



2015

How The World Tests

AN ANNUAL TESTING REPORT BY ZEPHYR



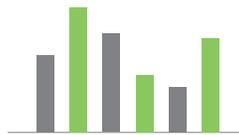
About this report

Zephyr conducts an annual survey of its 6000+ customers in 100 countries to determine where the testing industry is headed, as well as, better understand the latest trends in all things testing. The resulting data and the Analysis behind it is shared with participants and becomes a benchmark report with which testing teams and departments can measure themselves against.

Everybody dislikes filling out long and boring surveys that eventually tell you things you already knew, or present you with reams of data that you need a sabermetrics-like degree to make some sense of. We do, too. So we took it upon ourselves to design a super short and relevant survey and then analyzed the data to unearth really useful intelligence. We are now presenting this in an easy-to-read and digest format. The kind of information presented along with quantified data allows you to take action, plan your future testing initiatives, engage your upper management in quality strategy and upgrade your team's testing knowledge base. In short, it makes you look like a testing god!

Read on...

What We Cover



Who provided us with this useful data?



Mobile testing



Agile testing



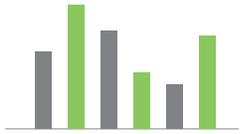
Internet of Things testing



Test environments

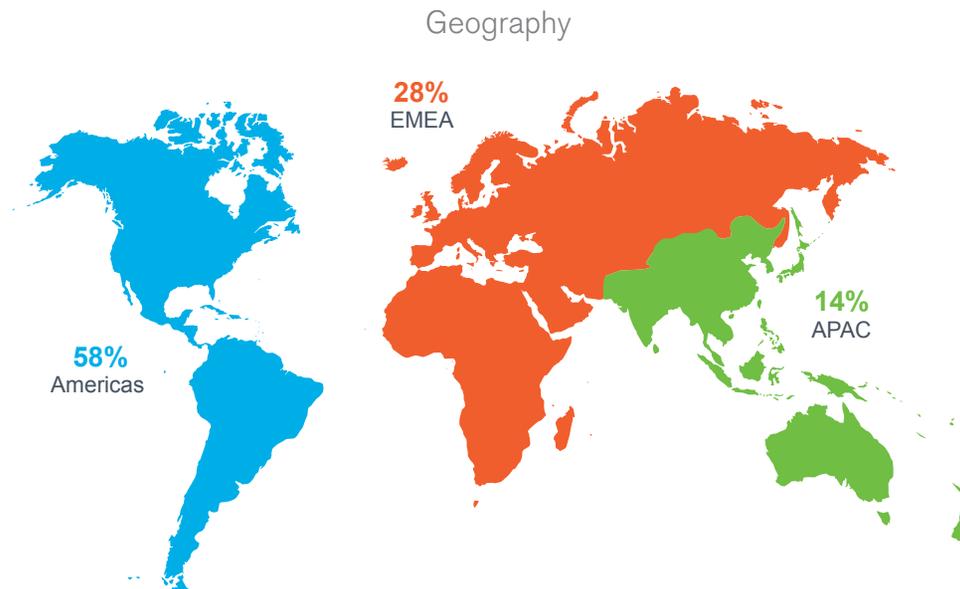


Automation



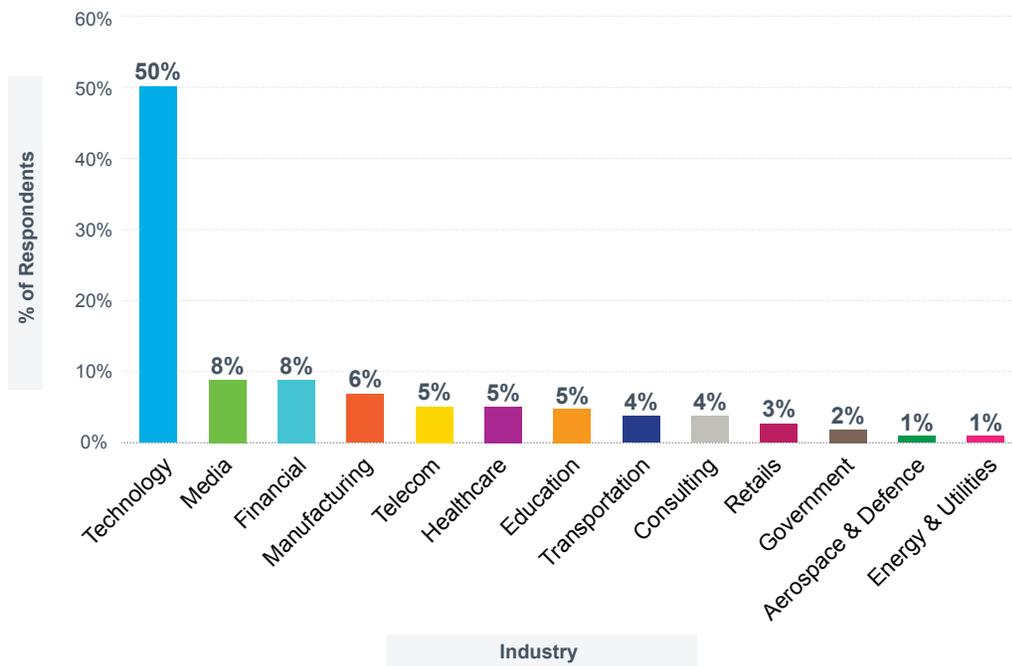
Who provided us with this useful data?

We wanted this to be a global report as testing happens on a global scale today (The world is flat and all that) and given that we have a large number of customers distributed in 100 countries around the world, that is who we focused on. We got fantastic responses from over 500 companies spanning a wide variety of industries. Over 50% of our respondents were from the Americas, with EMEA and APAC comprising 28% and 14% respectively.



Clearly, technology companies test a lot, so no surprises there, but what is interesting is how much software is disrupting other industries. Everyone from media to manufacturing, from healthcare to government, retail to defense, is building software that needs to be tested.

Industry verticals



What does this mean to me?

Well, for your next testing job, don't just limit yourself to technology companies – lots of other industries are building software and need your skill set.

