10 Predictions for Software Developers in 2018

By Siddhartha Agarwal,
Vice President, Product Management & Strategy, Oracle Cloud Platform
January 2018
Developers should be burning up with excitement about the opportunities ahead in 2018, with technologies such as blockchain, chatbots, and machine learning becoming mature enough for real-world projects. Instead, many are worried about holding up against the pressure they face: deliver code and functionality faster without compromising security or performance, since business success is so directly tied to technology innovations. For developers, 2018 will be defined by this tension between seizing transformative new opportunities while coping with the pressure to do more, with higher quality. Below are 10 developer predictions related to how those forces will play out in the year ahead.
Businesses have begun to understand the security, reliability, and most importantly efficiency from blockchain-enabled transactions. Developers will implement lots of blockchain use cases across financial services and manufacturing supply chains in the coming year. Blockchain is a technology that enables efficient, secure, immutable, trusted transactions among different organizations that don’t fully trust each other, eliminating intermediaries. Consider a company ordering products from an offshore manufacturer, which gets shipped via a shipping company, has to come through customs, through another shipping company, finally to the buyer. Today, the verification and reconciliation of each step mostly happens through emails and spreadsheets, with a lot of people and processes involved. Blockchain creates a trusted business network and eliminates manual processes and reconciliation because updates to the blockchain ledger are trusted, since it only gets updated when a minimum number of parties say, “Yes, this part of the transaction happened.”

Blockchain is a technology that enables efficient, secure, immutable, trusted transactions among different organizations that don’t fully trust each other, eliminating intermediaries.

Blockchain cloud services bring scalability, resiliency, security, and pre-built integrations with enterprise systems, which will make it much easier for developers to focus on the business use case as opposed to underlying hyperledger fabric implementation.
Chatbots routinely have real conversations with customers and employees

People are getting tired of needing multiple mobile apps to do the same job—like three different airlines apps with different ways to check in and get a boarding pass. A better way is to provide that same functionality but via the most popular app on your phone—messaging. Messaging has three attractive elements consistent across the medium: instant, expressive, and conversational—no training needed. Thanks to advances in artificial intelligence and natural language processing, people will use Facebook Messenger, Slack, WeChat, WhatsApp, or a voice assistant like Amazon Alexa or Google Home, to ask questions and get answers from intelligent bots.

Developers, using new intelligent bot-building cloud services, can quickly craft bots that understand the customer’s intent, maintain conversational state, and respond intelligently while making integration with backend systems easy. Imagine taking a picture of a dress you saw in a movie, friending your favorite clothing store’s bot, messaging the image on Facebook Messenger to the bot, which uses image recognition and AI to recommend similar style dresses, and then you pick one and purchase it. Employees could also be huge beneficiaries of bots for tasks such as asking Siri for analytics insight into data, asking how many vacation days they have left, and ordering a replacement laptop where the system already knows what laptops they are eligible for based on title and provides status updates on their order. Given it is much more forgiving to experiment with your own employee base, developers might first leverage their bot-building chops to build and test employee-facing bots.
The button disappears: AI becomes the app interface

AI becomes the UI, meaning that the pull-based/request-response model of using apps and services gradually disappears. Smartphones are still “low IQ,” because you have to pick them up, launch an application, ask for something to be done, and eventually get a response. In better-designed apps, the app initiates interactions via push notifications. Let’s take this a step further where an app, bot, or a virtual personal assistant using artificial intelligence will know what to do when, why, where, and how. And just do it. Some examples:

- Expense approvals app watches your pattern of approving expense reports, starts to auto-approve 99% of expense reports and only brings to your attention the rare report that requires your attention.

- An analytics app understands the underlying data, questions asked so far by the business user, questions that other business users in the company might have asked of the same data set, and each day provides a new insight that the analyst might not have thought of. As organizations gather more data, we might not know what questions to ask of the data.

Embedded AI can predict what you need, deliver info and functionality via the right medium at the right place and time, including before you need it, and automate many tasks you do manually today.

