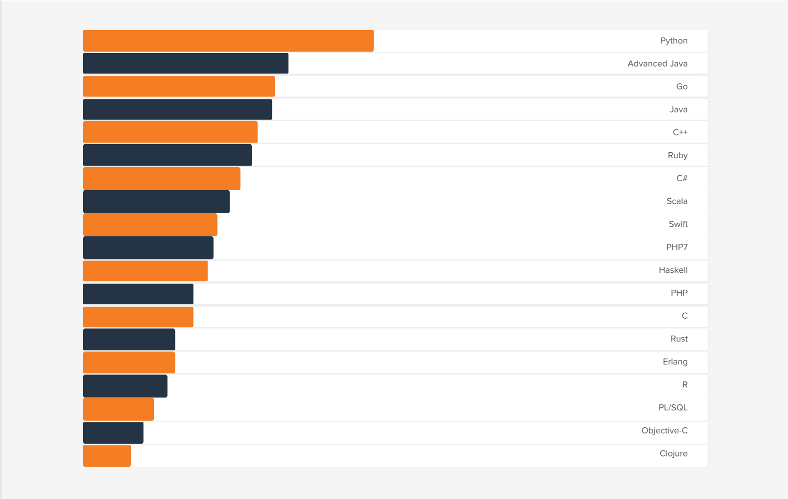
Source: Developer Economics: State of the Developer Nation 16th Edition

As you can see, there's quite a bit of overlap. PHP also made the top five on many lists. In 2019, StackOverflow declared that "Python is the fastest-growing major programming language today".

Finally, here's what our 2018 developer survey found for both front-end skills and general programming languages:



Training Needed by Programming Language



Source: DevelopIntelligence 2018 Developer Learning Survey Report

Why are these programming languages so popular?

JavaScript is used primarily for web development but also used for desktop software (see Electron) and Internet of Things development. Python is the Swiss army knife of programming languages and is used for front and back-end development. It has increasingly come to dominate machine learning, data science, and scientific computing. Java has been one of the most popular programming languages for several decades now and is widely used in government and finance. C has also been popular for several decades. The lower-level language enables high performance application development like the Linux kernel, word processors, hardware, and satellites. Because of its widespread applicability and use, some developers say that “C powers the world.” C# and .NET were developed by Microsoft and are used by sites like StackExchange, OpenTable, HealthScout, CarbonMade, and DocPlanner.

These five languages are unlikely to go anywhere any time soon. It’s likely that developers are either already using them, would like to learn how to use them, or want to advance their current programming skill set in one or more of these languages. Therefore, we recommend that companies prioritize training their technical talent here.

The new five: Go, Kotlin, Rust, Scala, and Swift

It's human nature to want to try the shiny new tool in your industry. Developers, perhaps more than other professions, really enjoy trying new technology, particularly new languages and frameworks. Our research indicates developers are most interested in learning Go, Kotlin, Rust, Scala, and Swift. These five technologies are the new kids on the block – over the last 5-10 years anyway. Go and Swift were designed by teams at Google and Apple, respectively. Rust, Scala, and Kotlin were designed by industry players that include Mozilla, JetBrains, and the École Polytechnique Fédérale de Lausanne, a French university.

All of these languages can take the best of what exists in languages like Java, C, and JavaScript, and then make them better, faster, and easier to use.

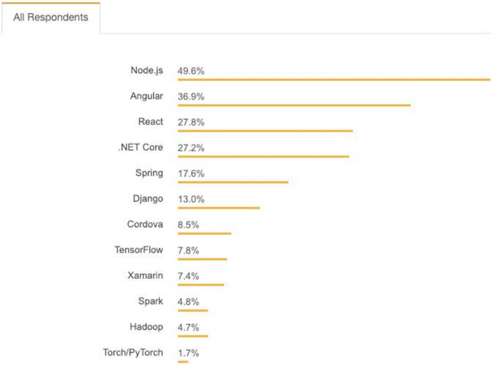
Go is quickly gaining traction and use in systems design, networking, and server development. Kotlin is now supported by Google for Android development and will be increasingly used instead of Java. Rust is used to develop systems software and backends. Scala can do everything that Java does, but with much cleaner syntax. Swift, like Kotlin, is being used for iPhone app development and quickly eclipsed Objective-C, the older way to build iOS apps, in popularity.

