TRICENTIS

How the Future of Test Automation Affects You

Wolfgang Platz, Founder and CPO
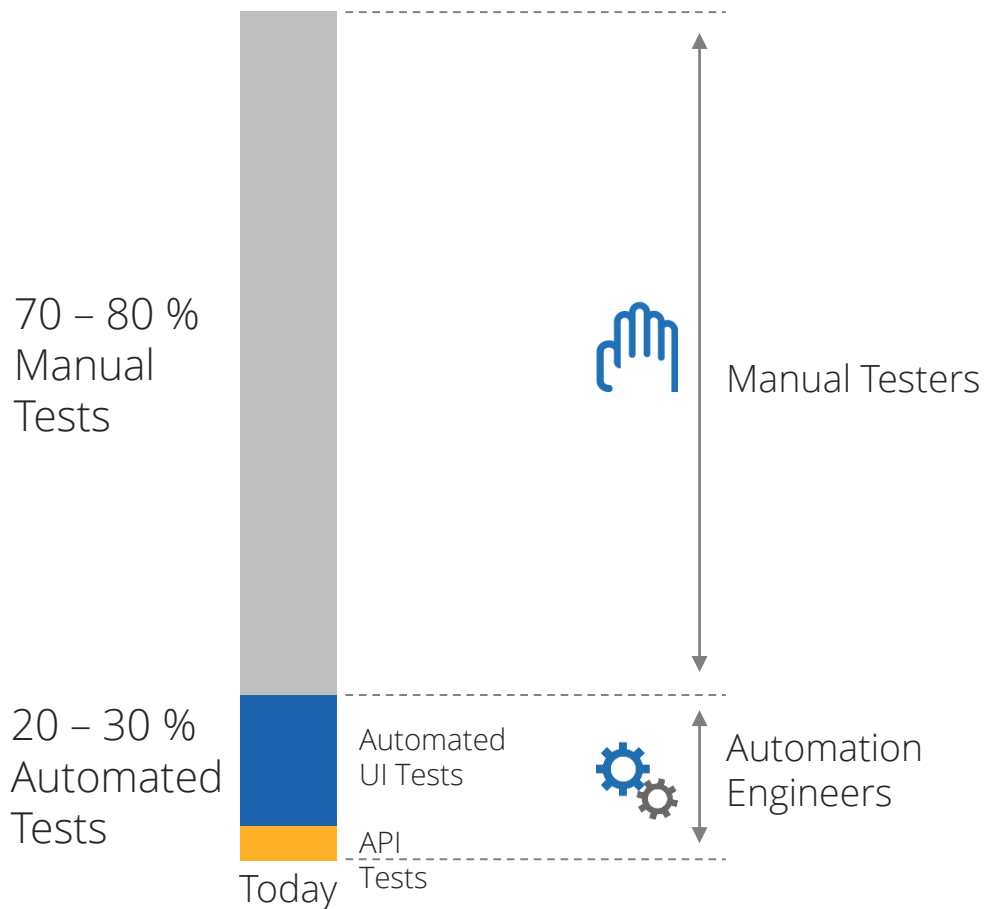
1

The Test Automation of Today

7RICENTIS

We have a problem.

Although Agile methodologies and DevOps have reduced development cycle times significantly, today's software testing is still dominated by manual tests. This is how the testing efforts of a typical enterprise looks today:

20–30%
of tests are **automated**

80–90%
of automated tests are **UI Tests**

10–20%
of automated tests are **API tests**

70 – 80 %
Manual
Tests

Manual Testers

20 – 30 %
Automated
Tests

Automated
UI Tests

Automation
Engineers

API
Tests

Today

Obviously, this testing model will not fare well once DevOps is introduced into the enterprise. Whether they like it or not, companies are being forced to adopt test automation early on in the development process just to keep up with the market demand. It is not an exaggeration to say that literally <u>all</u> of the companies we have talked to want to increase their test automation rates.

2

The Test Automation of Tomorrow

TRICENTIS

The testing landscape is
prone to seismic shifts.
Every time the
landscape changes the
question appears again:

Is software testing dead?

...two major roles typically found in enterprise testing...

Everyone wants to know whether robots have finally come to steal their jobs. It's a valid question, particularly to those who fill the two major roles typically found in enterprise testing.