What we predict:

Going out on a limb, we think this is only going to increase as more and more industries get into the act. Next year we should have a larger percentage for some of these non-technology specific industries.

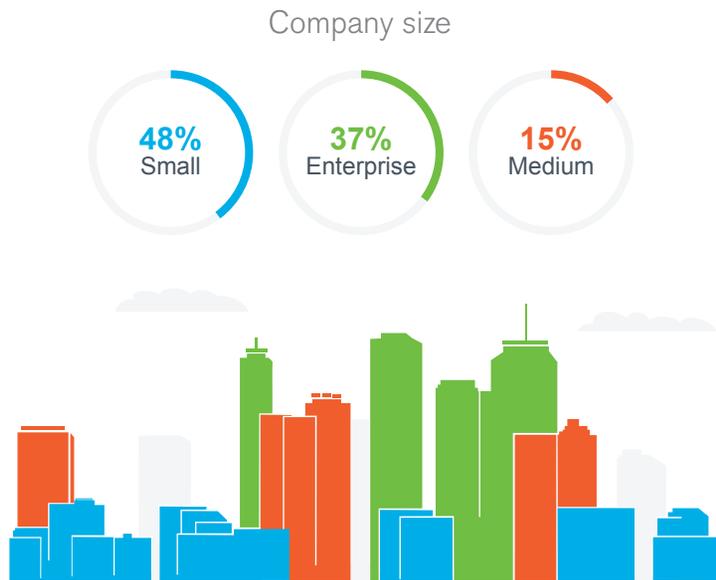
Now you may be asking these questions:

- Can we trust the data that we are about to read?
- Is this truly representative of all the companies out there?
- Who are these folks that are filling out silly surveys when they should be working?

Well, these are folks like you and us, folks who care deeply about whether they are doing all the right things when it comes to their testing department, folks who can be critical of themselves because they want to improve. These are folks up and down the chain of command who are responsible for software quality and their testing initiatives. We think this is a pretty good mix of job titles and is very representative of the pyramid-like organization structure that most companies or test departments have.



It also seems to be fairly evenly dispersed among small, medium and large enterprises, so thankfully the data is not skewed towards smaller companies and their world or larger companies and their issues.



What does this mean to me?

You should be able to identify with the analysis you are about to read. It should be fairly representative of who you are.



Agile testing

Aah, Agile testing. It means so many different things to so many people. Sometimes we think this is the biggest issue facing testing teams today (besides the fact that we need to automate more, but more on that later). Agile development processes have been around for almost a decade now and in the last few years we have seen the push for Agile increasing across the board. But is the testing team able to adopt Agile? Are they able to transform their testing to a more Agile way? What is holding this up? What challenges are they facing?

Some of what you are about to read might make you scratch your head (we know we did) and some of it might be something you are already experiencing. Either way, it is quite eye-opening. And don't miss the list of actual comments a little further in the section – you are bound to feel like your office was bugged and we somehow heard what was going on in your world!

Really, how Agile are you?

This was one of the most important questions we asked. We needed folks to be completely honest about this and we got some pretty interesting data. While everywhere you turn, its Agile-this and Agile-that, the actual data is quite shocking. Less than 60% of the respondents were halfway Agile! And less than 30% were completely Agile. This is after Agile having been in our midst for 10 years now. Living through it on a daily basis, you might say that this is accurate, but we were seriously hoping that the industry as a whole would have made a lot more progress.

Clearly Agile is a journey. We at Zephyr keep saying it's "a transformational journey" and this data is proving that. A lot of companies have a long way to go to become Agile in the way they approach their testing.

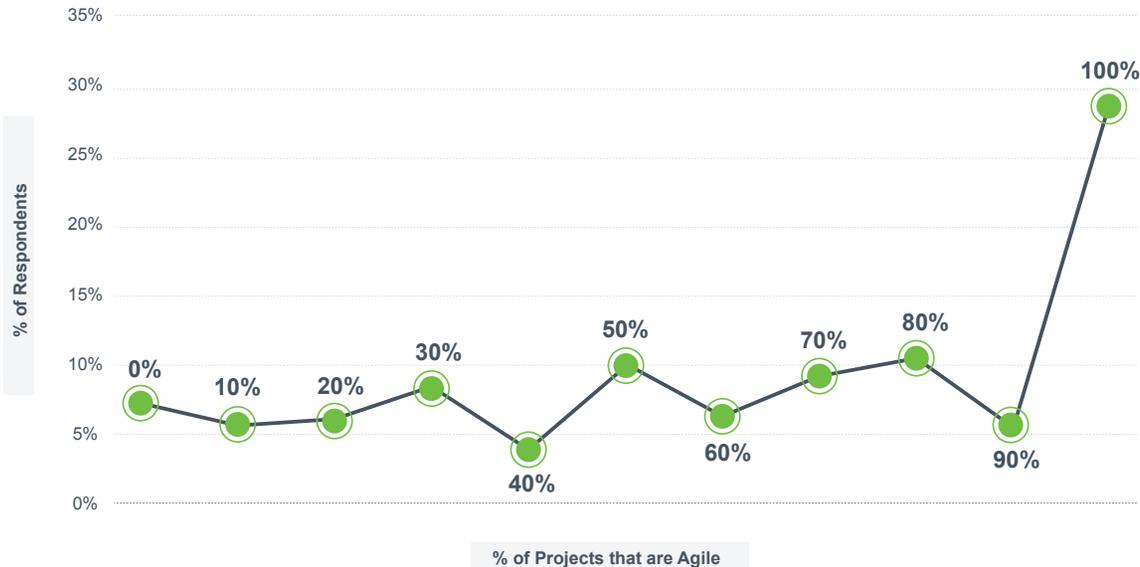
What does this mean to me?

If you are in the group of below 50% Agile, you have some work to do as you embark upon your transformative journey. If you are in the higher percentages, then read on, you might see what issues are holding you up from becoming completely Agile. Either way, you are not alone as many others are in the same boat. Clearly, there is no silver bullet here.

What we predict:

The drum beat for more Agile testing is growing — there seem to be better tools out there (though adoption is slow) and people genuinely want to make that change, so they are educating themselves on how to achieve this.

Q1: How Agile are you?