Developers need to figure out what data is really important to their business application, how to watch and learn from transactions, what business decisions would most benefit from this kind of proactive AI, and start experimenting. Embedded AI can predict what you need, deliver info and functionality via the right medium at the right place and time, including before you need it, and automate many tasks you do manually today.
Machine Learning (ML) is moving from the realm of obscure data science into mainstream application development, both because of the ready availability of pre-built modules in popular platforms, and because it is so useful when dealing with analysis across large, historical data sets. For ML, the most valuable insight comes with context—what you’ve done before, what questions you’ve asked, what other people are doing, what’s normal versus anomalous activity.

Developers will need to become more knowledgeable about domain specific use cases to understand what data to gather, what kinds of machine learning algorithms to apply, and what questions to ask.

But to be effective, ML needs to be tuned and trained in a domain-specific environment that includes both the data sets it will analyze and the questions it will answer. For example, ML designed to identify anomalous user behavior for a security analyst will be very different than ML designed to optimize factory robot operations, which may be very different than ML designed to do dependency-mapping of a microservices-based application.

Developers will need to become more knowledgeable about domain specific use cases to understand what data to gather, what kinds of machine learning algorithms to apply, and what questions to ask. Developers will also need to evaluate whether domain-specific packaged SaaS applications are a good fit for a given project, given the fact that large quantities of training data are required.

Using ML, developers can build intelligent and smart applications to create a recommendation system, predict outcomes, or make automated decisions.
First off, we all agree DevOps is critically important for helping developers release new applications and features fast, while maintaining high levels of quality and performance. The problem with DevOps is developers needing to spend 60% plus of their time on the Ops side of the equation, thus cutting into the Dev side where the new features get built. Developers are having to integrate various continuous integration/continuous delivery (CI/CD) tools, maintain those integrations, and constantly update the CI/CD tool chain as new technologies are released. Everyone does CI, but not too many people do CD. Developers will insist on cloud services to help the pendulum swing back in 2018 more to the Dev side. That will require more automation for real CI/CD.

Everyone does CI, but not too many people do CD. Developers will insist on cloud services to help the pendulum swing back in 2018 more to the Dev side.

Docker gives you packaging, portability, and the ability to do agile deployments. You need CD to be a part of the lifecycle of your Docker ecosystem. For example, if you are using containers, as soon as you commit a code change to Git, the default artifact built should be a Docker image with the new version of the code, the image should automatically get pushed into a Docker registry, and a container containing the image should get deployed into a dev-test environment. After QA/testing, as the container gets deployed into production, the orchestration, security, and scaling of these containers should be taken care of for you. Business leaders are putting pressure on developers to deliver new innovations faster; the DevOps model must free up more time for developers to make that possible.
Open source as-a-Service accelerates consumption of open source innovation

The open source model remains one of the best engines of innovation, which is wonderful. But implementing and maintaining that innovation is often too complex.

Developers will increasingly look for cloud services to deliver all that high-speed innovation from open source while taking care of operational and management aspects of these technologies.

- You want a streaming data/event management platform, so you turn to Kafka; but as you start leveraging Kafka at scale, you have to set up and load balance large Kafka clusters, manage their scaling, update these clusters as new releases of Kafka come out, and then integrate this service with the rest of your environment.

- You want Kubernetes for container orchestration; instead of you as the developer taking care of the master, running upgrades, backup, restores, patches for your Kubernetes cluster, the platform should do that for you. Kubernetes ships every six weeks, so the platform should have rolling deployments and self heal. Kubernetes is fast becoming the defacto application server.

- You want Cassandra for NoSQL databases; you should want the backup (incremental or full on a schedule), patching, clustering, scaling, and high availability of the Cassandra cluster to be managed by the platform.

