Future horizons - mapping out the next decade of testing

By Christin Wiedemann, February 2017

Abstract

Technology continues evolving at what seems like an ever-increasing speed, and we are constantly faced with new smart devices, and innovative software applications. Our use of, and reliance on, technology has grown, and we are close to being a truly connected society, collecting and exchanging astonishing amounts of data.

Reviewing the current technology landscape, it is possible to discern an underlying trend, and draw inferences about the future of testing. Big data, cybersecurity, fragmentation and the Internet of Things are areas that stand out and can be expected to have a significant impact on testing, and testers. New technology will require new skills, or new applications of existing skills. To remain relevant, testers will need to be flexible and adaptable, which necessitates continuous learning. Certain skills are likely to be more valuable, including code literacy, security literacy and data literacy.

Introduction

Big data, autonomous vehicles, and blockchains; every day we are faced with a technology change, whether it is a new adaption of something existing, or a fundamental change we would never even have been able to imagine. Our use of technology has expanded and it is now an integral part of everything we do, from managing our finances to exercising. As technology continues to evolve, the tester role, the skills required to test, and our test approaches will need to continue evolving and advancing too. To better prepare for the future, we need to identify current technology trends and consider how they will impact testing.

Mapping out the next decade of testing is a mighty task, and taking it on is probably also somewhat presumptuous of me. As a tester, I am acutely aware of how important it is to set expectations, so let us start there. Any attempt to predict the future is futile, but instead I am going to discuss current technology trends and how I think they will affect testing in the next decade. We cannot *predict* the future, but we can *influence* it. Our actions today will have an impact

Copyright © 2017 PQA Testing Ltd. All rights reserved.

tomorrow, which is why I also want to explore how we can contribute to a brighter future for testing.

Past, present and future

Physically, we all live here and now, but I think it is fair to say that, mentally, we tend to live in different eras. Some people are stuck in the past, relying on previous experiences when deciding how to handle challenges. Others are focused on the present moment, basing their actions on an analysis of the information currently at hand. Finally, there are people who are always looking to the future, trying to imagine completely new solutions that have never been conceived before. Is it better to be focused on the past, the present, or the future? I believe there is value in all three perspectives, and none of them can be left out, or ignored. The question is how much weight to put on each.

I am not questioning the importance of our pasts, but sometimes I think we focus too much on history, making ourselves blind to the opportunities of the future. An overreliance on historic data as an indicator of the future also leaves us vulnerable to change. In the software testing community, we frequently, and passionately, talk about whether there are "best practices" or not, and if there is any value to be found in industry standards. These discussions are not new, and I've heard plenty of arguments from both sides. What people refer to as best practices or industry standards are basically instructions for how to handle future events based on the past. There is certainly value in letting history influence how we deal with both the present and the future; that is called learning. But there is a line we should not cross where we go from learning from our past experiences to allowing our history to narrow our field of vision, and dictate how to handle events that are yet to come. We look back on all the products we tested on all the different projects and we create an approach that we decide is the one true way to test. We disregard new information and new contexts, and stick to our "proven" approach that is based on many years of valuable experience and extensive knowledge. It is important to learn from history rather than being driven by it.

We need to acknowledge that, even though we cannot predict the future, we still need to be prepared for it, and we cannot assume that our current approaches will apply unchanged. I believe that to succeed in the future we need to be flexible and adaptable, which in turn requires curiosity and a willingness to continue learning. The question is where to start; what new skills should we try to acquire first? To determine that, we need to take a closer look at current trends, and how we think they will impact testing. When talking about the future of testing, I think it is necessary to distinguish between technology trends and conceptual trends.

Technology trends relate to the evolution of tools, devices and applications. The introduction of smart home devices such as thermostats you can control from your phone is an example of a technology trend. Conceptual trends relate to the

Copyright © 2017 PQA Testing Ltd. All rights reserved.

evolution of our ideas and approaches, and I view the move towards having dedicated testers on a team, instead of other roles sharing the testing tasks, as a conceptual trend. Technology trends and conceptual trends can drive each other, but it is typically a loose, and not always easily discernible, coupling. Even though they are not necessarily independent we will talk about them separately, starting with technology trends.

Technology trends

Attempting to list all the technology changes we have seen just in the last year would be a Herculean, and futile, effort. Instead, I want to try to discern underlying technology trends to see what patterns emerge, and what we can infer about the future of testing. The technology trends that I think will have the biggest long-term impact on testing relate to big data and cybersecurity.