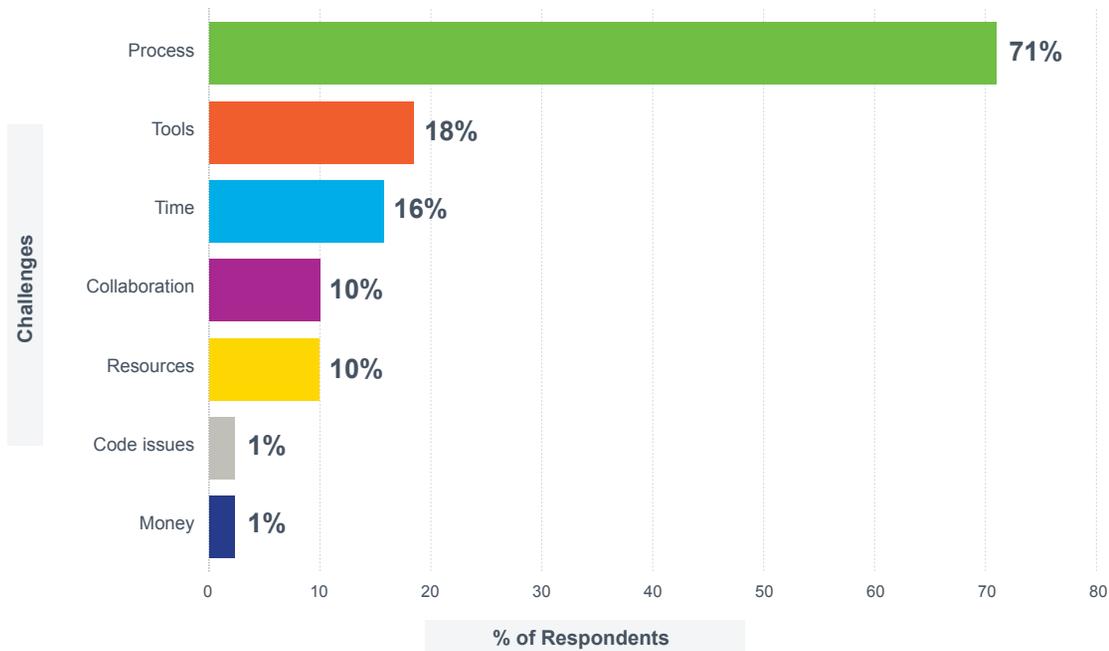


So, what is holding you back from becoming Agile?

We really wanted to know. And by far, the biggest issue, was *Process*. Over 70% of the respondents said that: not having process, not having it right or having a process but not being able to implement it, was what was holding them back. A more fundamental question we would like to ask is: whether you even know what process is needed to become Agile? The testing team – to a large extent – has to follow the overall Agile processes being put together for the entire project team and are really not able to dictate process. Does that then mean that adoption is low because it was thrust down their throats? Or is there a genuine endemic problem here? Either way, this is huge and surprising.

Besides process, we found lack of appropriate *Tools* come in second. In a way, that is tied to the process. If you don't have the right processes in place, how could you expect your tools to help you become successful since they are probably not aligned with those processes? And sometimes it's just that you have legacy tools that are impeding the adoption or rollout of Agile processes.

Biggest challenges with Agile testing



Frankly, we thought *Time* would be higher. Making that transition in your software release cycles from having enough time to complete a big chunk of your testing to completely reorienting your cycles around two and three week sprints generates a lot of angst. So, why aren't more people complaining about it? Possibly because the lack of process is seen as the bigger issue than not having enough time to complete testing and folks are just resigned to the fact that they cannot get through all their testing. A next level question we should have asked would have been: "Do you cut corners to complete your testing in time, as per the laid out schedule?", but we didn't. Ask yourself that and be brutally honest with your response.

Collaboration has dramatically increased, that is always a good thing, and we seem to have enough resources, as clearly, that is not a big problem testing teams have.

What does this mean to me?

Focus on this part. Clearly, you are not alone in this, but it seems like processes are the biggest issue holding folks up. So it's important for you to take a very critical view of your current ones; You need to find the root cause(s) for why they are not being implemented or working well, drum up enough support within your project and make changes happen. Be that person who brings about change.

What we predict:

Teams will figure out a process that works for them and adopt it eventually. It will probably not happen overnight, but tools will help. Having more team members trained (see later in this report) will help and the natural evolution of project team decisioning – fix something that doesn't work well – will make this better.

Does this sound like your world?

Here are some very interesting comments we got when we asked what were the three most pressing challenges in adopting Agile (a bunch of them sound very familiar, don't they?):

"Development teams are Agile, testing teams are not.... Access to environments, Testing too late (are our major issues)"

"Write and run test cases within the sprints, Have enough time to run test before sprint ends and Regressions"

"1) Training new users; 2) setting up the project workflow to make sure all aspects are covered correctly; 3) changing existing mindsets to convert from waterfall."

"There is a lack of understanding that the related QA effort is more long term, and that tests span across sprints and span across releases."

"Short time between releases (too short for manually doing full regression tests), incomplete implementation of Agile principles"

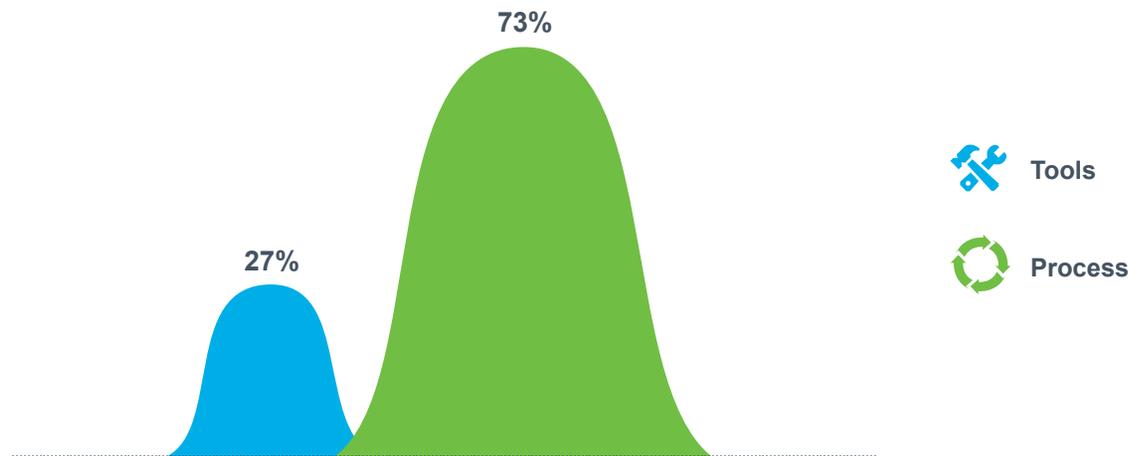
"Making accurate estimates, Marketing trying to sneak in features, and ambiguous acceptance criteria"