It's very likely that software developers at your company will be interested in at least one of these. Even if you're not actively using them in production, your developers likely would appreciate training on these languages. Further, you might consider how their use could enhance your current operations. Who knows what product and service innovations the business might reap as a result?

The JavaScript five: Node, React, Angular, TypeScript, and Vue

Given the enormous popularity of JavaScript, we wanted to dedicate some space to what we call the JavaScript five, the top JavaScript-based technologies that developers use or would like to use. The Stack Overflow report shows that Node, Angular, and React are the favorite libraries/frameworks in use today.

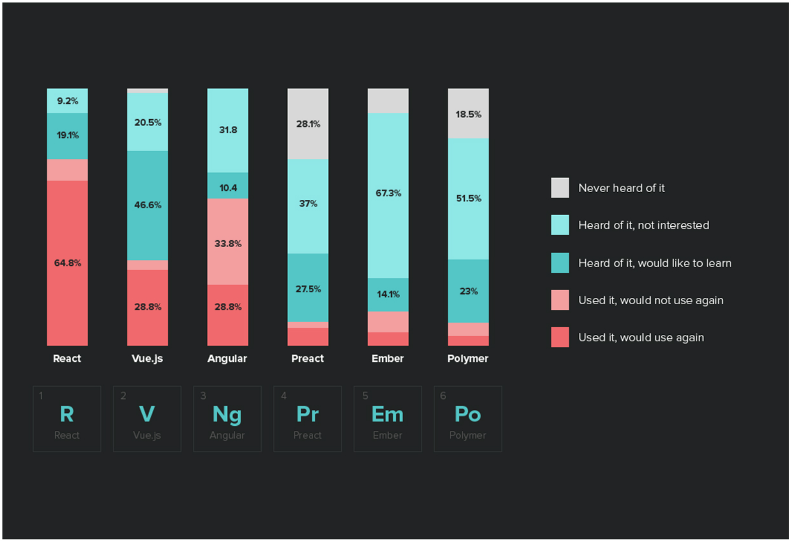
Frameworks, Libraries, And Tools



Source: StackOverflow 2018 Developer Survey Results

Node has become the de facto web application environment. If you'd like to learn more about Node specifically, in May 2018 The Linux Foundation published an extensive report on Node usage. It offers a ton of interesting node-related insights.

The State of JavaScript 2018 report is a great resource to understand how developers see and compare the top JavaScript libraries/frameworks. They have a very smart way to assess familiarity, satisfaction and use for different technologies; they ask developers for ratings: "I've never heard of it," "I've heard of it, and am not interested." "I've heard of it, and would like to learn it." "I've used it before, and would not use it again," or "I've used it before, and would use it again."



Source: https://stateofjs.com/

You could conclude several things from this data. First, React is king. In 2019, React is the primary way most teams will build front-end JavaScript applications. All of the survey reports corroborate React's popularity with developers.

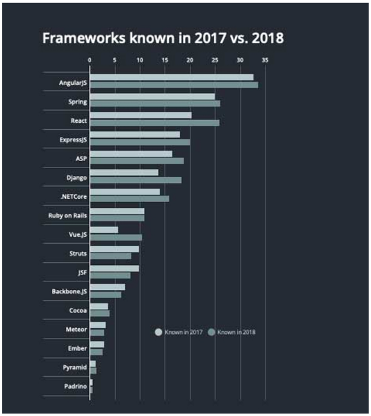
HackerRank went as far to say that React will overtake AngularJS in 2019:

React poised to overtake AngularJS in 2019

AngularJS continues to be the most popular framework that developers know. However, more and more developers have begun to learn React. The percentage to developers who know React jumped from 20% in 2017 to 26% in 2018.

Angular, on the other hand, is far less loved by developers. Note the salmon-colored 33.8% in the graphic above. One third have tried it and would not like to again.