Big Data

The term "big data" has been in use since the 1990's, and the threshold for what counts as big data keeps moving as technology advances. The definition that I find to be the currently most useful is that big data is data sets larger than what can be handled by common software tools. That definition will become obsolete soon enough, as the abilities of our most common software tools expand.

Is the emergence of big data a technology trend, or a conceptual trend? There is new technology involved; we now have technology to record and store amounts of data extensive enough to count as "big". We also have the computing capacity to process and analyze big data sets. Big data is however, simply a lot of data. Big data itself involves no technological breakthrough, and in my mind, the shift is primarily in how we view data and our ability to imagine new uses for such expansive data sets. I consider the increased importance of big data as being more significant as a conceptual trend than as a technological trend. Even so, I will discuss big data as part of technology trends since I do not think I would be able to separate the conceptual aspects from the technology side.

We are still far from realizing the full potential of big data, but there are already numerous examples of its application. In science, particle physicists routinely analyze huge data samples to detect very weak signals from particle collisions. The Data Centre at CERN processes 1 petabyte (1,000 terabytes) of data every day.[1] In industry, insurance companies are collecting vast sets of personal profile data and claims data that are analyzed and used as input for predictive modelling, allowing for more accurate assessment of risk, and potentially highly personalized premiums.

Big data is the trend that I expect to have the most profound impact on testing. There are several technological challenges relating to big data, including recording, storing, analyzing and curating the data. These challenges translate into testing challenges. There are more devices than ever recording and transmitting data. We

Copyright © 2017 PQA Testing Ltd. All rights reserved.

no longer live in a world where data is only collected on desktop machines and sent to a central database in the same building. Instead we now have mobile devices, aerial instruments, wearable technology and nanosensors collecting text-based information, sound, video, and biometric data that is transmitted wirelessly to distributed databases. Testing the recording of the data will require a wider understanding of how the different devices function, but also a broader base of knowledge of different types of data. Not only is the testing itself difficult, but setting up the appropriate test environments will be arduous, if not impossible. Will test environments become a thing of the past? I am sure there are many testers reading this who have tested data integrity by comparing records. That is unlikely to be feasible on a big data set, but data integrity will still be important. Similar challenges will arise when trying to test functionality based on data analysis, and when verifying curation processes.

Machine learning

The ability to record and analyze larger sets of data has facilitated the introduction of *machine learning*, which is one of many approaches to artificial intelligence. Unfortunately, artificial intelligence is often mistakenly identified as machine learning, but they are not the same. Machine learning is one of many approaches to artificial intelligence, which aims at mimicking human behaviour. Machine learning attempts to do this by learning from large sets of data. The learning consists of building a model from sample inputs, and then making predictions based on the model.

We want to train the machine to learn a function $y=f(x)$ by giving it examples of y and x. Once the machine has learnt the function $f(x)$, it will give us predictions for future values of y when we give it new values of x. This is a *probabilistic* approach based on statistics; the machine will learn the average outcomes and automate that. We do not know what the function $f(x)$ looks like, and if we did, we would just use the function directly instead of applying machine learning. Herein lies the conundrum for us as testers; how do we test whether the prediction of y that the machine gives us is the expected, or correct, one?

I believe testing applications of machine learning, and other approaches to artificial intelligence, will require a radical change of mindset. We are used to testing systems that are *deterministic*, and can be modelled as such.[2] Unless there is an error, the same input should always generate the same output in a predictable manner. Artificial intelligence is different and we will have to develop new test approaches. I think what we develop will inevitably have to be an antithesis of how we do much of our testing today. With the introduction of artificial intelligence there are no longer clear relationships between cause and effect, and the rules we often impose on testing will no longer be valid. The other interesting question to explore further is how we integrate artificial intelligence itself in software testing?

Copyright © 2017 PQA Testing Ltd. All rights reserved.

Security

Cybersecurity has gone from being a specialized topic with a limited audience to being a constant subject of articles, blogs and news media, ranging from reports and dissections of large-scale data breaches to stories about individuals who have had their online accounts hacked. The amount of data we are sharing, and how we are sharing it, makes us vulnerable. Testers, being user and stakeholder advocates, need to design tests to identify issues that could increase that vulnerability. Traditionally, we have relied on separate teams handling all internal security concerns, including security testing, but those days are gone. I think there will still be specialized teams, using specific tools and skill sets, but I also think that security is going to need to be an aspect that all testers consider at all times. Raising awareness of security threats in the applications we test is a responsibility we will not be able to delegate.