"1. Agile at times (read mostly) is referred to take shortcuts and avoid documentation. 2. Unless a proper structure is achieved starting from Business to IT, we'll be looking at a probable downfall of quality. 3. Consistent stream of work needs to be established prior for investing into specific expertise/setup."

"1.) Ad Hoc/Urgent needs or projects that come from the many leadership groups we work with. 2.) Leadership not sharing the full vision up front, team commits to a date, we receive MANY extra requirements and still need to make the due date 3.) Quality Standards understanding among all team members"

In summary, this chart tells us that (overwhelmingly) processes have to be fixed first before anything else.

What is more important to you when it comes to adopting Agile testing?



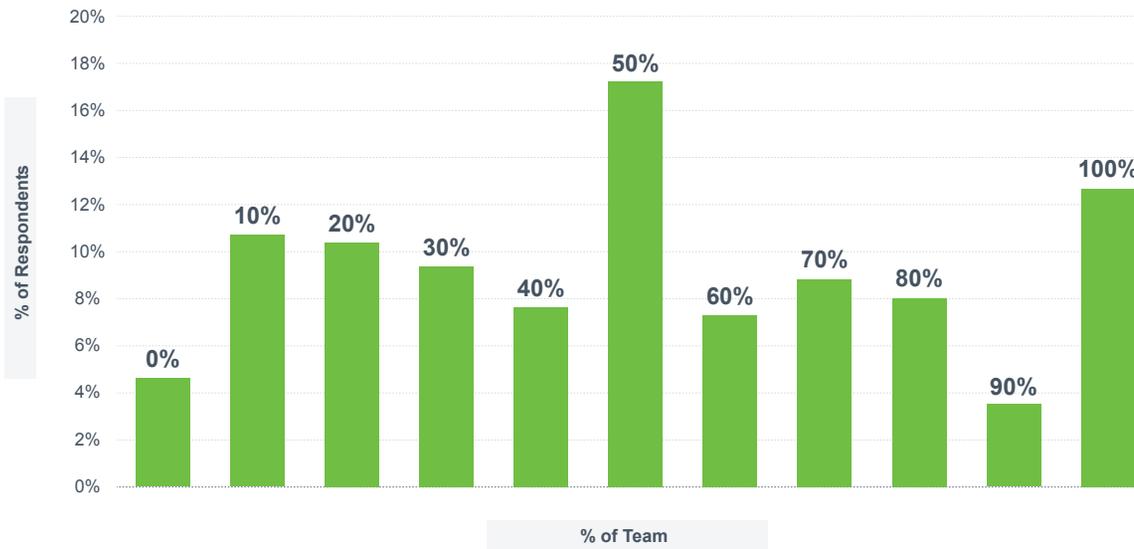
Is this a training problem?

Will training your people in better Agile testing techniques help increase your Agile adoption rate?

The data indicates that over 50% of folks think that their teams are not all fully acquainted with knowledge, to implement Agile testing successfully into their projects. So this is clearly an area of self-improvement. But why is that? In a way, this is not all that surprising when you realize that the same testing team has to test both new and legacy software products and services, where legacy projects tend to be more waterfall-based and the newer ones are more Agile-oriented. So in a way, there is bound to be confusion in the team especially as you are making that transformative journey to Agile. Processes have to be different, tools are different, techniques are different, mindsets have to be changed, etc.

Till then, when someone asks you what sort of testing process is being used in your department/company, the truthful answer is probably “Agile-ish”.

What part of your team is reasonably acquainted (or better) with Agile testing techniques?



What does this mean to me?

Some key changes should come your way. Take it upon yourself to very clearly understand (and document if your crazy schedule allows you to!) what your Agile testing techniques are. And if you don't have them, seek help. Train yourself and your team. Alongside process changes we talked about earlier, this will make a tremendous difference in your journey to better adopt Agile.

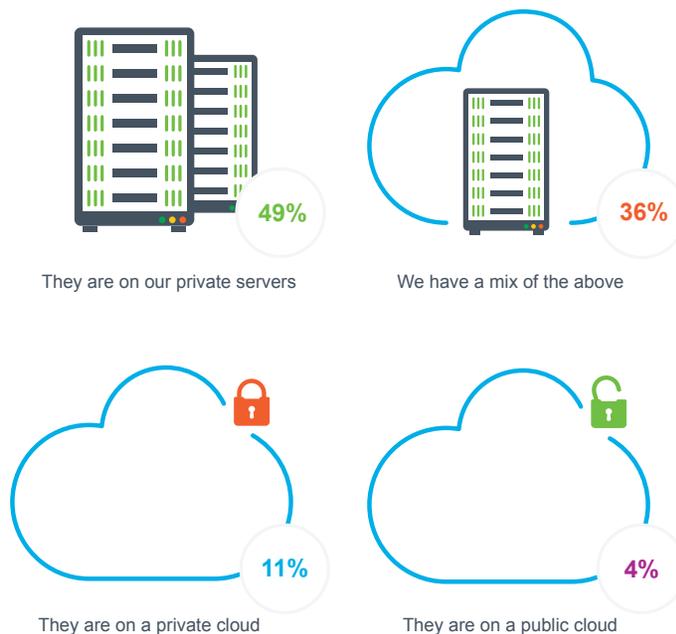


Test environments

The forecast for test environments is definitely cloudy. But today, the reality is that it's not all that cloudy, *yet*. With the meteoric rise of cloud adoption over the last five years, we were expecting that kind of adoption to also happen in the testing world. It would make sense, right? Unlimited instances that could be turned on/off whenever needed, more comprehensive test beds, dedicated 1-1 test environments for testers, dramatically reduced cost of using and maintaining them – all of these reasons would make us think that cloud-based test environments would be the rage. Not so much.

