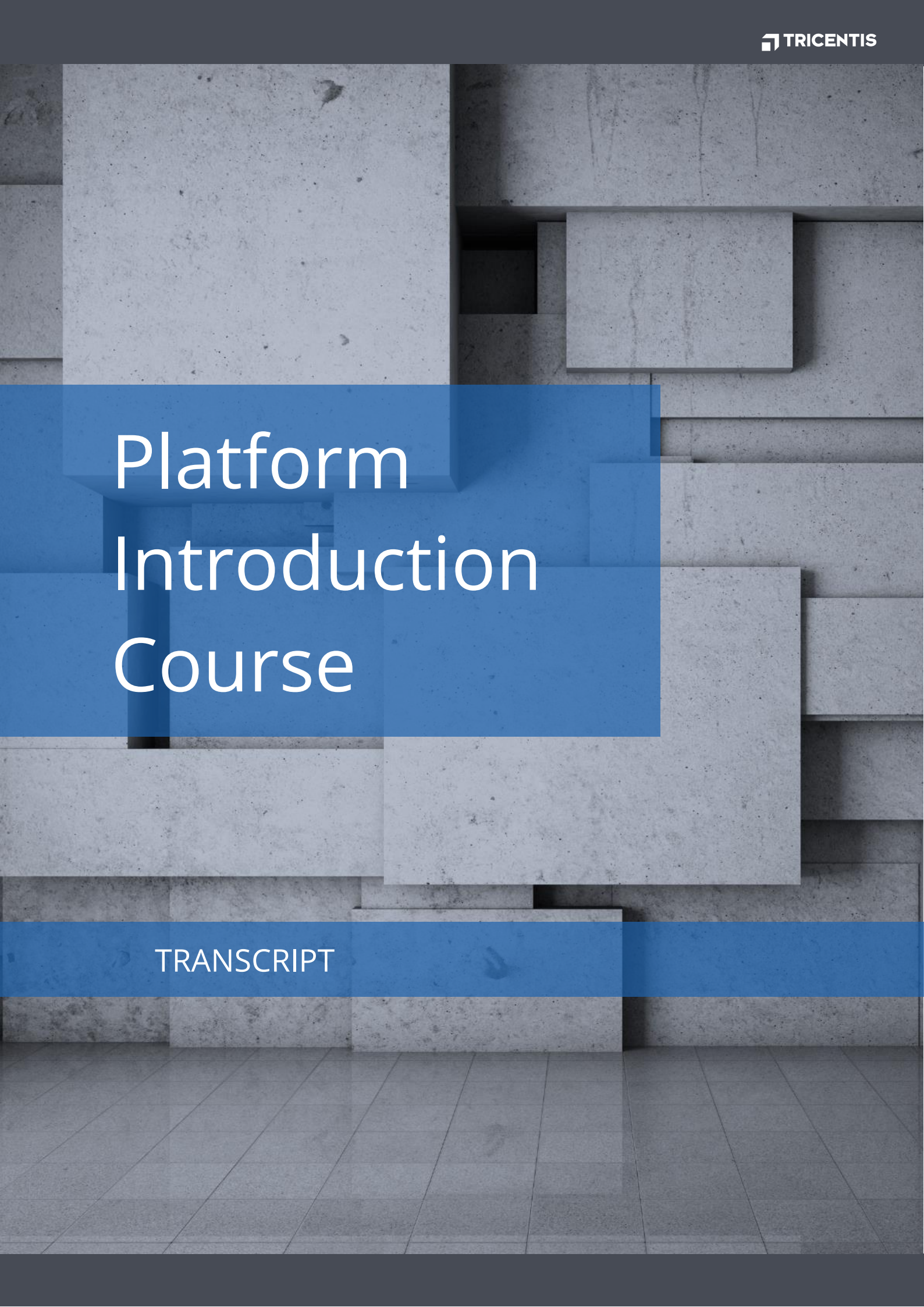
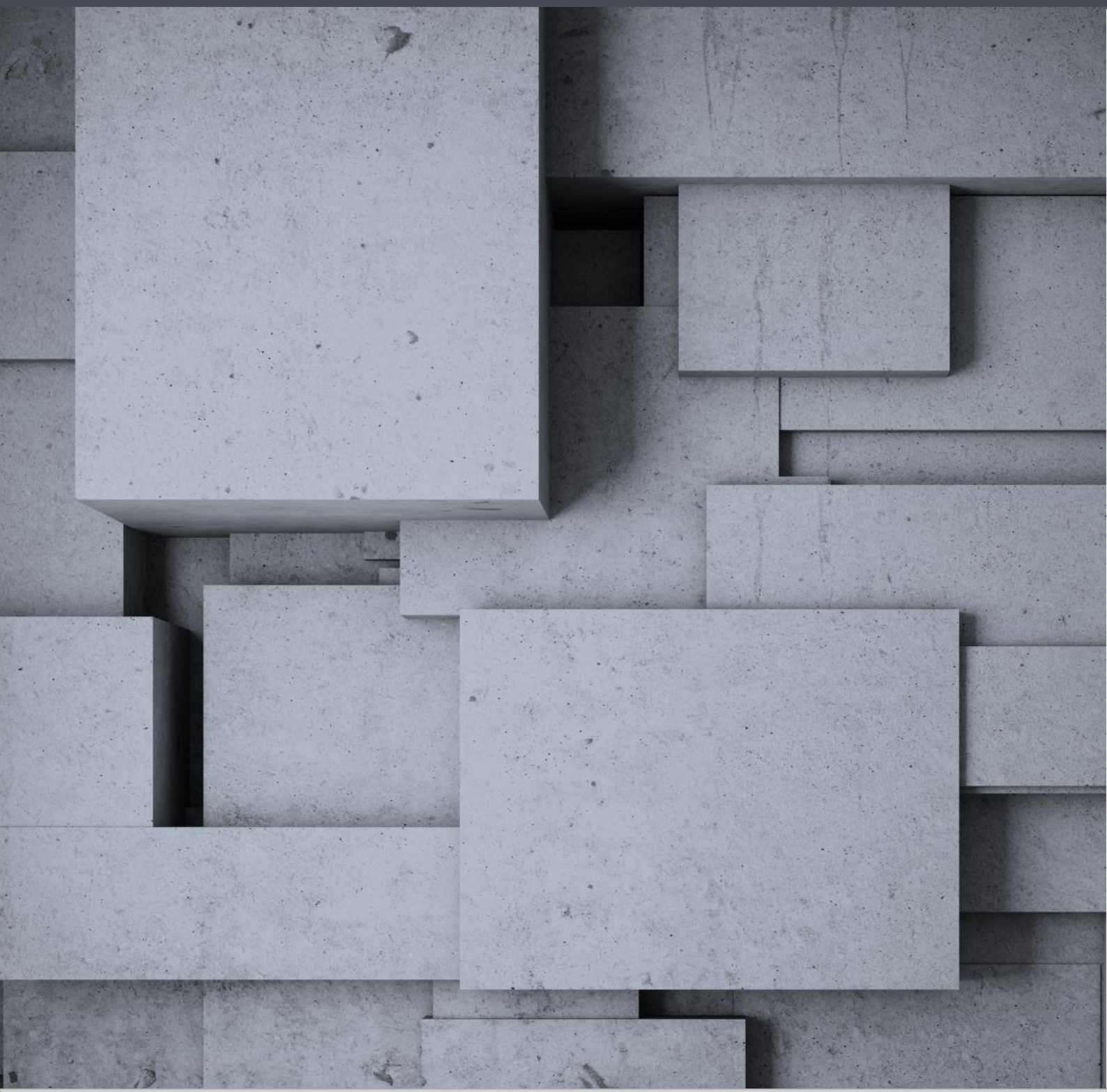

TRICENTIS
Platform Introduction Course
TRANSCRIPT

Contents



INTRODUCTION

INTRODUCTION

Introduction to Course

Hello, and thank you for joining this course. I want to take a few minutes of your time to explain why Tricentis has the most comprehensive continuous testing platform, and I want to help you to be able to use the tool to the best of your ability.

But before that, let's take a step back. Historically, in software development, testing was pretty much left until the end. With the dawn of DevOps, however, testing has become an integral part of the whole development and operational cycle, and Tricentis aims to make this a reality with our comprehensive platform by accelerating your release speed and by reducing the length of the test cycle. Not only that, but at the same time, we want to improve the quality of the software and business risk coverage. Sounds like a tall order, but let's look at how we can do that.

Tricentis offers enterprise automation from development right through to production.

Firstly, we look at analyzing and assessing risk to an application. This could be either through impact analysis of code changes in a release, or a risk assessment. We expose the risk and find the key aspects of your application to gain the highest risk coverage, with the fewest number of tests.

Then come the tests. We manage the tests that you will need to build in one central location -this includes manual tests, open source and model-based automated tasks. With this centralized test management, we also integrate into tools right across a CI pipeline.

Then once the testcases are built, we execute them at scale, end-to-end across over 150 technologies. But we don't stop at just test automation, we optimize by injecting test data, be it synthetic and masked data, and use Service virtualization to simulate key integrations.

And don't forget about streamlining operations. We can also use all of these automation components and reuse them in an RPA scenario. In this course we will focus on Continuous Testing. For more information on Tricentis RPA, check out our RPA pathway on our website.

All of this is supported by AI capabilities and powerful reporting to be able to pull valuable data from the whole process.

As mentioned before, we want you to get the most out of the platform, and for that, you need to know where to start and what parts of our platform are relevant for you and your role. We have developed some training paths to bring you up to speed on how to get the benefits out of our Tricentis products. Take a look at this next video to find out more.