Test design aimed at revealing security risks will most likely have to focus on data, and environments. Multi-purpose devices are convenient, but they have introduced a new type of threat. It was not that long ago that all bank transactions we had to do were done inside a bank, by a bank employee, on a bank computer running only banking software. Today we do our banking on our phones; the same phones we use to access our social networks. Applications that require a high degree of security now exist in the same software ecosystem as step counters and podcasts.

Testing applications in isolation will not be sufficient. We already know that other software running on a computer or smart device can impact the functionality and performance of the application under test, but how is the security of the application impacted? I predict that testers will need to include a security perspective in their test design, looking at both how data is stored and shared, and how the application integrates with the environment in which it will be running.

Quantum computing

Quantum computing may sound like science fiction, but quantum computers are already commercially available. On January 24, 2017, Vancouver-based company D-Wave announced that they received the first order for their latest quantum computer system, and if you are interested in buying one yourself, they are expected to ship the first quarter of 2017.[3]

Quantum computers use quantum mechanics to process information, and without getting into any of the highly fascinating details, it means that quantum computers can run a large number of calculations simultaneously. The immense capacity of quantum computers opens up new opportunities, and introduces new risks and test challenges. The potential of quantum computers is probably beyond what we can even imagine, but the immediate impact will be on cybersecurity. One thing that quantum computers can do very well is to factor large numbers, which is the basis for most cryptography today.

Copyright © 2017 PQA Testing Ltd. All rights reserved.

Unsurprisingly, the first buyer of D-Wave's quantum computer is a cybersecurity firm.

Quantum computing poses a threat to classical cryptography, but it also introduces new types of cryptography. The consequence for testing is that we need to be aware that classic cryptography can no longer be considered secure, and we must prepare for the emergence of new cryptographic protocols. Even though they are *commercially* available, quantum computers are not *commonly* available yet, and I suspect it will be quite a while before they are. Still, our security testing practices need to evolve alongside cryptography, and it is probably not a bad idea to brush up on our quantum mechanics.

Ongoing trends

Big data and cybersecurity may be the technology trends that will have the most significant influence on the future of testing, but there are additional trends that are having an impact right now, and will have an effect going forward as well. I believe these trends include fragmentation, the Internet of Things (IoT) and input controls.

Fragmentation

Version fragmentation is a problem that has been creeping up on us for a while, and it is a problem that it is becoming increasingly hard to ignore. Fragmentation refers to the fact that there are numerous different operating systems available on a plethora of significantly different hardware platforms, in particular when talking about mobile devices. As of February 2017, 97% of Android users were on one of four different versions of the Android OS,[4] which actually does not sound very bad. Unfortunately, at the same time, there were more than 24,000 distinctly different Android devices[5] already back in 2015. How do you test an application that could end up running on hundreds of thousands of different unique combinations of operating systems and hardware?

It is easy enough to agree that we cannot test all combinations, but the challenge we are going to have to address is how we select which ones we test on, and how to be smarter about our testing, rather than running the same tests on all of our chosen combinations of operating systems and hardware. I believe this will require a better understanding of the platforms themselves, and the differences between different versions of operating systems. It will also require using tools, existing as well as new ones, more extensively to increase test coverage.

The Internet of Things

The Internet of Things (IoT) is a general term that refers to the connectivity between physical devices, vehicles, sensors, etc. that enables the collection and exchange of data. There are smart phones, smart watches, fitness trackers, smart fridges, intelligent locks and numerous other devices that are no longer isolated pieces of hardware, but part of a rapidly growing network. Technology consulting firm Gartner forecasts that in 2017 we will see 8.4 billion connected devices in use.[6] Everything is

Copyright © 2017 PQA Testing Ltd. All rights reserved.

becoming connected to everything else. As previously discussed, this opens us up for security vulnerabilities, but the impact on testing goes far beyond that.

Simplicity can be surprisingly complex, which I think summarizes the challenges encountered when testing smart devices. The user interfaces tend to be minimalistic with small displays, limited configurability and few buttons or other physical input controls. Consequently, there is little room to support the user through hints or help messages, and yet usability is more important than ever. User interface constraints translate to test constraints. There is likely to be more testing that we simply cannot do through the user interface as the functionality moves from the frontend to the backend.

Furthermore, what makes these devices smart is the connectivity, but we still do not live in a seamlessly connected world. We move between networks, we switch between cellular and wifi networks, and we lose connectivity. Understanding the functionality and performance of the device under these different conditions is important, but unfortunately, the conditions can be hard to replicate in a test lab.