Developers will increasingly look for cloud services to deliver all that high-speed innovation from open source while taking care of operational and management aspects of these technologies.
Serverless functions go big in production

The appeal of serverless functions is clear: when there is demand for my code to be executed based on a certain event, infrastructure is instantiated, my code deployed and run on it, and I am charged only for the time my code is actually running. Most serverless functions are implemented as microservices, invoked via events triggered in the environment with the key economic advantage of significantly reduced operating costs from only paying for what is actually used. For example, someone can build a travel booking function to book/cancel flights, hotels, and rental cars. Each of these actions can be built as a serverless function written in different languages such as Java, Ruby, JavaScript, and Python—there is no application server running with my code on it, rather these booking/cancellation functions are instantiated and executed on infrastructure when needed.

In the coming year, more and more developers will move functions out of the labs and into production.

For developers, these disconnected serverless functions strung together to execute complex transactions create new challenges: debugging distributed transactions, describing how these functions should be chained together, and on failure of one function in a chain, how to create compensating transactions to cancel inappropriate changes. Developers will look to cloud services and open source solutions to be able to easily manage the programming and lifecycle management of serverless functions. In the coming year, more and more developers will move functions out of the labs and into production. Look for open-source tools, like the FN project, to flourish by helping developers manage composition and debugging, to be able to deploy and test serverless functions on your laptop, or across any cloud. The key is going to be picking a serverless platform that provides maximum portability.
The only question about containers becomes “Why not?”

Containers will become the default for dev/test work and become commonplace for production applications. Expect continued improvements in security, manageability, orchestration, debugging, and other enterprise-scale challenges for containers, driven by open source innovations and industry standards. Containers provide the building blocks for many of the trends driving modern development—including microservices architectures, cloud-native apps, serverless functions, and DevOps.

Containers won’t make sense everywhere—for example, when you need a more prescriptive cloud platform, such as integration PaaS or a mobile PaaS—but these higher level cloud services will themselves run on containers, and will be the exceptions that prove the trend.

Containers provide the building blocks for many of the trends driving modern development—including microservices architectures, cloud-native apps, serverless functions, and DevOps.

In addition, software licensing models for high value, commercial, on-premises software will have to adapt to and embrace the spread of container adoption. Pricing models for software will have to support “turn on” and “turn off” licensing as containers are instantiated, scaled up, and scaled down.
Software and systems become self healing, self tuning, and self managing

Developers and production operations teams are drowning in the amount of data generated around logs, web/app/database performance, user experience, and configuration. In addition, these various types of data are siloed, so you have to bring many people into a room to debug issues. Developers spend a lot of time telling production ops what thresholds to set, server topologies to monitor for a transaction, etc.

With large amounts of data being generated and aggregated into one data repository (across logs, performance metrics, user experience and configuration, for example), and with lots of compute capacity, machine learning, and purpose-built algorithms, systems management cloud services will provide the next generation of performance/log/configuration monitoring. These cloud services will establish baselines for thresholds by watching transactions (saving time of ops team having to manage thresholds), and understand the server topology associated with transactions automatically. Using anomaly detection against these baselines, systems management services will automatically be able to tell developers when things are moving away from normal behavior, and be able to show the root cause of problems for a specific transaction.

Establish baselines for thresholds by watching transactions, and understand the server topology associated with transactions automatically.

Developers need to think about how to leverage this automation when writing their applications to be able to create self healing, self managing applications on top of these intelligent management systems in the cloud.
Highly-automated security and compliance efforts become a new ally of developers

Security breaches are on the rise, the bad guys are winning, and developers’ careers and projects are at risk. While developers often think of security and compliance as “someone else’s job” or “bottlenecks to delivering code,” the advent of comprehensive ML-based security and compliance regimes delivered as SaaS will allow these efforts to align with the fast pace of development. Specifically, highly-automated cyber defense will be deployed both “upstream” throughout the DevOps cycle (identifying and remediating potential security risks such as misconfigurations earlier) and “downstream” in production to automatically adapt a company’s security profile to ongoing application and environment changes (identifying attacks, remediating vulnerabilities, and assessing continuous compliance). Such protections will be required in some cases, with continuous compliance assessment a hallmark of GDPR and similar mandates. Developers, security professionals, and end-users will all benefit from a more rigorous, automated approach to security throughout the development lifecycle.