Introduction to Roles

Welcome to our platform introduction course. I'd like to introduce you to the core Tricentis Testing Roles. These roles have been created to help you identify your place in the continuous testing journey. Each role will have a tailored learning approach to help you best utilize the Tricentis continuous testing platform. As you watch this video, try to see which role you identify with most closely, based on your day-to-day tasks and responsibilities.

The 6 core roles are:

- Test Analyst (also commonly referred to as a manual tester)
- Automation Specialist
- Extension Developer
- Performance Specialist
- Test Architect
- And Test Manager

Let's take a closer look at them.

The Test Analyst is responsible for initially identifying and subsequently defining the required tests, monitoring the test coverage and evaluating the overall quality experienced when testing the Target Test Items. This also involves specifying the required Test Data and evaluating the outcome of the testing conducted in each test cycle.

Test Analysts usually perform manual tests on applications, helping to detect errors earlier, and ensuring that the software in development satisfies the release requirements.

Complementary to the Test Analysts, Automation Specialists take care of the Automated part of the software Testing. Automation Specialists help to improve an organization's operating systems through the review, testing, repair, and maintenance of these systems.

They provide help with writing Requirements, provide input to the Test Plan, and have deep practical knowledge of the application being tested. Automation Specialists create and execute automated Test Cases according to Best Practices and often determine which data combinations are required for the different test scenarios. Automation Specialists create instances which reflect real business use cases.

They will also be responsible for maintaining the TestCases and making sure that they reflect the current status of the application. Automation Specialists then need to be able to analyze results to debug tests, identify defects in the application as well as generate the relevant reports. Overall, these specialists are often concerned with End to end testing of various UI and non-UI technologies.

However, standard out-of-the-box solutions are not always enough.

Extension Developers come into the picture when scripting, meaning customizations or extensions to the test objects or testing tools, is needed. They are responsible for the design, testing and maintenance of the software programs and their main focus is on developing efficient code. Thanks to their ability to deeply analyze the application,

Extension Developers help resolve any challenges of automating the software. They work closely with both Management and Stakeholders to overcome the issues with existing extensions after system upgrades and create any required ad-hoc customizations.

Additionally, Extension Developers integrate software components and build code to trigger the test execution via the Distributed Execution or CI pipeline. Overall, they are able to recognize whether the need for a new customization or extension is the best approach, or if an out-of-the-box solution would be possible.

While many other testers check an application's ability to perform the desired tasks, Performance Specialists test that the software will be able to satisfy requirements also when under enormous stress, or load. Performance Specialists develop and execute performance tests and identify risks to evaluate, measure, and improve application performance. They usually work together with the testing team to develop performance test plans and cases.

In a nutshell, Performance Specialists ensure system stability, reliability, capacity, scalability and work speed.

Additionally, they analyze the impact of performance interventions on the application's performance and provide reports to management. After the testing is done, performance specialists often analyze test results and coordinate with development teams for bug fixes. As well as coordination for bug fixes, they provide support in application design, development and deployment activities.

So far, we have seen how each Tester takes care of a part of the software testing project. But how do you coordinate different testers or testing groups?

Test Architects make certain that the approaches, tools and techniques used are built into a relevant methodology. They ensure the quality of the project, often steering it in the right direction thanks to their ability to generate and interpret risk assessments and prioritize accordingly.

Test Architects have overall responsibility for the test infrastructure, including environments and software. This is why it is crucial for them to have a full and deep understanding of the testing tool architecture to know how best to set up their environment.

Thanks to this knowledge, Test Architects can then:

- Identify what tools and technologies can be utilized.

-Align these technologies with the technologies already used across the broader development function, in-line with the skill set of the team.

- Make sure the right people work on the right test cases.

- Review Test Cases and Repository health.

- Design and develop the test framework and code libraries to enable the team to both use and enhance them across successive projects

- and Ensure best practices in testing projects.

As you may well know, the data you produce and the results your team achieves are only as good as the team's ability to showcase them.

This is usually the Test Manager's duty. The main responsibilities of the Test Manager are the setup of test methodology using appropriate testing tools, techniques, and agreed process standards as well as managing, coordinating, and reporting on test execution during the test period.

They need to keep track of the testing progress, bug status and the needs from the team. In addition to managing the test teams, the Test Managers will also need to create the test concepts, which include test strategy, test plan, test metrics, test levels, and test entry criteria. The test manager will define test abortion criteria, test finalization criteria, and test requirements. They will also analyze the test effort and automation potential at a high level.

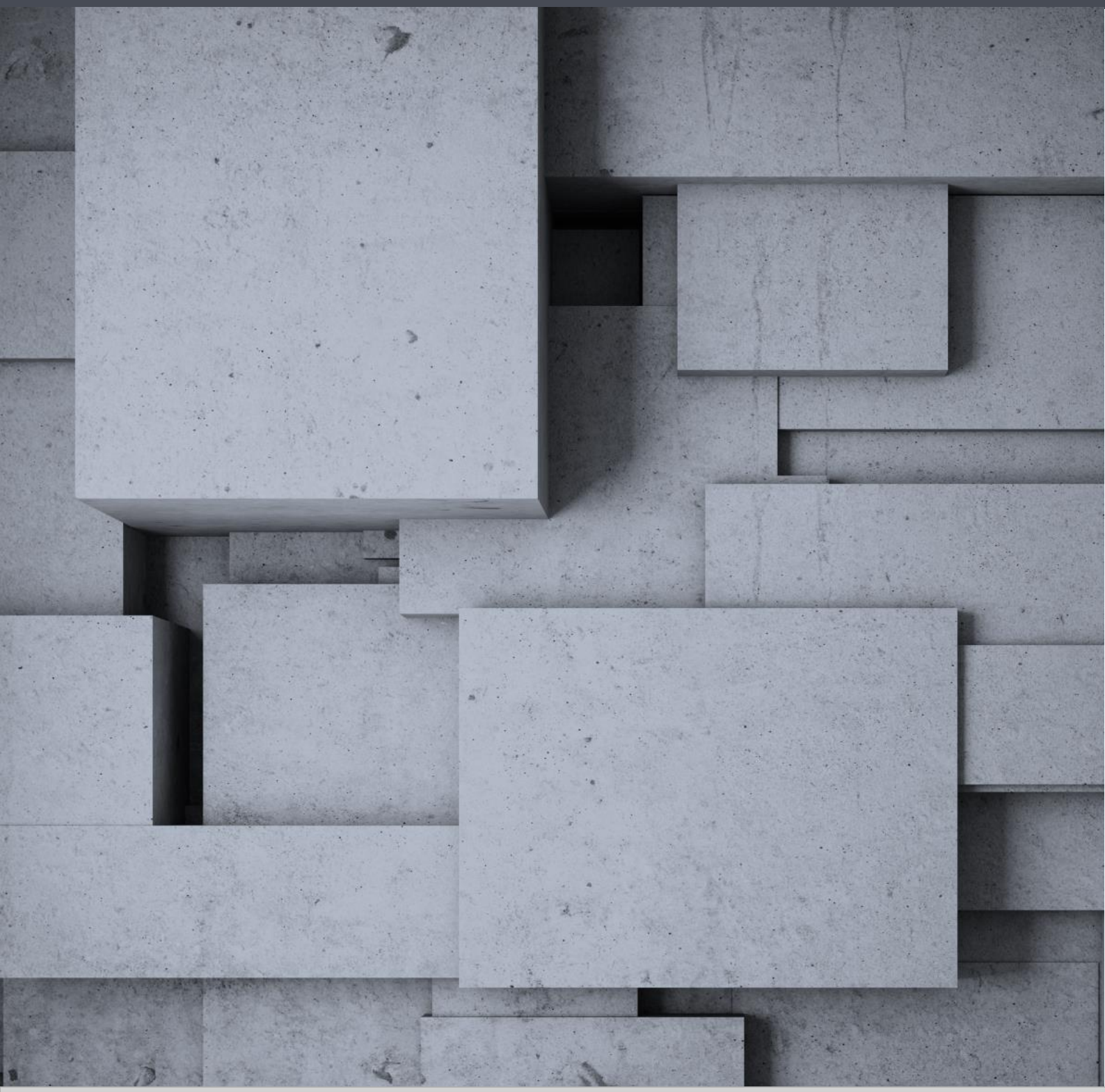
Test managers are responsible for defining guidelines for test case creation and ensuring the acceptance of the tests. Overall their goal is to improve the team efficiency and traceability. Test managers are often the key figures for future expansions of the existing engagements as they are the ones who will summarize the success or failure of the implementation of the testing tools and select the tools and techniques to use moving forward to increase efficiency.

Now that we have an overview of the different roles within a testing project, take a moment to find the roles which most closely correlate to your tasks and professional goals. But keep in mind that depending on your organizational or project structure, these roles may overlap or be divided even further.

The rest of this course is structured in such a way that key features and benefits of the Tricentis platform have been identified for each role. For this reason, have a look at some of the other sections as well to see if there are any added benefits or features which you may have missed that will help you in your business.

At the end of each section, we have compiled an overview of the available training resources to help you upskill and achieve the highest quality testing process possible. Good luck and we hope that our learning pathways help you achieve your career goals.

TRICENTIS



AUTOMATION SPECIALIST

AUTOMATION SPECIALIST

Create Reusable and Code-free Automated Tests

Welcome to the Automation Specialist Role Path video: Create reusable and code free automated tests.

As you can probably imagine, there is no time to waste when working with Agile and DevOps! Releases of new applications or versions of applications are now released in much faster cycles. Still, today most testers continue to wrestle with the high-maintenance, script-based testing approach designed decades ago.