Almost 50% said they were still using internal test environments, with just 15% using either public or private clouds. That is quite surprising, this being 2015. The benefits of using cloud-based test environments far outweigh the need to continue using internal test environments, so we definitely see this changing going forward. What is gratifying to see is that a third of the respondents have both or a mixture, which tells us that the tide is slowly starting to turn.

Where are your test environments located?

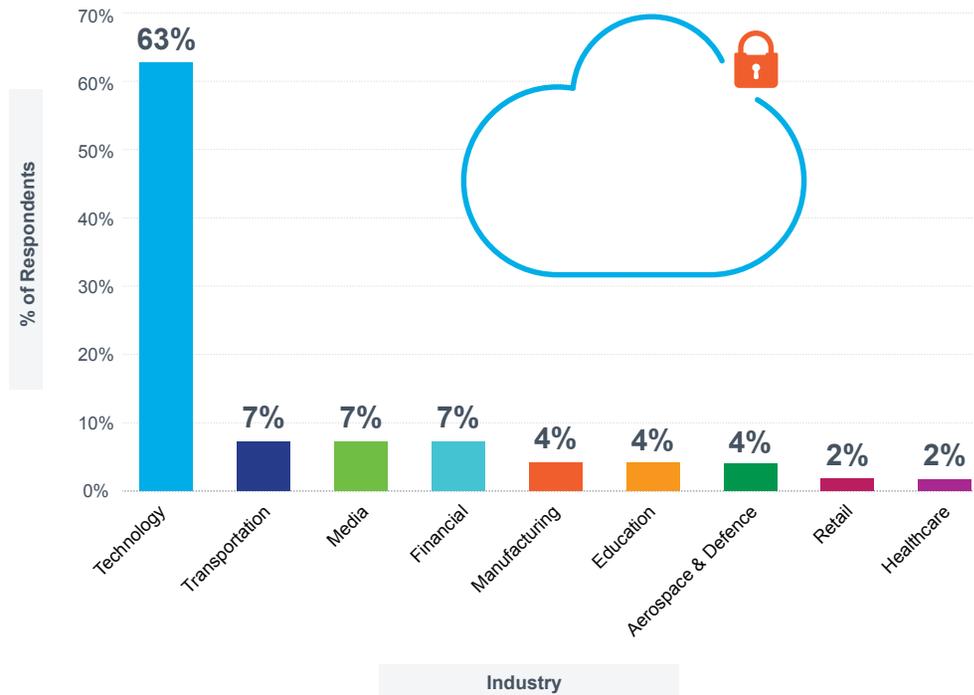


What does this mean to me?

If you are among those 50% or so that haven't made the move to cloud-based test environments, you should. And you need to do it quickly. Saving costs is just one aspect of it (and something that will make you quite popular with upper management!), but having access to a more comprehensive test bed so that you can expand your testing, being able to increase your automation footprint, decreasing the impact of environment setup time on your testing schedules and a host of other reasons behooves you to act on this quickly.

This next part isn't very surprising but shows a tremendous area of growth. Which industries are more prone to using the cloud? Clearly technology-focused companies are early adopters, be it internal or external clouds, but we can foresee the other industries starting to adopt more cloud-based test environments over the next few years.

Our test environment is on a private cloud



Another insight we gleaned was that the Americas were more amenable to adopting public cloud-based test environments while EMEA and APAC had a very tiny percentage.

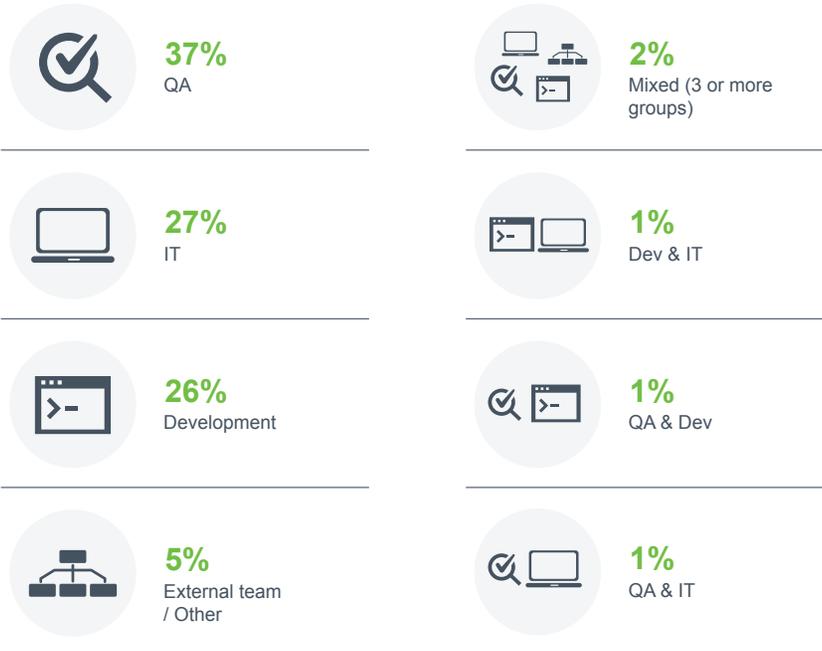
What we predict:

Public and private cloud test environments are a nascent area, but one that will grow substantially in the next few years as more and more testing departments start getting a handle on implementing their cloud strategy. APAC will see tremendous growth in this area given the large population of testing teams in that region.

Who manages your test environments?

While it seems that a third of the time test environments are managed by the QA team, it also means that two-thirds of the time it is not. As you can imagine, this leads to dependencies on other teams like IT and Dev to get your test environments ready for you to test - which means a schedule impact and potentially a coverage impact. While more complex environment setups might not be within the realm of the QA team's capabilities, having access to (and the ability to manage) test instances is an important area for a team to consider in speeding up testing, increasing testing efficiency by allowing every team member to have their own test beds and not allowing the lack of test environments or know-how slow down testing.