Controls

There are more options than ever when it comes to how we *control* the software we use. The first big step was moving from being restricted to keyboard input, to having pointing devices (computer mouse). Not too long ago, touchscreens became common place, shortly followed by voice control. The different kinds of controls at our disposal make the testing task that much harder. There are more types of controls to test, and the relationship between input and output is less straightforward. Different users press the letters on the keyboard with varying force, but with few exceptions pressing the key "K" will produce a "K". On the other hand, anyone using voice control on a mobile device will testify that saying a word or a sentence can generate a lot of different responses, most of them highly undesired. To further complicate things, emotion-detection technology is making progress, and emotion-based access control is one application. Imagine having to be in the right mood to even be able to log in to the test application.

Conceptual trends

I believe that conceptual trends are driven more by our use of, and perception of, technology, than by technology itself. Then again, it is hard to distinguish between when our usage guides technology, and when technology influences our usage.

In the last decade, we have seen software testing maturing, and getting some well-deserved recognition as a discipline of its own. At the same time as testing has been gaining its independence, it has also started to become more integrated with development activities. We are close to a point where testing and development activities are perceived as integral parts of software development, each having equal value. Using a tenuous analogy, I think software testing has gone through some

Copyright © 2017 PQA Testing Ltd. All rights reserved.

rough and slightly rebellious teenage years, temporarily left the family to go to college, and now software testing has graduated and returned home, ready to settle into the family business.

Test automation

When talking about the evolution and advancement of testing, I think that the first thing a lot of people, and organizations, think of is test automation. In people's minds, making testing better tends to equal making testing automated, as if humans are so flawed they need to be removed altogether. I do not think that is what will happen, and I do not see that as the direction in which we are really headed.

At one point, there were programmers who wrote the code and tested it for the errors they could think of. When the tester role was created, the job consisted of writing and manually executing test scripts based on requirements created by someone else. In some organizations, the introduction of quality assurance (QA) as a job title shifted the discussion from testing to including, and focusing on, quality. There were still testers executing manual tests, but they were joined by people in QA roles. Today, test automation has started to take part of the test execution away from the tester and QA roles, but these roles are still needed to do the test design and test analysis.

I do not believe the role of the manual tester will disappear, but it will change; in fact it is already changing. I see it as a trend of manual testing moving from *doing* to *thinking*. I believe that, in the next decade, we will see the execution of manual test scripts disappear, but manual testers will remain, and they will still be creating test strategies, developing test approaches, designing tests, exploring the software and analyzing test results. Their role as question askers and conveyors of information will not change, but instead it will be further emphasized.

How we view testing in general is changing, and it is affecting the *what*, the *how* and the *who* of testing. Traditionally, the *what* has been given by requirements and meticulously documented in test scripts. Manual execution of those same test scripts has been the *how*, and it has been manual testers *who* have executed the scripts. I believe we are currently already on our way to a significantly different future. The *what* will be given by cross-functional discussions facilitated by testers, and including all roles on the project. The *how* will be a complementary combination of execution of automated test scripts and manual exploratory testing. It will be a team of testers, test automation specialists and developers *who* get the work done by close collaboration and sharing of ideas and knowledge.

The world is continuously both expanding and contracting. It is expanding as new technology provides new opportunities, and allows us previously unimaginable growth, as individuals and as a society. It is also contracting, making distances seem smaller, as technology makes it possible to stay connected across vast geographical spaces and to experience

Copyright © 2017 PQA Testing Ltd. All rights reserved.

places and cultures without physically going somewhere. It makes me think that maybe the biggest conceptual trend in software testing is a similar simultaneous expansion and contraction.

I anticipate that our roles as testers will expand as the current boundaries of the role expand and blur. I expect there to be fewer constraints on what we *can* do, what we are *expected* to do, and what we see as part of our job. Maybe job descriptions will be tied directly to individuals rather than to roles. At the same time, I think the distances that often exist between roles will shrink, and people both in the same, and different, roles will work more closely together. I realize I am putting my rose-coloured glasses on here, picturing an expansion of mindset through the removal of constraints. Maybe it is a trend I want to see, rather than a trend that is actually there.

Moving from stability to chaos

Taking inspiration from the Cynefin framework,[7] I think testing started out in a *simple* state where there was an expected result for each test or test step. The relationship between cause and effect was straightforward, and test approaches were based on best practices that were often treated as rules. Then software development got increasingly involved, and testing started requiring more analysis and more expertise. There was still a relationship between cause and effect, but understanding it required investigation and exploration. There could be more than one right answer. We moved into a *complicated* state, which is where I think we are now. The introduction of big data and artificial intelligence has started pushing us out of the complicated state and into a *complex* or even *chaotic* state. The relationship between cause and effect is dissolving, and can at best only be deduced in retrospect. Unless we are willing to give up certainty and embrace ambiguity, I believe we will fail.