Manual testers.

Manual testers define test cases and execute them manually. They identify the required test data objects and also do a fair amount of exploratory testing.
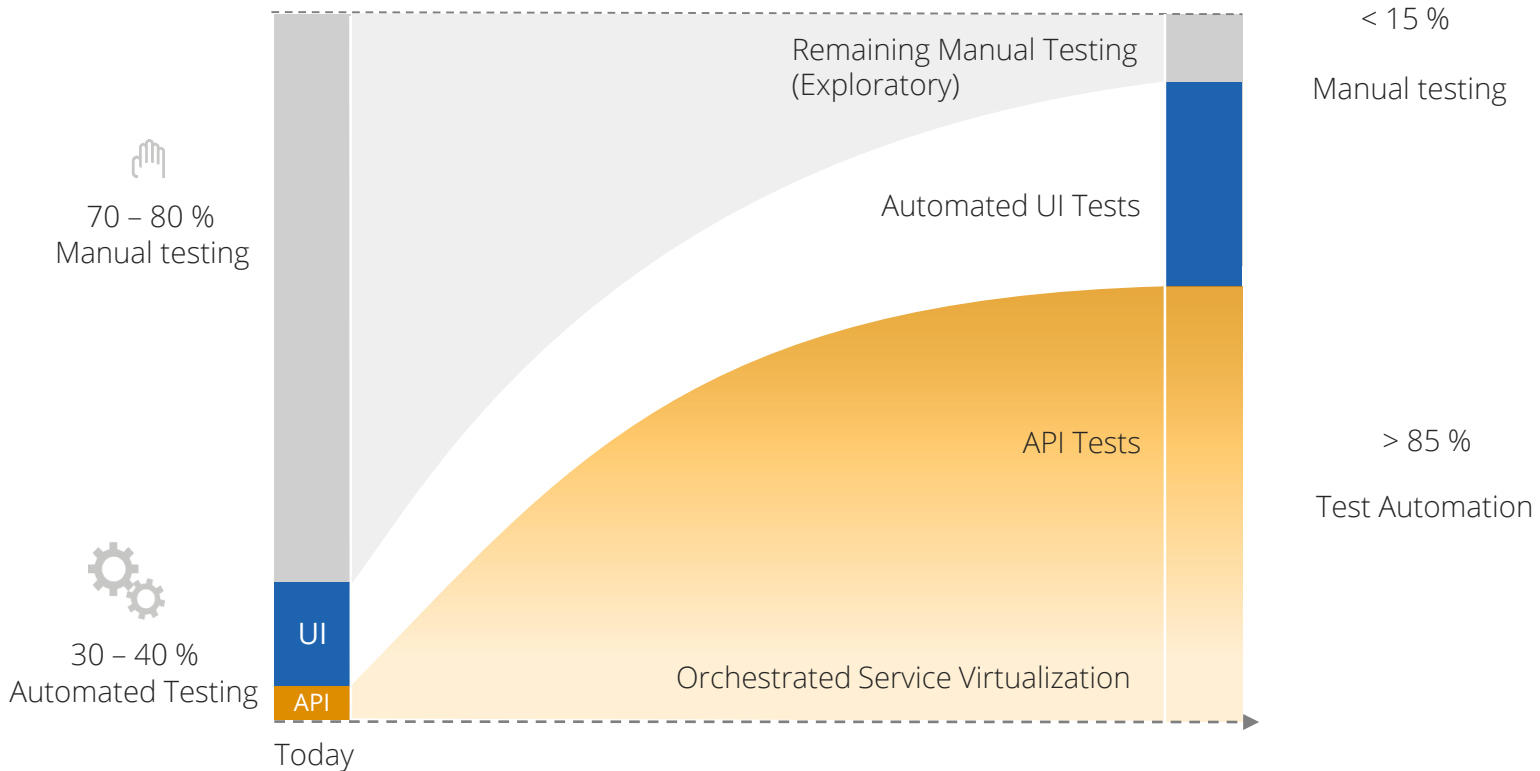
Automation engineers.

Automation engineers design sophisticated test frameworks and are skilled software developers. They automate, maintain, and run test cases.

These roles are bound to evolve with continual changes to the software landscape. As more enterprises increase automation rates, less test cases will be left for testers to run manually. As more manual testers learn to automate their tests, automation engineers will find that the focus of their work changes as well.