This slows down testing and, as a result, releasing. For teams that follow this approach, test case writing can be an enormous task, hindered by the level of knowledge required to build even the simplest test cases. Even once the scripts are written, they are inherently fragile. Any minor change to the application can result in dozens, maybe even hundreds, of your Test Cases being out of commission. Additionally, scripted test cases are difficult to put into context. This may result in testers who don't have a deep and comprehensive understanding of all the outcomes expected from the application.

The Tricentis platform addresses these concerns by leveraging Model Based Test Automation to allow testers to create resilient business-readable test cases that can be easily maintained. Thanks to this approach testers can achieve test automation rates up to 90% from release to release. Let's have a closer look: Instead of programming a test automation framework, with Tricentis Tosca, you can rapidly scan the application's UI or APIs to create a business-readable automation model.

The Automation models, called Modules, are created by scanning the System under test, without scripting. The Tosca Modules will then provide Lego-like "building blocks" that can be combined and reused to create your tests. If your application changes, e.g., a field is added or removed, you can simply update the Module.

The change will then be automatically propagated from this centralized location to all impacted tests. What if you are dealing with applications with dynamic object identifiers, that can affect the reliability of test automation? With Tosca, you can assign and modify the identifiers being used so that each object can be consistently identified during test execution.

On top of this, Tricentis also provides you with a library of ready-to-use Standard Modules, which are models to help you quickly steer the most commonly used applications, and Special Execution Tasks which allow you to perform tasks that would otherwise be difficult or time-consuming to automate – such as opening and closing programs, taking screenshots or performing clipboard operations.

Tricentis Tosca's strength does not end after Modules have been created. Projects artifacts can be linked across sections by simply dragging and dropping them, all for a quick and seamless work experience.

This is how simple it is, with Tosca, to test an application: We can then build our TestCases by dragging and dropping the modules into Tosca's Blue Section and filling in the relevant business information. To execute our Test Cases, we then simply drag and drop them into

our execution lists within the green Execution Section. In the Execution Section all your test results and log info are also saved and versioned.

This means that they can be accessed at any time. All these enhancements allow QAs to fully concentrate on the actual tests and the application's behaviors. With Tricentis testers can focus on meeting the functional and business requirements of the software. You can find out more about how Tricentis products help you in your role, by viewing another video or take a look at how the Tricentis Academy enables you in our where to go next video.

Centralized Test Scenario Creation and Maintenance

Welcome to the Automation Specialist Role Path video: Centralize test scenario creation and maintenance.

One of the most daunting testing challenges is the ability to test a required function extensively. You don't only test a function with one set of data but with multiple sets of data, covering various scenarios.

However, it can be challenging to verify if a function, or requirement, is sufficiently covered, using the right amount of test cases, containing the correct data combinations. Many teams use excel to store and maintain test data. However, if you use software like excel to store your test data, there won't be a simple way to design and control how test cases and scenarios would unfold.

This means that you might miss out some scenarios and, as a result, lack coverage on some of the application's requirements. Additionally, Excel does not perform well when it comes to maintaining a changing database or handle high volumes of data. It is also difficult to have a clear and full overview of what has already been stored. This obviously results in potential duplicates and redundant data.

All of these issues can lead to duplicity of Test cases and Irrelevant Test scenarios. The bottom line is that you will probably end up with many testing scenarios being partially or even totally irrelevant while other important scenarios are missed entirely, because you did not even realize that they had not been covered already. An additional problem is, of course, creating the test cases themselves. Usually, they will be very similar, with only certain aspects being different.

It is a lot of work to create and adapt the test cases accordingly to the data combinations. It is even more work to maintain those test cases. Every time something in the system changes, you will need to change tens or even hundreds of test cases, manually, one by one.

This mix of reasons is why most of the Test portfolios covers only 40% of the business risks with 67% of redundancies in the regression test portfolios. To summarize, we are wasting lot of time in testing the wrong things, duplicating efforts or creating test cases. To answer these needs, Tricentis embraces the "shift left" concept (heartbeat concept). We use risk prioritization to cover 40% of the risk just in the beginning phase of the implementation with only around 10% of the test cases created.

But this is just the initial phase of the implementation. Following this, and additional 50% of TestCases is used to optimize testing and arrive at 90% risk coverage. This obviously means a sharp increase in business risk coverage, combined with a reduction in the number of test cases. Tosca users can utilize the Test Case Design section to push business risk coverage by utilizing the template concept.

This means covering over 90% of business risk with only 60% of the test cases. In fewer words, Tosca's Test Case Design achieves 90% of the maintainable automation. But how do we do that? Well, thanks to Tosca, and more specifically thanks to the Test Case Design section of Tosca. Test case design, or TCD, is Tricentis Tosca most resilient Solution to centralized test scenario creation and its maintenance. Tosca's TestCaseDesign section is a built-in functionality that works out of the box.

TCD enables you to put your scenarios into a logical structure and provides an overview of which test cases are needed, to cover the Requirements you have defined. Tosca Test Case Design helps you create your data-driven test scenarios.

Our approach assists you by provisioning and managing test data, while locating and alerting you to redundancies in your test case portfolio. You can also specify which data is used to verify results. After this, by linking to TestCase templates, you can quickly achieve full coverage in automation with just a few drag & drops and clicks.

Then, thanks to Instantiation, you can generate all of the required Test case scenarios with a click, because the templates are linked directly to your data in the TestSheet. This link allows you to plan and specify your test cases wisely in order to maximize test coverage and reduce effort and redundancy later. This is because all data combinations are expressed already in the testsheet while the Template provides the structure and conditions to apply those combinations.

Tosca TestCase Design, not only helps you create meaningful TestCases quickly, but also helps Execute the right ones first. Additionally, the link between TestSheets and Test cases in Tricentis Tosca, results in each TestCase being assigned a unique and well-defined purpose. This happens via single easily identifiable verification, that allows a quick and easy error identification.

All this boosts and simplifies maintenance, transparency, and traceability. Also, thanks to the link between your TestSheet, your execution and your requirements sections within Tosca, it becomes easy to spot which requirement has an error within it, following an error during testcase execution.

This means, that due to risk weighting you can see the overall effect that this error has on you overall risk coverage for the application. You can find out more about how Tricentis products help you in your role, by watching another video or take a look at how the Tricentis Academy enables you in our where to go next video.

Reduce Time Spent in Test Execution

Welcome to the Automation Specialist Role Path video: Reduce time spent in test execution.

In an increasingly agile world, the demand for speed in testing becomes critical. Nobody wants to be forced to delay a release just because of the amount of time spent executing the required tests. The question becomes how can we reduce the time spent in execution and how quickly can we view, analyze and react to results?

To tackle these issues, Automation Specialists will be looking for: the ability to quickly configure and run test cases simultaneously on multiple machines with little to no human input to speed up the total testing time as well as a clear overview of the Test Log and Execution Results. With Tricentis Tosca Distributed Execution, or DEX, you can reduce your execution time by distributing tests across all available computing resources.

These include computers in your network, virtual machines or the cloud. These resources are called agents and can be configured to ensure that the right agent is selected depending on the technical requirements of the batch of tests to be run. These batches or groups of tests are organized into Test Events, which are checked into the common repository in a multiuser environment and distributed to the appropriate agents to be run simultaneously.

Using remote desktop connection or RDP, it is even possible to run UI tests on machines with locked screens via unattended execution. The status of the distributed test events is easy to monitor in the Distributed Execution monitor – you can see an overview of which tests are being run on which agents, and the results of those test runs. A detailed breakdown of the test log info and step by step results are stored in the dedicated Execution section in Tricentis Tosca.

These are linked directly back to the project requirements. In this way, you have a clear overview of the status of the release as well as the ability to identify any potential issues before release. You can find out more about how Tricentis products help you in your role, by watching another video or take a look at how the Tricentis Academy enables you in our where to go next video.

Manage Your Test Data in Distributed Environments

Welcome to the Automation Specialist Role Path video: Manage your test data in distributed environments.

As we all know, any test will need test data for it to properly function. Having the right data is critical to achieving comprehensive and successful testing results. The more representative the data, the more thorough your tests will be. To build applications quickly, you need to procure test data quickly. Stateful test data not only produces the specific application conditions that are required to set up a realistic test; it also enables you to drive the test through a complex series of steps.

It's essential for executing end-to-end regression tests. An example for Stateful Test Data is Data that changes during a test case run, such as the registration of an account that goes from new, to active, to blocked. Test data management is about identifying and implementing the right test data for the execution of your tests. One of the challenges associated with test data management is the process of setting up and manipulating stateful data.

The identification of the appropriate test data is the most time-consuming and resource-intensive challenges faced by testers today. Test automation and continuous testing, required to meet the speed promise of Agile and DevOps, put even more pressure on this process. In larger enterprises, test data is managed by either some External Test Management Tools or in Excel which takes a lot of manual effort to manage and maintain. The process to link test data to tests might also take a lot of time.

Another problem that companies and organizations must face is compliance. For years, many companies have extracted data from production environments to use as test data. However, regulations such as the GDPR regulation that came into effect in Europe in 2018 require that any test data extracted from production systems be masked irreversibly and deterministically.

With Tricentis Test Data Services and TDM Studio, Tricentis offers a central test data hub with a structured approach to creating and maintaining structured, stateful test data that will be 100% compliant with data privacy regulations. Test Data Services is a web-based application that enables you to manage your stateful test data in distributed environments.

With Tricentis Test data services, any tester can rapidly access the exact data needed on demand for a given scenario and evolve it throughout the course of the test. With a comprehensive repository of stateful test data objects, you simply select the data set that corresponds to your test and indicate how it should change at each step. This enables continuous testing of scenarios that might otherwise be impractical to automate.