Who manages your test environments?



What does this mean to me?

If you are currently not managing some or all of your test environments, start now. Start small if you want to by signing up with a public cloud provider and setting up a few instances with various flavors of OSs, browsers and databases (you'll be amazed at how easy some of this stuff is to set up these days) and hand them to your testers. You'll see adoption grow quickly and its impact on your testing schedules will be measurable. Welcome to the cloud.

What we predict:

While IT and Dev will always have some say in the installation/setting up of test environments, we expect QA teams to start taking more control and managing their own testing world a lot more adeptly. The plethora of such cloud providers and easy management tools will definitely help and the dramatic reduction in cost can only but make this a priority item for testing teams.

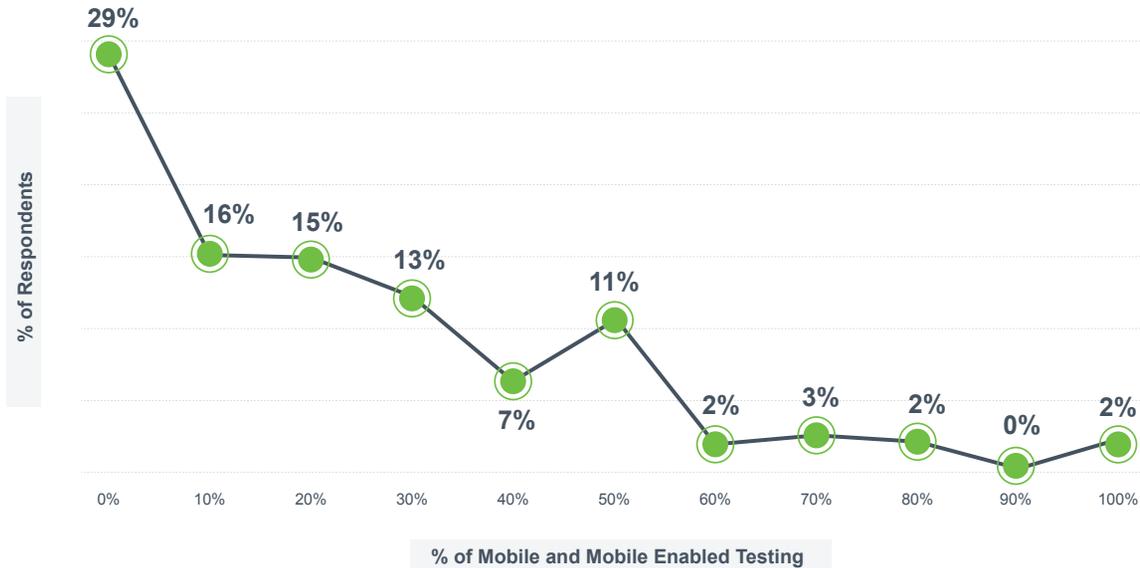


Mobile testing

Something is off here. We are a little confused, but numbers don't lie. With all things mobile being so "hot" these days, you would think that there would be a lot of mobile related testing going on. After all, there are millions of apps out there, right? Well, less than 10% of the testing population spends more than half their time testing mobile apps and mobile-enabled applications. About 30% of them have no mobile testing whatsoever. When we thought about this a little more, we realized it could mean one or a combination of these things – mobile testing is still not a priority, mobile development and its tools have evolved that it needs less testing (ha!), mobile apps are popular but not as much as everyone would like to believe, mobile testing is just too hard, or something else we can't put a finger on.

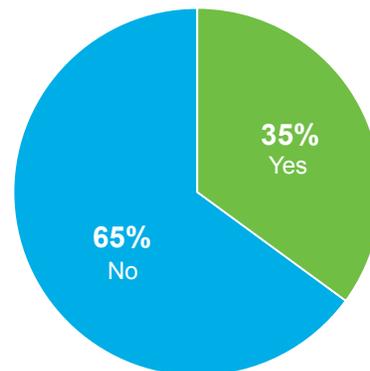
So ask yourself this – what is it in your world that puts you in the bottom 50% of this graph?

What portion of your testing deals with mobile apps and mobile-enabled applications?



Looking at this graph, clearly there is a lot of room for improvement and increasing adoption of mobile testing tools.

When it comes to mobile apps and mobile-enabled applications, do you have the right tools?



What does this mean to me?

Time to start researching mobile tools and figuring out what works best in your world. There is definitely a significant portion of testing teams – that could include yours – that do not have the right mobile tools. Yet. So go change that and make your mobile testing a lot more effective.

What we predict:

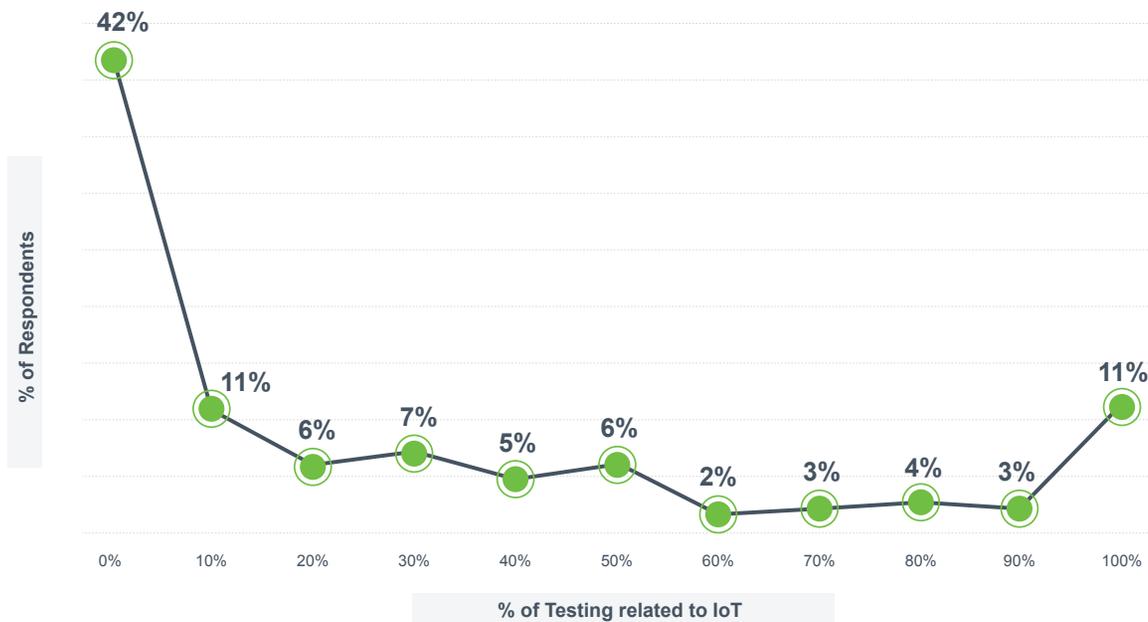
Mobile testing has to get better – with the advent of many mobile testing tools, easier environments and tracking and teams gaining experience, we expect a larger percentage of a testing team's time being spent on mobile apps and mobile-enabled app testing. We cannot expect customers to continue bearing the burden of being our guinea pigs.