Manual testers and automation engineers will see their job descriptions changing – and as a result, become more valuable to their respective companies more than ever before.

Let's take a deeper look at this by first predicting how software testing will change as enterprises adopt higher rates of test automation.

70 – 80 %
Manual testing

30 – 40 %
Automated Testing

Today

Remaining Manual Testing (Exploratory)

Automated UI Tests

API Tests

Orchestrated Service Virtualization

UI

API

< 15 %

Manual testing

> 85 %

Test Automation

- API tests are much more stable and faster than UI tests. They will be the dominant method of testing apps in the future.

- Service Virtualization will play a vital role as an enabler of high test automation rates in complex and interconnected landscapes**. It will be an integral part of test automation.

What does this mean for you?

- The share of testing in IT budgets will not increase*, despite the demand for speed. Testing will become much faster and more efficient in comparison to today.

- Test automation rates will increase dramatically, simply because they *have to* in order to keep testing in parallel with the increased pace of agile development.

- Remaining manual tests will be predominantly exploratory. Automated UI tests will be more important than they are today – but they will require a smaller share of the effort in relation to full automation.

*... IT's share on overall budgets is increasing, and currently even QA budgets are increasing faster than IT budgets – but this trend will flatten (see World Quality Report 2015/16)
**... The border between API tests and Service Virtualization is not a sharp line but an area of crossover.

Assuming that these predictions are accurate, how would these changes affect today's manual testers and automation engineers?

If we have a brief look at the history of test automation, we recognize that maintenance efforts have always been the biggest threat to automated test cases: when the automated testing maintenance efforts outweigh the manual execution efforts, why stick with automation? Using a sophisticated framework eases the pain, but on top of being extremely expensive to implement and maintain, they turn test automation into a software development project –excluding business testers from the automation process.

Any successful test automation in the past has required heavy software development.

3

What It All Means For You

TRICENTIS

Does that mean that all our manual testers of today need to be replaced by "development-testers" in order to meet the rigors of Agile and DevOps practices?

To complicate things further, API testing today is purely the domain of the technically savvy. How can enterprise API testing be expanded without first hiring more developers?

Don't forget Service Virtualization! Which will soon be a fundamental requirement in complex system landscapes. Without Service Virtualization, testing environments will not be available to support the systems in Continuous Test – who will deal with that?

Is the future of testing a developer's discipline?

No.
The future of testing is no developer's discipline.

Testing will not become a developer's discipline because it simply can't.