New skills

Future software testing trends are likely to require new skills, or new applications of existing skills, but rather than trying to guess what the most important new skill next year will be, we need to stay curious and continue learning. I do not anticipate the pace of change in the technology industry will slow down in the next ten years, which means we will have to continue picking up new skills. If we are to remain relevant, we need to be flexible and adaptable, which in turn means that we cannot stop evolving. With that being said, I do think there are certain skills that will be more valuable than others in the next decade of software testing.

Even though I think there is always going to be manual testing, I also think that *programming skills* are always going to be an asset for a tester. How deep those skills need to be is a different story altogether. At a minimum, I think all testers need to have *code literacy*, which I define as the ability to read code and write simple statements. Out of the two, I think being able to read code is even more important than being able to write it. Code literacy provides testers and

Copyright © 2017 PQA Testing Ltd. All rights reserved.

developers with a common language, facilitating increased communication and collaboration. This is true even if the language the tester is familiar with is not the product code language.

The primary purpose of code literacy is not to write test automation code, but to be able to read and analyze production and test code with the objective of designing better manual and automated tests. Code literacy gives breadth to a tester's skill set, providing a foundation that makes it easier for testers to pair with developers and to contribute beyond the traditional tester tasks.

The next set of skills I think testers should focus on evolve around data and *data literacy*. Even in today's software testing, testers need to have a basic understanding of data structures and data manipulation, but I do not think that will be enough for much longer. The amount of data we use, and how we use it, is changing rapidly. Data literacy needs to include a deeper understanding of data analytics, statistics and distributed databases such as blockchains.

Finally, I believe that testers will also need a higher degree of *security literacy*. Incorporating security risks into all test design requires a solid comprehension of application and data security. Security literacy will allow testers to find potential security issues early in the development process, without relying on specific tools. The unique position of the tester as a communicator also enables them to spread an increased security awareness and understanding to users and business stakeholders.

To summarize, I think the most important skills for testers to focus on are:

- Code literacy
- Data literacy
- Security literacy

I have listed them in the order I think they need to be addressed. Code literacy is the foundation, and is needed to be able to improve data and security literacy. Since security literacy is closely related to what data we store, and how we store and access it, I think it is important to increase your data literacy before tackling security literacy.

Outlook

How can we be prepared when we do not know exactly what we are preparing for? The answer is adaptability. No matter what the future holds, we will need to adapt to be successful. New technology will require new skills, or new application of skills we already have. We need to be modern renaissance testers; continuously learning and honing our skills. The context might change, but our tester skills will remain valuable and essential, and recognizing that will be pivotal for the success of testers, and testing.

If I were to create my own Testing Manifesto for the Future, it would read something like this:

- Questions over answers
- Learning over lecturing
- Assistance over assessment
- Curiosity over fear

Copyright © 2017 PQA Testing Ltd. All rights reserved.

Fortunately, we cannot predict the future, but technology and hence testing will assuredly continue changing. The best way to prepare is to open our minds and embrace the changes.

> *"Your assumptions are your windows on the world. Scrub them off every once in a while, or the light won't come in."*
>
> — Isaac Asimov

Copyright © 2017 PQA Testing Ltd. All rights reserved.

Acknowledgement

My ideas and thoughts do not evolve in isolation; they are the product of reading, listening and discussing. I am constantly reading articles, blogs, books and magazines. I listen to the radio, podcasts and people. I discuss my ideas with anyone who is willing to take the time to give me feedback and input. In writing up my thoughts on the future of testing I am especially grateful for the insights provided by Sherry Heinze and Jim Peers. Thank you.

---

[1] CERN Computing, https://home.cern/about/computing

[2] Machine learning can be deterministic, but for the sake of discussion I decided to focus on probabilistic machine learning.

[3] D-Wave press release, https://www.dwavesys.com/press-releases/d-wave%C2%A0announces%C2%A0d-wave-2000q-quantum-computer-and-first-system-order

[4] Wikipedia, accessed February 2017, https://en.wikipedia.org/wiki/Android_version_history

[5] VentureBeat, http://venturebeat.com/2015/08/05/fragmentation-report-there-are-now-24093-distinct-android-devices-up-78-from-last-year/

[6] Gartner press release, http://www.gartner.com/newsroom/id/3598917

[7] Cynefin framework, https://en.wikipedia.org/wiki/Cynefin_framework

Copyright © 2017 PQA Testing Ltd. All rights reserved.