Synthetically generated data enables broader coverage and negative testing. It lets you simulate data types and ranges that might be difficult to find in production data. Using advanced queries, written in plain English, you can create, retrieve, modify, and delete data, both through the web interface as well as from Tosca. You can also track the status of a data record. Most importantly, test data records are locked during execution and unlocked to be used in other test cases when execution is done. This is how TDS can prevent errors and ensures that distributed environments have access to data simultaneously, without issues such as values being overwritten. We also need to address

the issue of compliance - TDM Studio wizards offer a unique functionality for generating test data simply, safely, and quickly.

TDM Studio's intuitive extraction wizards work hand-in-hand with advanced anonymization techniques. They extract data from production and then mask it to meet GDPR privacy requirements. Now, your test data is 100% compliant. Not only that, but you also don't need any expert knowledge to create masked subsets of consistent data across all systems.

Furthermore, no data breach is possible since no production data is ever touched. That means, no risk that data masking and sub setting was done incorrectly. You can find out more about how Tricentis products help you in your role, by watching another video or take a look at how the Tricentis Academy enables you in our where to go next video.

Shift Left With API Testing

Welcome to the Automation Specialist Role Path video: Shift left with API Testing.

As more and more of the world we live in relies on software applications, the cost to companies and users of defects in these systems is increasing exponentially. To avoid these costs, QA teams need to test earlier in the development process and test faster to keep up with agile release cycles.

API automation is crucial for this shift-left. Unfortunately, as the need for API testing has increased, so has the awareness of problems that API testers face daily. But what are the issues with API testing? First of all, API testing requires a higher level of knowledge than most other testing. This is because, with other testing tools, testers need to know how to test using JSON or other markup languages. That means that many companies will have to set up specialized testing teams to test their APIs. Secondly, API tests are particularly hard to automate.

The scripts that the API tests are built upon, rely on a request or response type of interaction with the system backend. This interaction is quite difficult to automate properly using the tools that are out there at our disposal. And this is where Tricentis' tools come into play. Tricentis API Scan and API testing engine simplify the transition from UI testing to API testing. This allows any tester to rapidly create API tests from an intuitive business-readable interface.

These request/response pairs created by developers' can automatically be converted into business readable Tricentis Tosca TestCases. These test cases have the same look and feel as Tosca UI test cases. This helps jump-start the test creation, reduces the learning curve and time required for QA. The intuitive interface allows non-technical testers and business analysts to easily review, modify and extend the test cases without worrying about JSON, XML, or XPaths. The API steps can also be combined with any GUI steps to create fully automated end-to-end test cases which can jump in and out of the UI and API of an application.

The API Scan standalone application can be accessed, for free, by Developers. As Developers perform quick sanity tests on the APIs they're developing, key request/response pairs can be captured through our light-weight API Scan tool. The results can be used by QAs to build the tests. Thanks to this feature, the Tricentis API Testing tool will bring the operations of your QAs and Development team closer together. As the API Scan is Part of Tosca, the resulting test cases can be totally automated. Test Executions can be scheduled, and fully automated tests can be run, even on a daily basis to ensure a quick, solid regression tests bench. Additionally, most API testing tools only support REST and SOAP protocols. This means that these tools, while still being helpful to an extent, hardly can supply the needed testing infrastructure to take care of all your API tests.

On the other hand, Tricentis supports your entire API journey. From web APIs to legacy backed protocols Tricentis is with you to enable true end-to-end testing. You can find out more about how Tricentis products help you in your role, by watching another video or take a look at how the Tricentis Academy enables you in our where to go next video.

Avoid Dependencies and Test Earlier with Service Virtualization

Welcome to the Automation Specialist Role Path video: Avoid dependencies and test earlier with Service Virtualization.

Chances are that the application you or your company is developing will be working in conjunction with other applications, services or systems. This means that during the testing process it won't be enough to simply ensure that the software under test itself is ready for release. You will also need to test that all the connections with the other systems are up to scratch.

After all, the best car engine in the world will not be able to perform if it is not connected properly to the fuel tank and wheels. Many times, the need for other systems, sometimes dozens of them, to be available for testing at once becomes a fundamental issue.

And that can result in a poorly performing application at release. Many services that work around the application might be expensive to run just for testing. In other cases, the applications your software must interact with might be under development, offline, have bugs or be right out broken.

For all these reasons Service Virtualization is crucial in an application testing lifecycle. However, many times the Service Virtualization infrastructure that many testing platforms can build is very limited in size and scope. And then they are only able to reproduce a very limited number of services at the time. Additionally, Service Virtualization has always been a very demanding topic.

Requiring elevated skills and time to properly setup. With Tricentis Tosca Orchestrated Service Virtualization or OSV, Tricentis brings its trademark speed and ease of use to service virtualization.

Tricentis Test-Driven Service Virtualization simulates the behavior of dependent systems that are difficult to access or configure so you can continuously test without delays. Thanks to OSV intercept, Tosca can quickly record an existing service and its messages, then use the recording to simulate the service.

Testers can leverage the same business view developed during API testing to create and manage service virtualization scenarios. Tosca OSV can also mimic data consumption by various application components. This, together with the effortless Message Verification and Analysis, makes it possible to check thousands, if not millions of messages. Tosca will automatically flag messages that are improperly formed or sent in the incorrect order.

And that means one thing: No more manual validation. Once OSV is setup, it is also easy to maintain. That is thanks to the tight integration between OSV and Tricentis Tosca, that connect front-end and back-end. Teams can create instances with Tosca's test case design and template automation functionality. This, in conjunction with stateful data in Tricentis' test data service, is used to create end-to-end tests with re-usable virtual

services. This enhances the risk coverage and shifts the testing left, allowing enterprises to expose software defects in earlier development lifecycles. And that's where they are faster, easier, and cheaper to fix.

OSV Test Cases can also be used for laying the groundwork for future regression tests and UI tests once the application has gone live. In the same way you can use your regression test repository to create a service virtualization of an already tested service.

Now you have seen how Tricentis Test-Driven Service Virtualization removes the single biggest barrier to achieving continuous testing: By accessing to a complete test environment with all dependent systems with the appropriate configuration, functionality and test data.

You can find out more about how Tricentis products help you in your role, by watching another video or take a look at how the Tricentis Academy enables you in our where to go next video.

Accelerate SAP and Packaged Application Testing

Welcome to the Automation Specialist Role Path video: Accelerate SAP and packaged application testing.

The popularity of packaged applications like ServiceNow, Salesforce or SAP is booming! We see an increase in prevalence in organizations as well as year on year growth in budget allocation to packaged applications. This growth and popularity come in large part thanks to the speed with which they are delivered.

Of course, this means that there is a constant and increasing demand for continuous testing of these applications. Today's enterprise applications are highly distributed and interconnected. To truly protect the user experience as all the systems, evolve in parallel, it's critical to test complete end-to-end business processes.

However, these may span multiple application types and components. With Tricentis, this is simple, as we support multiple technologies in various industries. From a single intuitive interface, you can rapidly define end-to-end tests across mobile UIs, web services, ESBs, APIs, web UIs, mainframes, and enterprise applications.

Tricentis Tosca helps enterprises plan, construct, stabilize, and execute resilient automated regression tests for packaged applications. This helps simplify testing and ensures that functionalities work flawlessly in the context of complete, end-to-end, business transactions upon release. How does this all work? You can utilize all of the standard Tosca benefits, as well as additional packaged app content and extensions provided by Tricentis and our partners.

Tosca has a business focused interface meant to increase the ease of use when creating test cases. Testers can also leverage Tosca's Model-Based Test Automation for rapid test creation, management and execution. This also means a substantial reduction of redundancies in the object repository and optimizes effort required to adjust to fast release cycles.

Finally, consolidated analytics and reporting help the testing team keep track of the status of their overall testing progress. Packaged application testing is even easier and faster in Tosca than with other technologies! We have found that over 80% of the business process requirements and account collateral used in packaged applications testing are generic across the industry.

This results in 60% of the test case instances or test scenarios being generic as well. With Tosca, you get out-of-the-box scan functionality for PDFs and Salesforce. Standard Modules for SAP and PDF come with every installation of Tosca free-of-charge. If more help is needed, Tricentis offers Accelerator Packages, which contain generalized end to end scenarios to help you speed up testing, maintenance and execution of your test portfolios.

These packages include the pre-created models that you can then reuse an infinite number of times to build up your test portfolio. This means that whatever packaged app you're working with, whether it be SAP, Oracle, Finacle, Pega, or more, you can have a jump start in Automation with end-2-end tests ready to go. Tricentis also teams up with our partners to deliver Tosca extensions for more tailored model creation and technical steering. Let's look at SAP as an example of our comprehensive support: Tosca allows you

to automate on any SAP stack, any version. We provide support for all SAP UIs like SAP WinGUI, SAP Fiori, SAP Hana & many more: Tricentis Tosca was the first in the industry to feature SAP-certified integration with SAP Solution Manager, providing customers with risk-based and change-based testing for SAP software. We also provide assistance with SAP migration and update analysis, thanks to Tricentis LiveCompare, to ensure that you have a 100% defect free release.

With LiveCompare, Automation specialists don't need to worry about which tests need to be updated when SAP has a new release, because LiveCompare will take care of that for you. All you will need to do is worry about the Gap – updated features which do not have created test cases. Live compare will tell you exactly where this gap is.

You can find out more about how Tricentis products help you in your role, by watching another video or take a look at how the Tricentis Academy enables you in our where to go next video.