Internet of Things testing

The arguments go both ways – IoT always existed but didn't have a fancy moniker OR IoT is a brand new category of always connected devices and “things”. Either way, we are surprised at how much testing for IoT is going on. See for yourself.

What percent of your testing has to do with the Internet of Things?



Almost 60% of the testing population had some testing going on that was related to IoT. Small amounts of testing, we agree, but nevertheless an important trend. And 10-25% had half to all of their testing focused on this space. If we take the side of the argument that claims this is a new category, then this is super-fast adoption from a testing perspective. And if we side with the other camp that it has always been around and is just getting a new fancy name, then 42% hasn't been doing much with it and that is statistically significant.

What we predict:

Fad or trend, new moniker or same-old, same-old, IoT needs testing and we expect that to grow significantly into the next few years. The industry will benefit from better tools, newer testing techniques as testing teams will be challenged and hopefully higher quality always-connected products.

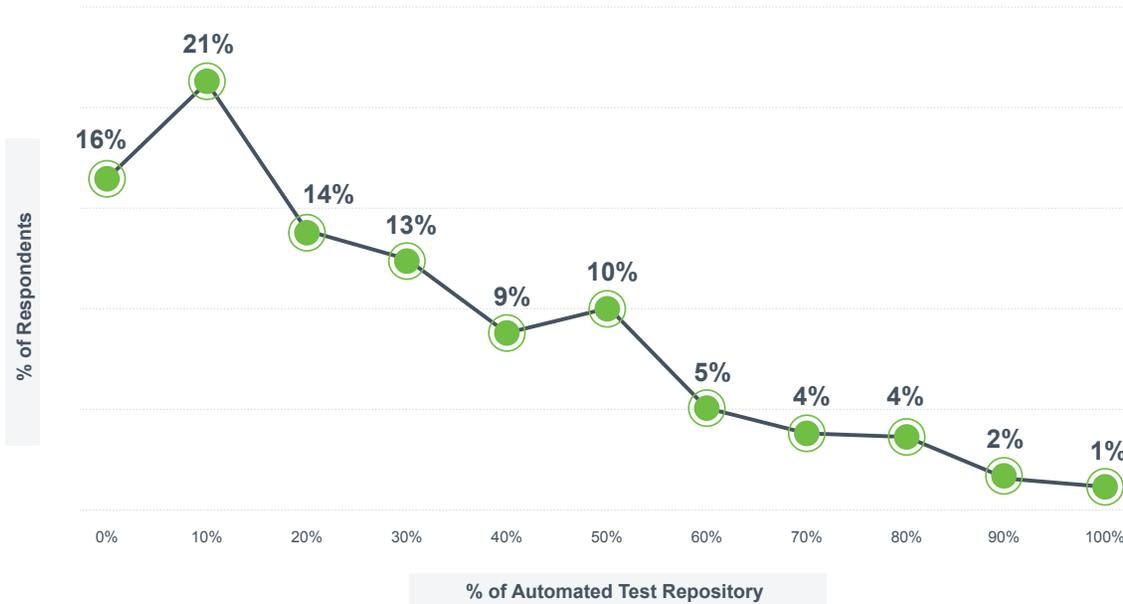


Automation

Everyone's favorite subject. So we are not going to spend time here debating the pros and cons of test automation, how to do it best, is it really worth the time and effort, etc. Let us just look at some hard data. Let us ask the question (and be brutally honest here) about how much of your testing repository is automated.

The answers, as much as we would not like to admit, are both realistic and sad.

What portion of your test repository is automated?



We just can't seem to get this completely figured out. As a mature industry, the testing world has grappled with this problem for a long, long time. We all agree that automation is the answer, we should automate more (for all the good reasons), and we have a ton of commercial, open-source and homegrown automation tools. Yet, the percentage of companies that have more than half their repositories automated are a measly 16%. In other words, 7 out of 8 companies are not even halfway there. And a whopping 37% have little or no automation. We all agree that automation is tough to do (again, a myriad of reasons exist, you probably have a long list of your own) but this is the reality. And it is sad.

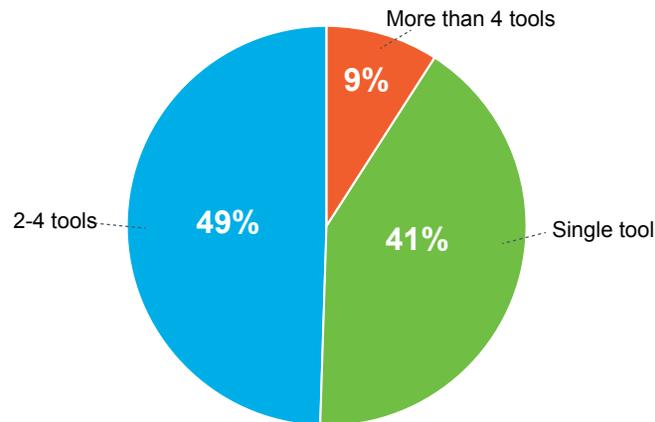
What does this mean to me?

This will remain one of your top 3 challenges. And there are no easy answers here. Automation is a good idea and bringing it to life, making it really effective in your constantly changing world of new technologies, Agile cycles, unsupported tools, and release pressures will become your biggest achievement in your current role. The thing is, you are not alone.