There is one important caveat to consider, many developers tried AngularJS (Angular 1) sometime in 2012-2015 when it was very popular. AngularJS is easy to make basic apps, quickly and efficiently, however, it's quite difficult to scale to larger datasets with greater complexity. Maintaining or extending AngularJS codebases was often difficult as well. However, Angular – a framework without the JS – has evolved quite a bit since AngularJS. There have been five major releases since Angular 1. Current Angular 7 code is not compatible with Angular 1 code; they are very different frameworks. Angular as a brand may have lost its cool factor, but it's still widely used to build front-end applications, according to the Stack Overflow data.



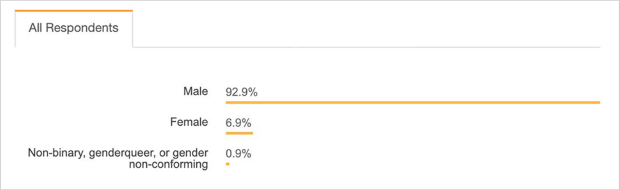
2018 Developer Skills Report by HackerRank

The other interesting point in this data is developers' continued infatuation with Vue. Vue is a library with one-way data-binding that is used to build user interfaces. Vue has a smaller user base than React but a very dedicated fanbase. Vue recently passed React in the number of Github stars. Some consider Github a vanity metric, but it's not without merit. Companies like Gitlab have written extensively about why the company uses Vue over React. The primary reason seems to be that Vue has an easier learning curve. Thus, developers can be productive more quickly. NASA, Adobe, and Nintendo are just a few large organizations that use Vue in production.

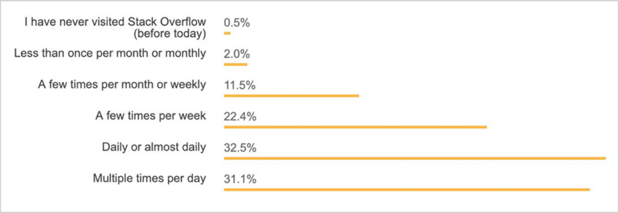
There are potential limitations in these datasets.

As much as we're fascinated by these different datasets, there are a few caveats to consider about some of them. First, the Stack Overflow report is comprised almost entirely of men under 34, without children, who use Stack Overflow everyday. Plenty of developers fit this profile, but there are also plenty of developers out there who have children and don't use Stack Overflow constantly. Their technology wants and needs are likely slightly different.

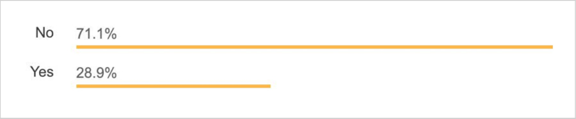
Gender



Visiting Stack Overflow



Children And Other Dependents



The same goes for the HackerRank samples. That sample of developers is actively engaged in online coding competitions. It's certainly a decent sized group, but it's still only a subset of software developers. That's why it's important to survey experienced enterprise-level developers as well, which we did in the 2018 Developer Learning Survey Report.

If your organization really wants to understand the software developers in your workforce better, we regularly help companies create surveys, and then analyze the data generated. With that data, we can help your company design the right training plan, and create course offerings that your developers will love.

Conclusion

Every developer survey report we've ever read or created emphasizes that software developers are zealous learners and self-teachers. They do this to move up in their careers, to solve problems on their jobs, to create personal projects, and to satisfy their insatiable curiosity. It's essential that organizations create work environments that keep them constantly learning. Learning and talent development is an asset to attract, empower, and retain software developers.

DevelopIntelligence has been helping organizations create talent development and developer learning programs for the last 16 years. Contact us today to learn how we can help your organization create a learning program that your developers will love.

Addendum: Where to read more about specific technologies

If you'd like to read more about these topics or specific languages and technologies, here are a few other interesting survey reports and datasets you might want to consider:

→ JetBrains State of Developer Ecosystem in 2018

JetBrains is a company famous for its IDEs (IntelliJ and WebStorm) as well as creating the Rust language. It surveyed 6,000 developers and condensed the insights into this report. This report had many questions that focused on development tools and IDE usage.

→ Gitlab 2018 Global Developer Report

Gitlabs' survey report focuses on team coordination tools, development tools and DevOps processes. This is a good report to read if you're looking to find more data on Agile, DevOps, and teamwork environments.