Ensure Data Integrity with BI/DWH Testing

Welcome to the Automation Specialist Role Path video: Ensure data integrity with BI DataWareHouse Testing.

In many successful businesses, leadership takes decisions based on Data out of Business Intelligence reports. Risk Management requires a proper baseline related to the quality of data. Quality data can build the trust needed to invest in further innovation as well as a good risk management strategy.

However, one of the biggest challenges with BI and Data Warehouse projects is guaranteeing the integrity of the data and ensuring that any errors are detected as early as possible. This is a challenge as both the size and complexity of databases increase every day. Additionally, most enterprises are using agile and DevOps practices in their projects that mandate frequent changes in data.

Obviously, there is a need for testing all these changes, but existing procedures struggle to keep up. Complex queries, limited number of verifications and manual testing for BI reports are not able to keep up the pace of the changes. If this wasn't difficult enough, high skills are required for this type of testing. Test developers need ad-hoc training to be able to perform.

For all these reasons BI and data testing can be very cost intensive. To face these issues, Tricentis Tosca BI is designed to limit SQL scripting and focus on Business Analysis and Data stores. This is achieved with our wizard-based approach to create tests. Our solution is built on Model based Test Automation which is readable in plain English. Moreover, by using test design and risk assessment we are shifting the attention from technical data testing to a business context.

Our solution provides tangible risk coverage for data quality and business logic at each stage of Data Warehouse and BI testing. Now you can save time and money by avoiding the manual creation of TestCases, being able to easily maintain TestCases and obtain sustainable regression testing. The great thing is that this works end-to-end in your data warehouse.

Rather than sitting in your data warehouse it sits on top of your data warehouse. Basically, we don't impact your process, we write on top of it with our priority in memory caching technology. This results in a fast and secure testing process. Tricentis Tosca BI is the only tool that tests data quality of your big data landscape from beginning to end. Another advantage is that with tosca BI we can connect to any type of file, Database, API or Report wherever they are hosted and ensure the data is checked efficiently.

How does this work? Pre-Screening file Tests are basic initial tests that are performed on files before going for intensive tests. Then Tosca BI Vital Checks enables critical errors in the DWH acquisition process to be detected earlier in the testing cycle. This way a fast decision of "go" or "no go" for more exhaustive tests can be made. Tosca BI Reconciliation compares two data sets to each other.

Here all rows and columns from both the source and target data sources are compared. Report testing allows you to test content of the BI report against your data in the database. BI reports can involve HTML, PDF file, Excel etc. and all these reports can be automated using Tosca Commander. In this way you can monitor your data by ensuring data quality

over time and regular execution. To summarize, you have seen how thanks to Tosca BI you will be able to Detect data integrity issues earlier, avoiding manual testing. At the same time, you can ensure that regulatory requirements are fulfilled and improve your confidence in the quality of your data. This is all necessary for the ability to make the right business decisions.

You can find out more about how Tricentis products help you in your role, by watching another video or take a look at how the Tricentis Academy enables you in our where to go next video.

Automation Specialist - Where to Go Next

Welcome to the Automation Specialist Role Path video: Where to go next.

In the previous videos we have seen how Tricentis Tosca can help you store and maintain high volumes of data. As well as, how you can use the Test Case Design section of Tosca, in conjunction with the template function to create and easily maintain large numbers of TestCases.

We have also seen how you can cut down execution times by using Tosca Distributed Execution functionality. Additionally, we have covered how Tosca's TDS can help view and manage your stateful testing data. This while TDM Studio will help you meet the data privacy requirements. And when it comes to packaged applications, we have seen how thanks to the integration between Tosca and Live Compare, you can test updates of several packaged applications.

We have also covered how thanks to Tosca BI you can ensure the integrity of data in your data warehouse. Finally, we have seen how thanks to Tosca API scan, you can test earlier, on all of the most widely used technologies. Also, thanks to Tosca OSV you can simulate services that are offline or unavailable, so that you can test integrated applications in the early stages.

The Tricentis Academy offers you a wide variety of courses to learn how to master all of these Tricentis offerings. Thanks to Automation Specialist Level 1 and 2 you will gain the ability to automate your TestCases, link to Requirements, manage your Execution lists and create several automated test cases at once thanks to the Template function. These courses are available as both Instructor-led and Self-paced learning and the duration is approximately 5 days combined.

Test Design Specialist Level 1 will provide you a deep dive into test planning in Tricentis Tosca. You will learn how to work within the Requirements and TestCase-Design sections of Tosca. You will learn how to implement risk-based testing in Tosca, how to weight your requirements, how to create TestSheet structures and how to combine your values both manually and automatically. This course is available as both Instructor led and Self-paced learning.

The duration of this course is approximately 3 days. Thanks to the Test Design Specialist Level 2 course you will be equipped with the necessary theoretical and practical knowledge to work with test data which is stored in the TDS database. This includes: setting up, installing and configuring Tosca Server and TDS as well as creating, retrieving, searching for and updating or moving test data for efficient stateful data management.

This course is available as both Instructor led and Self-paced learning. The duration of this course is approximately 1 day. Automation Specialist for SAP will teach you how to create SAP modules in Tosca and how to use the SAP standard modules to achieve full SAP test case automation.

This course is available as both Instructor led and Self-paced learning. The duration of this course is approximately 1 day.

Additionally, thanks to the extensive library of videos and tutorials found on both our website and YouTube channel you will be able to master the use of Live Compare together with Tosca. Thanks to the Automation Specialist for API course you will learn how to scan your application's API and create Tosca Modules and TestCases from it.

Additionally, you will learn how to work and automate API TestCases. This course is best taken after Automation Specialist Level 2 as the potential of API test cases are best understood when put in conjunction with Tosca's template capabilities. Automation Specialist for API is available as both Instructor led and Self-paced learning. The duration of this course is approximately a half day.

By completing the BI Specialist course, you will learn about core BI functionalities such as Database queries, File tests and using the Vital Checks Wizard, as well as the basics of test automation within Tricentis Tosca, allowing you to create basic automated TestCases as well as expand your knowledge to fully test the BI process. This course is only available Instructor led. The duration of this course is approximately 2 and half days.

We also offer the Automation Specialist Practitioner course. This course is a hands-on instructor led coaching session, that will take the core skills developed in the Tricentis online trainings and hone them with a particular focus on your own SUT and tailored to your project's unique requirements.

In addition to all our courses you will find a large variety of videos covering in-depth details surrounding automation with Tricentis Tosca on both our website and YouTube channel. Remember to subscribe to our YouTube Channel to be always up to date with our weekly video releases.

Contact us, if you have any suggestions or requests for new videos!! We also recommend that you attend Tricentis events such as Accelerate, Users Conferences and others. These events serve as opportunity for you to meet the people behind the products and trainings and learn more about all the current, as well as future offerings.

We hope you have enjoyed this short overview and are now ready to dive into the Academy content! Good luck and happy testing!

TRICENTIS

TEST ANALYST

TEST ANALYST

Adapt According to Your Development Strategy and Improve Traceability

Welcome to the Test Analyst role path video: Adapt according to your development strategy and improve traceability.

No machine can adapt to a changing environment like humans can. Therefore, manual testing is and will remain a crucial part of any software lifecycle and release. Even though automated testing is becoming more prominent, manual testing will still play a key role -- For example with procedures such as smoke or boundary testing.

For validating new applications and feature releases manual testing is still important. When you have both manual and automated testing in a project, Test Analysts and Test Managers often suffer from not having a proper view on the Requirements traceability. There needs to be a way for test managers to consolidate data from different automation testing tools alongside manual tests.

This makes it easier to know what has been automated and to see the test requirements and trace defects. Test Analysts or Test Managers also need help when it comes to triggering and scheduling execution; solving the problem that Automation Engineers are often bottlenecks. The main challenges that Test Analysts face, are the following: -Keeping track of the requirements, that their tests need to fulfill. -Understanding how these requirements relate to the Test Cases, that provide the guideline for the actions the tester will perform on the application. - And how to link the defects that were found to the test cases themselves?

In the end effective testing is like a fine-tuned orchestra. The result will be disappointing if not all parts are directed properly. A lot of the Test Analyst teams rely on “simpler” tracking technologies. For example, Excel can be used to store your Test Cases. However, it is not easy to create bulk updates. Additionally, It is almost impossible to keep track of hundreds of requirements and maybe even thousands of Test Cases. Versioning test cases is another need.

A Test Analyst might want to modify the tests as the application-under-test changes. That way, they can execute multiple versions of the test case when they test different versions of the application. Also, a Test Analyst would want to reuse test cases. This is so they can build a library of test cases that can be re-executed across different releases, sprints, or environments. All of this is difficult in excel.

To use an ALM to keep track of your testing elements is another option. And yes, these tools are usually more efficient than simple spreadsheet programs. However, the biggest problem with using an ALM for tracking tests is that ALMs are NOT testing tools. They lack full-featured comprehensive test management functionalities such as Versioning test cases and reusing test cases.

Overall, testers complain about the lack of one all-inclusive tool. Something that can assist Test Analysts from Test Design through to Test Execution and result handling. Tricentis qTest Manager is our response to all of these needs. qTest Manager is a comprehensive Test Management suite that incorporates Test and Requirement Planning capabilities with

active tracking and reporting of test results for manual, exploratory and automated testing. It includes a wide range of features to track, merge and organize all major types of testing. Using the Test Explorer engine, your exploratory sessions can be recorded, categorized and translated into manual or even automated Test Cases, ready for future use.