So can tools help us get past this hump?

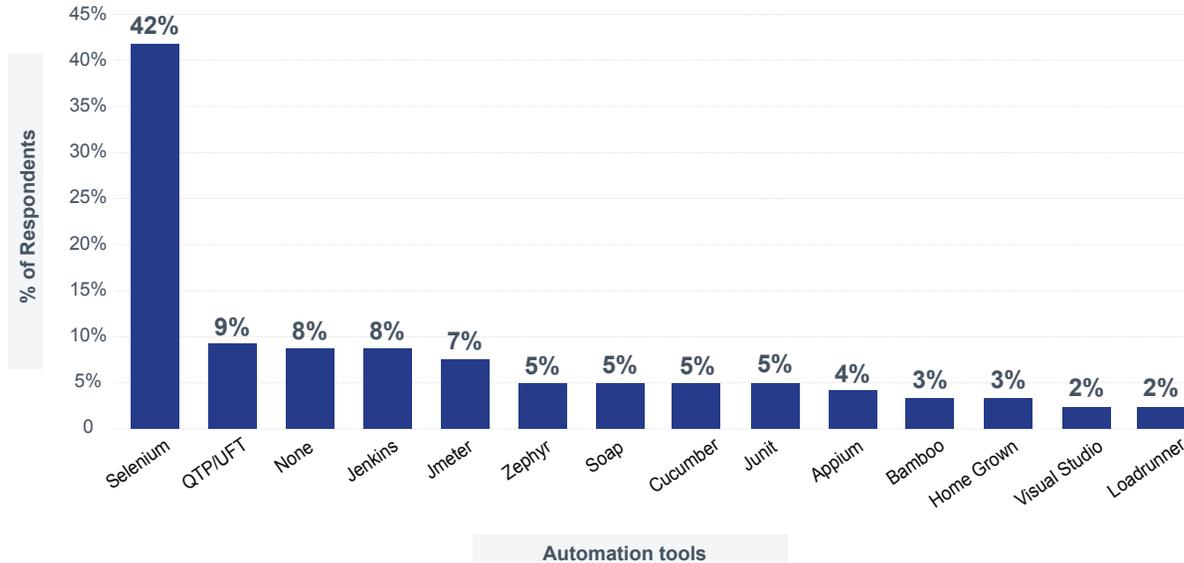
Yes and no. The number of tools are increasing to the point where a single automation tool just does not cut it anymore (a lot of reasons for that). More than 50% of the companies out there use more than one automation tool, sometimes 3 or 4 or more. Testing has become more complex over time as the sheer number of technologies and systems to test have dramatically increased. Testing tools are not just able to keep up with this rapid evolution and testing teams have to keep looking for and implementing best-of-breed solutions all the time. And with that come the challenges of evaluating, implementing, learning and maintaining these tools and their siloed repository of tests. You can quickly see how this can become a problem in trying to increase your percentage of test repository that is automated.

How many automation tools are used?



Check out this next piece of data. We all knew Selenium was popular, didn't we? It's just that it doesn't do everything and hence we need other tools too and that goes back to the same problem discussed above.

Name any automation tools you're currently using or planning to use.



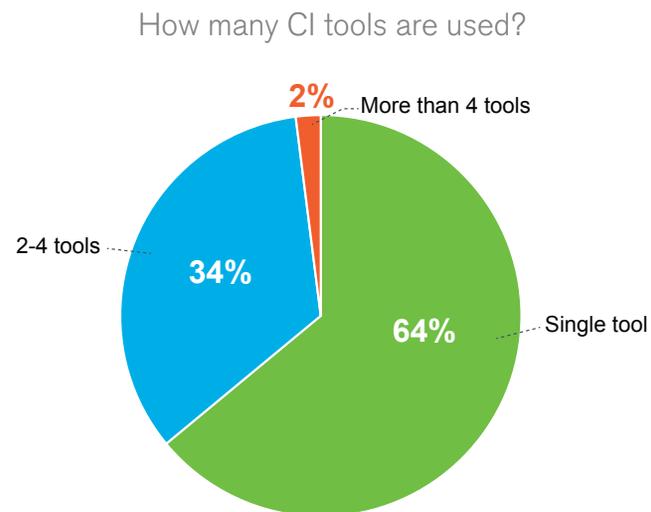
What we predict:

Nothing much will change here, there are too many automation tools. While Selenium will gain a little ground, we'll still have a bunch of other tools that will fracture the usage.

And now for some Continuous Integration

What *will* change is the adoption of CI tools in our testing toolkit. For the better.

A significant portion of project teams have adopted Continuous Integration and are using at least one tool to help them realize better upstream quality. The problem of multiple siloed tools is less apparent here as almost two-thirds seem to have standardized on a tool and are hopefully finding that effective.



Last few words

Well, there you have it. A snapshot of how the world tests in 2015. Some of this data was eye-opening, some of it was stuff we already knew, some despairing and some enervating. What this tells us though is that the testing industry is a vibrant one with a tremendous amount of activity and growth. It is evolving. It continues to be significant and it continues to strive to test and release high quality software for its users.

We hope you enjoyed reading this, found it relevant and can apply some of these learnings in your world.

See you in 2016.

Thank you

Zephyr would like to thank the 505 software quality professionals, business leaders and subject matter experts who took part in the research study this year for their time and contribution to the report.

Main Author	Samir Shah , CEO, Zephyr
Lead Researcher	Francis Adanza , VP Marketing, Zephyr
Project Management	Kristina Kamenov , Brand Advocate, Zephyr Lilian Buitrago , Brand Advocate, Zephyr
Creative Design	Robert Purcarea , tadspace.com
Proofreading	Excelon Development



About the company

Zephyr is a leading provider of quality management solutions, powering intelligent DevTestOps for more than 11,000 global customers across 100 countries. Project teams and enterprises of all sizes use Zephyr's products to enable continuous testing throughout their entire software delivery pipeline to release higher quality software, faster. Zephyr is headquartered in San Jose, CA with offices in King of Prussia, PA, Europe and India.

For more information, please visit www.getzephyr.com.