You may be aware of how hard it is to find suitable software developer talent these days. The sheer number of developers that would be required for Continuous Testing simply does not exist on the market. Not to mention that software developers themselves are not usually too fond of testing.
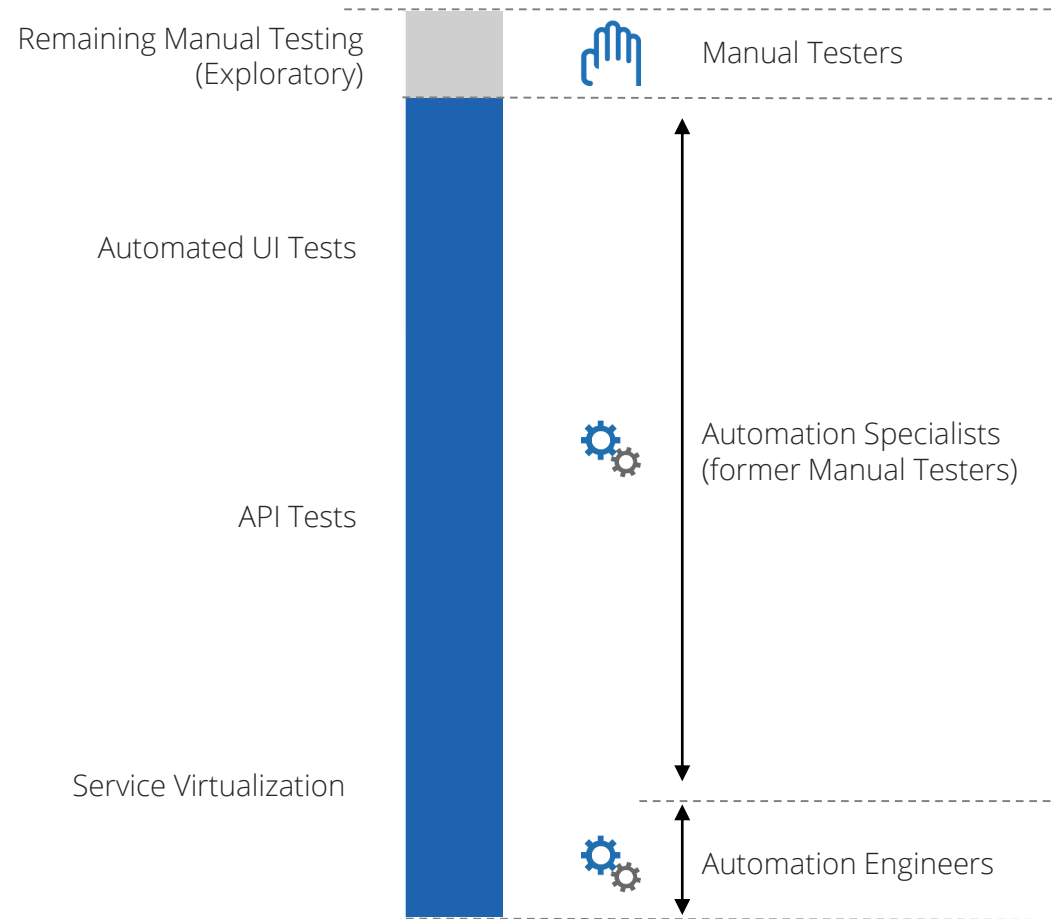
If we were to ask software developers to do 85% of the testing, the testing would either never happen, or distract developers of their primary job – which is to innovate.

The key to success doesn't lie in restructuring the testing department, but in **changing the way we leverage the tools** of software testing.

Besides solving the maintenance issue of test automation, software testing tools of the future will need to enable and encourage today's manual testers to take on a more proactive role in test automation. Better yet, these tools should enable manual testers to automate all kinds of tests: UI tests, API tests, and enable Service Virtualization.

This is a true paradigm shift:

Test automation needs to become a **business discipline**.

At Tricentis, we firmly believe that the following role model can (and has) become a reality:

Remaining Manual Testing (Exploratory)	Manual Testers
Automated UI Tests	Automation Specialists (former Manual Testers)
API Tests	
Service Virtualization	Automation Engineers

Today's manual testers are the automation specialists of tomorrow.

Once the tools allows test automation to be embraced as a business, rather than a developer's discipline, today's manual testers will shift the bulk of their work to automated testing. As a result, their job profile will be enriched and their career pushed forward: while still doing some manual testing, particularly exploratory, they will also have the capability to automate UI and API tests, as well as Service Virtualization.

The educational path to becoming an automation specialist is straightforward: after a few days of training with the testing tool, manual testers will be able to deliver automated test cases.

Today's automation engineers will find their skills leveraged.

With manual testers becoming automation specialists, will today's automation engineers lose their jobs?

No, the need for technical skills will never decline. Automation engineers will need to provide the customer specific basis to make automation specialists productive: writing custom-control steering for user-interfaces, and providing business abstractions of messages on the API and Service Virtualization level. They will even be required to ensure that object-oriented principles in the set-up of automated test cases are being applied by the automation specialists.

Automation engineers of today will find that their technical skills become all the more valuable in the future as they enable the work of the automation specialists.

Conclusion

The test automation rates of most enterprises today are way too low to enter the age of DevOps. Attempting to solve the problem by turning manual testers into savvy technicians dealing with sophisticated test automation frameworks simply does not work.

Instead, the tools of the future needs to embrace manual testers and make them productive for any type of test automation, be it UI test automation, API tests or Service Virtualization. With such a tool, manual testers will become automation specialists while automation engineers provide the foundation for automation specialists to work from.

What will be the result of this new approach to automated testing? Not only will today's manual testers and automation engineers find that they have become more valuable than ever, but that together they will achieve the highest test automation rates we have ever seen.

I have to add that this "tool of the future" actually exists right now. Making automated testing a business discipline was the dream that inspired the development of Tricentis' software testing suite, Tosca Testsuite.

Stay in touch