Recorded sessions can include screens and steps this is beneficial for User Acceptance Testing, where the testers are business users and not necessarily QA members. Recorded sessions are also good for executing traditional scripted tests by using the „Run with Explorer“ button in Test Execution, and provide additional testing evidence. The recorded sessions can also be used to generate new test cases, whether manual or automated, so they don’t need to be created from scratch.

Finally, the recorded sessions can also be used as evidence to create defects or as documentation for training material or user guides. QTest dashboards are comprehensive and flexible, for example when organizing your projects according to different development methodologies.

This combined with JIRA integration of all manual qTest objects and Tosca automation objects will help Test Analysts achieve the highest level of traceability for their efforts. After all, our motto is: Test Smarter, not Harder.

You can find out more about how Tricentis products help you in your role, by viewing another video or take a look at how the Tricentis Academy enables you in our where to go next video.

Seamlessly Integrate Your ALM Tools

Welcome to the Test Analyst role path video: Seamlessly Integrate Your ALM Tools.

Working as a Test Analyst, you are probably familiar with the following Tool challenge: the tools you use to project manage the testing artifacts as well as track progress and results, are not the same as the tools used to schedule executions and track defects.

Although some ALMs offer more options such as advanced tracking which try to fit more stages of the testing lifecycle, the current reality is that none can cover all requirements fully. This is the reason why testing tools are trying to integrate with ALMs more and more. Usually the integration works in a one-way method only. The data is moved from the ALM into the tool.

This means the data is taken and migrated into the testing tool, but it is not able to report back changes into the ALM. This then compromises the report generation and Management capabilities of the ALM.

However, Tricentis offers a solution for a deep and varied integration with the most used ALMs. Our tool provides integration with ALMs like Jira, Version one and Rally.

The bi-directional integration offered, makes the Tricentis approach to the ALM integration problem revolutionary. Data is not only pushed from the ALM to our testing tools, but also from the testing tool back to the ALM. So, if different departments of your company use an ALM to coordinate efforts, you will be able to share test results without using additional software.

But where Tricentis really excels is in its integration with Jira. Tricentis qTest is the No. 1 test management solution for Jira Software, as it integrates in real-time with Jira releases, issues, and defects; enabling you to plan, track and test smarter and faster than ever before.

Many of the vanilla Jira functions and apps are aimed at small teams. Thanks to its targeted integrations, Tricentis qTest delivers comprehensive test management, built for agile, with the enterprise features that other platforms simply lack.

Features of the Tricentis integration with Jira that stand out include: The ability to integrate any type of Jira deployment, such as, Cloud, Server and Data Center. The capability to easily map test projects to multiple Jira projects and instances, that makes the integration scalable to enterprise level.

The possibility to streamline test scenario creation directly within Jira. And our first-class enterprise reporting engine, that allows you to know when your issues are ready for production. On top of the main Jira integration, Tricentis is proud to field deep integrations with the Jira apps like: our renowned Real-time integration lets you Quickly access and analyze your latest Cucumber test scenario results.

Additionally, all projects will be easier to track because you will be able to consolidate all Cucumber automation into one location.

So that was a short overview of how Tricentis solves your ALM problems, watch more videos on the Test Analyst role or watch the where to go next video to see our training offering.

Generate Custom Reports for Key Metrics

Welcome to the Test Analyst role path video: Generate custom reports for key metrics.

To provide quick, and intelligible reporting is one of the major difficulties all testers run into. And this, of course, includes Test Analysts as well. If your team is using Excel this can be an especially challenging task. Creating properly generated Excel reports requires a certain level of application knowledge. And can be easily broken by accidently hitting one wrong key.

Additionally, Excel is difficult to scale. And keeping track of hundreds or even thousands of evolving elements is nearly impossible. Some of these issues might be easier to cope with, if your team is using an ALM to track the testing project. However, ALMs don't act as a centralized repository for all elements of testing. Like requirements, Test Cases and Defects.

Therefore, it is still going to be difficult to generate comprehensive reports. Additionally, using ALMs may be difficult because the ALM tool is managed by a different group, meaning that there are organizational barriers to customizing fields or building reports that fully satisfy your needs. The Tricentis platform offers two reporting tools, depending on your needs: Tricentis Analytics or qTest Insights.

Tricentis Analytics is our most comprehensive reporting tool. Whether you are using just one Tricentis products, or multiple different products, you will be able to centralize your data gathering in Tricentis Analytics. Tricentis Analytics gives you an overview of all test results from Tosca, Flood, and qTest.

You will have an overview of the status and progress of all projects, including automated functional and API tests, manual tests, and performance testing. Tricentis Analytics provides portfolio level visibility through a BI reporting engine to consolidate, manage, and analyze activities across the Continuous Testing Platform for your enterprise agile and DevOps initiatives.

Thanks to the custom reporting potential of Tricentis Analytics you will be able to perform Risk reductions, time to resolution, speed to market and testing progress analysis in real time. Additionally, you will be able to build traceability and coverage reports from your integrated DevOps tools with the intuitive drag and drop interface to estimate your release readiness.

If your team is only using qTest, you also have the option to utilize the qTest Insights, a powerful built-in reporting engine for the qTest Suite. It gives the testing team a self-service business intelligence tool to consolidate, manage and analyze all the testing data along with the integrated ALM issues. Custom Dashboards, smart filtering, interactive charts and visual heat maps - all qTest features- will enable you to create a comprehensive picture of your testing journey. We hope this video has given you a good idea of how you can best utilize the Tricentis reporting tools in your projects.

Please continue to watch the rest of the videos to learn how the Tricentis platform can assist you in your Test Analyst responsibilities or jump into the where to go next video for a guide on how the Tricentis Academy can help enable you to use these features.

Test Analyst - Where to Go Next

Welcome to the Test Analyst role path video: Where to Go Next.

In the previous videos we have seen how, thanks to qTest, you can keep track of your testing artifacts and how it can seamlessly adapt to different development methodologies. We have also seen how Tricentis qTest flawlessly integrates with several ALMs in real time.

During the qTest Specialist Level 1 course you will learn how to set up your own test project from start to finish. In this 1-day online course you will learn how to efficiently track all your Test Cases, Requirements, Releases and Test Runs in a fast and organized way. We will explore how to create Releases and link Requirements, how to quickly update any modified Test Cases and test objectives, then, how to create Test Runs as well as manage results.

qTest Specialist Level 1 will also touch on how to work in environments where an ALM has been integrated with qTest as well as how to manage and perform Exploratory Testing using qTest Explorer. To get enrolled for the qTest online course visit the Academy website here:

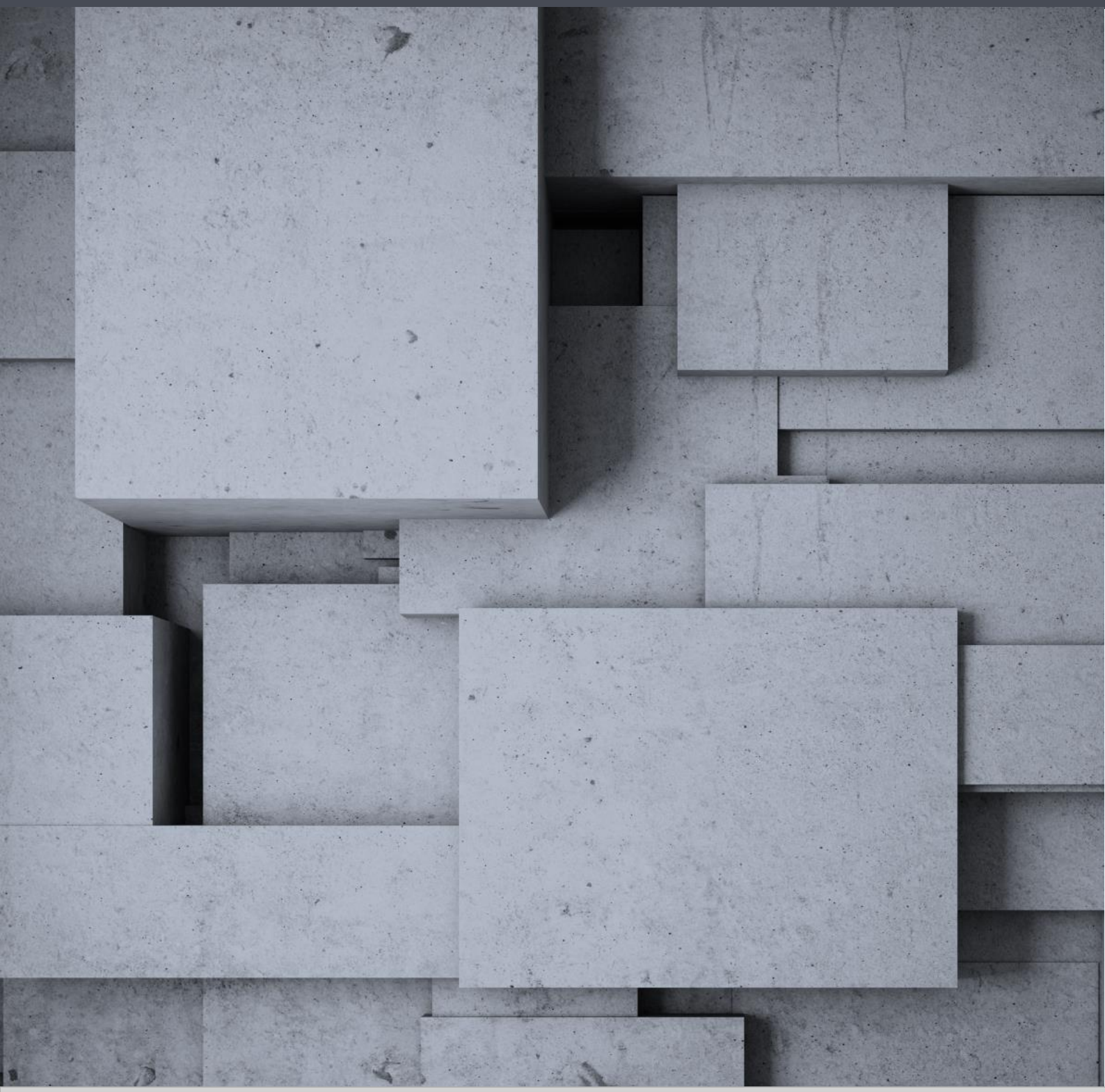
However, managing and executing Tests is not all. Earlier we have seen how reporting is a challenge for testing projects. Now, thanks to qTest Insights and Tricentis Analytics, you can create smart, customizable consolidated reports across platforms.