→ Accelerate: State of DevOps 2018

Some 1,900 developers took this survey that focuses primarily on DevOps, cloud, and open source.

→ LinkedIn 2018 Workplace Learning Trends

LinkedIn surveyed 4,000 Talent Development professionals and condensed the insights into this well-made 51-page report which focuses primarily on skill development tools and challenges talent development people face.

→ JetBrains Python Developers Survey 2017 Results

This survey report was created after surveying 9,500 Python developers. The results are focused on Python development, developer tools, and usage of different Python libraries/frameworks.

→ Stripe The Developer Coefficient

Stripe surveyed thousands of executives and developers across dozens of industries to examine how businesses leverage developer talent today, and what they could do differently. The company asked software developers many interesting questions including, "Which of the following do you believe is hindering developer productivity at your company?" It's definitely worth a skim if you manage developers.

Wrapping Up

This survey can be used to better understand general industry trends; however, in order to be ahead of the curve, managers must be aware of the ways in which their organizations align with the survey results, and how trends affect their organizations specifically.

Recommendations

Learning and Development managers need to work together to share the strategies behind their software development programs, and to determine how best to support those initiatives.

Managers should understand:

1) Which new technologies are being considered and why
2) Potential implementation timelines
3) What level of difficulty to expect while transitioning from one technology to another
4) Which major architectural changes may be on the horizon
5) Any new skills required for implementation

Begin by exploring the ways in which popular programming languages are being used, or will be used, in the organization. Are the teams primarily using Java? Is the Chief Information Officer mandating a move to web applications–a move that requires new or deeper knowledge of responsive design, mobile-application development, JSON, and HTML5? Are teams keeping up with JavaScript and front-end development trends? Are NoSQL data stores being considered? Are development teams familiar with these technologies? Begin by exploring the ways in which popular programming languages are being used.

Administer a short, anonymous survey to the development teams. Are they spending their time coding or on production and process issues?

Managers should understand:

Remember, most developers value productivity and are looking for ways to reduce repetitive tasks.

Find out which agile tools they are currently using and which tools they would like to learn. Likewise, look into current and desired build and automation tools and methodologies. Improving efficiency and productivity will help your organization thrive, and will demonstrate a clear return on training investments.

When determining learning objectives for software developer training, try writing them in an agile use case format. For example, write a statement such as: "As a [developer role], I need to be able to write a/an [application type], so that [objective] can be implemented." Writing objectives using this template should help align the team in regards to its goals, objectives, and the success criteria for training.

In short, design learning solutions as if designing a software product:

1. Identify needs, objectives, and requirements
2. Define a budget, project plan, and rollout plan
3. Identify solutions, consolidate a purchase
4. Deliver using a phased, agile rollout
5. Pivot and adjust

We Can Help

Thanks to the results of our 2019 DI Developer Survey, we recognize that the best learning occurs when it is relevant to the three **"P"s**: your people, your project, and your development platform. With all the cookie-cutter solutions available, it's a challenge to find a learning solution that fully meets the needs of your team—that's why we customize each and every one of our learning solutions to fit your people, your project, and your productivity goals.

Our goal is to continually help HR, L&D, and R&D managers better equip their development teams with the skills needed to be the best at what they do. We are adept at creating company-specific learning plans designed to meet the strategic business initiatives, as well as enable efficient and effective technology adoption, helping your company stay competitive and retain its top talent.

DI has trained more than 50,000 developers over the last 16 years. We have achieved a satisfaction rating of 98%, based on student feedback collected through post-course evaluations.

We currently offer courses in Python, Java, JavaScript, Scala, HTML5, CSS, Software Architecture, C++, Mobile Development, and much, much more.

If your organization is ready to better equip your development staff, but are unsure where to begin, we can help. Our proven process of evaluating needs, designing a learning plan, and constructing solutions will help you understand the needs of your team and ultimately deliver relevant and comprehensive solutions.

DI delivers blended learning solutions by coupling highly customized expert-led training with subscriptions to Pluralsight—a powerful combination which ensures your developers get the training they need.

If you have questions about our offerings or need help planning training for your team, give us a call at 720-445-4360 or send an email to info@developintelligence.com.