In-depth guides on how to configure and prepare your reports in both qTest Insights and Tricentis Analytics can be found on our Website and YouTube channel. In the User Tutorial for qTest Insights you will learn how to manage your data, perform quality, coverage and velocity analysis and create ad-hoc Dashboards.

In the Tricentis Analytics User Tutorial you will get an introduction to the software and its architecture before looking at how to modify sheets, reports and use Master Items. As well as User Tutorials there are a large number of short videos meant to address more specific single topics. for example, how to configure qTest manager or how to activate a specific license.

Remember to subscribe to be always up to date with our weekly video releases. Contact us, if you have any suggestions or requests for new videos! We hope you have enjoyed this short overview and are ready to dive into the Academy content! Good luck and happy testing!

TRICENTIS



PERFORMANCE SPECIALIST

PERFORMANCE SPECIALIST

Scale Continuous Integration in Agile Projects Objective

Welcome to the Test Performance Specialist video: Scale Continuous Integration in Agile Projects.

Let's quickly recap what continuous integration is. It is a practice in which developers, or members of a team, need to integrate their work regularly. Usually daily. Afterwards, the whole repository would be tested completely. This way, problems can be detected and resolved early to save time, effort and resources.

That is the theory anyway. Practically however, Continuous Integration faces many obstacles, one example being load testing. This is because it's almost impossible for Performance Specialists to keep up with Continuous Integration.

First of all, scripts are hard to write. Each load testing tool requires specialized scripting skillsets that are difficult to master. They also take a lot of time to create. So, this makes it hard to fit into the agile development cycle. Another issue is that load testing tools are usually not well integrated out of the box, with other tools in the DevOps toolchain.

And after all, Continuous Integration is expensive to run. Why is this? Tests are not optimized to run in the cloud and require too many permanent resources. The infrastructure must be re-scaled every time the requirement changes. That causes further delays.

Not to forget to mention that Continuous Integration demands considerable maintenance. Scripts are too brittle and require constant maintenance as the API infrastructure changes.

Not only that, you also need to maintain the servers used for the tests. Tricentis has an answer for those issues and roadblocks: Tricentis Flood

One of Tricentis Flood's main advantages is its simple and easy to use interface. Thanks to Flood, defining, creating and maintaining load tests becomes straight forward. Users can set up testing properties such as servers, number of nodes, and so on from the web interface. There is no need for coding knowledge!

And all of this can be done with a few clicks. Furthermore, you can reuse existing API TestCases from Tosca and convert ExecutionLists from Tosca directly into Flood Streams. Since the Module based testing approach is utilized, the need for scripting is reduced.

Flood enables you to have a higher degree of flexibility. If you are already proficient in JMeter or Gatling or other load testing methods, you won't need to change your skillset. Tricentis Flood supports a variety of open source load test tools as well.

That's great because you can start testing immediately without needing extensive upskilling! Tricentis Flood has another advantage. It offers out of the box support for leading application performance management- tools, single-sign-on providers, and more, which makes integration to your ecosystem simpler.

As you can imagine, this gives you a lot of flexibility when integrating with the tool of choice for your project or organization. The need for customizing or extra setting up for it to work is reduced.

But there are more advantages! Tricentis Flood will help you save costs! How? Usually, load testing requires a lot of physical servers which will cause an increase in infrastructure cost. Tricentis Flood, however, is a cloud based distributed Load testing platform. What this means is, all the load is generated using web services and will help you cut cost on purchasing physical servers.

On top of this, one of the main features of Flood is its scalability. Loads can be sent out on demand, there is no limit on horizontal scale. You can easily scale up to 10 million virtual users with just a few clicks instead of, again, having to add more servers.

No more waiting for purchasing and setting up processes which delay your project and cause losses, you can deploy the additional number of users immediately. All this is paired with reduced maintenance for both tests and infrastructure.

Let's focus on the tests first: Flood's unique Browser Level User approach reduces the complexity of script creation and ongoing maintenance. Now, you can spend time fixing performance issues, not fixing tests. The ability to integrate with Tosca makes it even better!

Secondly for infrastructure: Rather than constantly maintaining, updating, and troubleshooting expensive on-premise infrastructure, you can shift that cost and burden to the cloud provider.

With all the advantages mentioned, Tricentis flood is a great tool to increase your speed and productivity. Tricentis Flood will bring you one step closer to continuous integration.

You can find out more about how Tricentis products help you in your role, by viewing another video or take a look at how the Tricentis Academy enables you in our where to go next video.

Measuring Real User Experience and Identifying Issues

Welcome to the Performance Specialist Role Path video: Measuring real user experience and identifying issues.

As Performance Specialists, we would like to see how an application performs. Of course, that's easier said than done. There are many things we need to look out for. First and foremost, we want to see if the application can handle stress, or if it is reliable, as well as many other criteria.

It is one thing to test an application's uptime and response time, but how can we test if the users are getting the experience that we intended? Usually with load testing, you only send a large amount of network requests also referred to as a "load" to your application under test.

You then receive responses to indicate whether the connection was successful, and the task was carried out. However, nowadays, technologies have developed, and browsers have become more powerful.

For this reason, more and more web apps utilize the browser features to improve the customers' experience. An example for this is the use of JavaScript or CSS for dynamic websites.

When a website is not working as intended when it is under load, those features would also be affected. This means that if we only test applications on the protocol level, we don't know if the users are still getting the intended experience or features.

Additionally, a potential roadblock for our testing work is how we can effectively and accurately identify the issues. So, what do we usually need to test for?

There are 4 important metrics: The first thing is, of course, Performance, or the speed of our application. The second metric is the availability. Third is reliability. And finally, scalability.

Knowing that we should look for these metrics is good. But how do we go about extracting them from the tests and make sense of them? With traditional load testing methods of simply sending requests and receiving responses, it's not very intuitive.

Tracing exactly where the problem is takes time and effort. In many cases, when facing a long response time, you are left wondering, what happened to that request? What was actually returned by the server? What caused the problem?

Tricentis Flood offers both Browser-level Users and Protocol Level Users solutions to solve the difficulties just mentioned. As usual, you can do protocol level testing. That means generating load by simulating direct protocol level network requests.

Flood supports HTTP/S, SOAP and REST web services, FTP, LDAP, Mail, TCP, UDP and other protocols and webservices to fit your testing needs.

Here is where it gets more interesting:

With Flood Elements, you can also create Browser-level Users to test the whole app's performance. Including how HTML, JavaScript and more behave, not just the network and server infrastructure.

Simply put, Flood Elements interacts with a live application in the same way a real user would. The difference is that the simulation is repeated continuously and concurrently.

This allows testers to measure how many times a real person would be able to complete a transaction. This gives the ability to extrapolate the performance traits of the user experience. When elements run on Flood, each Element's instance opens a separate browser window.

You can even specify from which server region the browsers should be. Every part of the performance that a user will experience is considered. That includes not only network performance, but also on-page script performance and third-party scripts such as analytics or advertisement add-ins.

The result is a performance test that is much closer to the actual user experience. Another issue of Load Testing is the time needed to figure out where the problems occur.

All JMeter tests run on Tricentis Flood can now inspect the slowest successful transactions' request and response headers. This all happens live, whilst your test is running. Plus, you also can view request and response data returned by the server for failed transactions.

This offers you much better fault-finding capability under load. You don't have to configure anything else in your test plan. We do this automatically with some innovation inside JMeter and our own reporting engine.

This feature is available to all users. Another great thing about Tricentis Flood is the Real-time Dashboard. You can analyze your load test results easily with this view.

The necessary numbers are shown to you in real time as you inspect anywhere on the graph. From this view, you will be able to analyze concurrency, response time and transaction rates. There are more descriptive statistics for all load test metrics, down to individual transactions and labels.

We provide high resolution granularity for enterprise customers, which goes down to 1 second, so you can see the spikes in traffic as they occur. You can even review individual response times and error rates to narrow down the scope of investigation.

Deep dive into detailed error reports, response time statistics and more to determine the root cause of performance bottlenecks.

This way, you can pinpoint the potential bottlenecks or problems immediately and can fix them before the application goes live instead of making customers suffer outages which leads to dissatisfaction, monetary and brand value loss.

Creating reports and collaborating with the developers are also simple with Tricentis Flood. You can share the link to the dashboard and the recipient will have the same live view in their browser.

The test result can also be exported into CSV files for other reporting needs. To sum it all up, Tricentis Flood is here to make your life as a Performance Specialist a lot easier, and more productive. From the ability to test a web-app on the browser level to our intuitive live dashboard.

With it, Load Testing is no longer a daunting task. You can find out more about how Tricentis products help you in your role, by viewing another video or take a look at how the Tricentis Academy enables you in our where to go next video.

Performance Specialist - Where to Go Next

Welcome to the Performance Specialist Role Path video: Where to go next.

In the previous videos we have seen how Tricentis Flood helps Performance specialists achieve continuous integration thanks to its simple interface, reusability of components and low maintenance needs.

We have also covered how, thanks to Flood, you can lower the maintenance cost. Additionally, we have learned about Tricentis Flood's scalability, being able to simulate millions of users' activities with a few clicks.

Finally, we have seen how thanks to Flood and Flood Elements you have the possibility to test with real browsers and create robust dashboards to quickly identify issues. You can learn more about Tricentis Flood thanks to a wide variety of user Tutorials and Micro-learning content on both the Tricentis Website and our YouTube Channel.

Here you will find in depth guides covering how to best configure Tricentis Flood and how to best approach your testing challenges. Additionally, we offer many videos exploring best practices, configuration and administration of all products that make up the Tricentis offering.

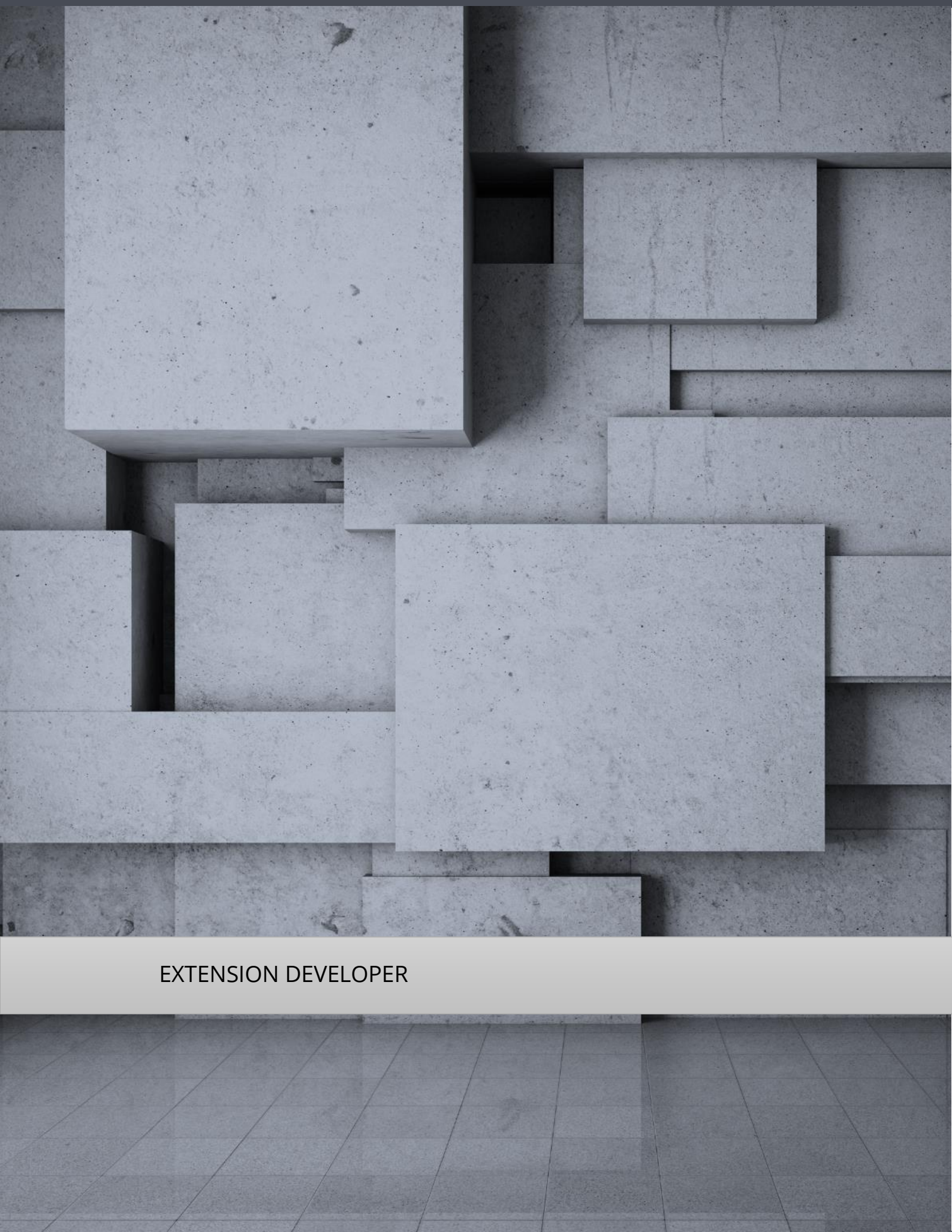
But as we have seen, one of Flood's strengths, lies in its integration with Tosca. This means that to master Flood, you will need knowledge of Tricentis Tosca as well. There are courses that will help you familiarize with Tosca. Thanks to Automation Specialist Level 1 and Level 2 you will learn how to automate your TestCases, deal with testing results and result versions.

These courses are respectively 3 and 1.5 days long. Remember to subscribe to our YouTube Channel to be always up to date with our weekly video releases. Contact us, if you have any suggestions or requests for new videos!!

Don't forget to visit the Flood website for more product information and the Tricentis Academy website for more course's information. We also recommend that you attend Tricentis events such as Accelerate, Users Conferences and others. These events serve as opportunity for you to meet the people behind the products and trainings and learn more about all the current, as well as future offerings.

We hope you have enjoyed this short overview and are now ready to dive into the Academy content! Good luck and happy testing!

TRICENTIS



EXTENSION DEVELOPER

EXTENSION DEVELOPER

Access and Steer Tricentis Tosca Via APIs

Welcome to the Extension Developer Role Path video: Access and steer Tricentis Tosca via APIs. In software development, builds must be constantly tested, and committed to the repository, to achieve continuous integration.

Ideally, the tests are done overnight to save time. If they aren't done in time, they delay project delivery. However, it is not very convenient or cost efficient to have an employee present at night just to trigger those tests, which need to run regularly. The answer for this is to schedule the tests to run automatically.

Oftentimes, the testing solution will also need to be integrated with other tools such as Jenkins, Visual Studio Team Services, and others. They can be the tools that trigger the tests or store the testing results. Yet, for many testing suites, even after integration, scheduled and recursive test executions still need to be done under supervision.

For example, sometimes, due to system authentications, it's difficult to open and access the test tool's objects, especially through the command line. This causes difficulties in automatic execution of tests or tests failing without supervision. As a result, teams have to spend more time and effort on testing. This is not a problem with Tricentis Tosca, thanks to its APIs.

It is possible to integrate Tosca with other tools in the DevOps tool chain and have them access and steer Tosca as well as its elements, even without having to use the Tosca UI. This ability to access the Tosca API helps simplify the process, as you have direct access to things you need. This also cuts down on the time needed to build and run automated test executions.

There are 3 ways to access and steer Tosca as well as Tosca elements with the APIs. The first is using TC Shell. With this method, the Windows command line is used. We can access Tosca Objects and perform functionalities without having to open the Tosca GUI. This method is how we can schedule test executions.

The second way is using the TC API. Like the last method, this also helps access Tosca objects without opening the Tosca GUI. This is how you can access Tosca via Visual Studio, using C#. Last, but not least, is using the TC REST API. This API is platform independent. With this, we can send requests and responses to Tosca. The TC REST API is mainly used to integrate Tosca with other tools, which is essential for DevOps. This is also the technology behind the Tosca Connect plug-in.

The various APIs will fit any need you have for steering Tosca, from scheduling execution to integrating with other tools, saving you a lot of time and effort in continuous integration. You can find out more about how Tricentis products help you in your role, by watching another video or take a look at how the Tricentis Academy enables you in our where to go next video.

Reduce Customization Effort and Improve Usability

Welcome to the Extension Developer Role Path video: Reduce customization effort and improve usability.

The code used to build applications comes in all shapes and sizes. This means that while analyzing an application under test, it can be challenging to identify and steer certain complex controls or perform special difficult tasks.

Sometimes, it might even be impossible for a testing tool to do it out of the box, depending on how the developers have coded the application. This is when an Extension Developer is needed. Usually, in a testing project, the Extension Developer is not a member of a testing team.

They are Software Developers, asked to temporarily join the team to create extensions that helps testers deal with the difficult controls or customizations. For this reason, the extension developer might not be as well versed in the System Under Test as the other members in the team.

Thus, when they try to analyze the application, it will take them time to understand the business functionalities and interface or the expected user behaviors. Because of this, it can be tough for an Extension Developer to write scripts that can steer the application the way the testing team needs.

Another problem that extension developers may experience is that they need to write a script or an extension every time the same difficult control is used. Then, when the SUT or test updates, the Extension Developer has to update those extensions or scripts individually as well.

It might even cause roadblocks for the testing team, as they are dependent on the Extension Developer's availability every time something is changed. With Tricentis Tosca, special controls can be scanned and steered with customizations and challenging automation tasks can be performed using Special Execution Tasks.

They can be easily created using the existing TBox framework, saving you a lot of effort with coding your own framework from the ground up. Once a customized control is created and deployed, it can be used to recognize and steer the same type of control at any place in the application for an unlimited number of times. Special Execution Tasks can also be reused across different projects or teams wherever relevant.

It's also easy to use. Once the customization is ready, the Automation Specialists will only need to drop the customization files into a folder and then they can go directly to scanning. No extra set up or configuration is needed. The files can also be shared around easily, leading to faster testing.

With the latest Tosca versions, recompilation of Customizations and Special Execution Tasks is no longer required during Tosca upgrades. This means that testing teams can save even more time and effort! Using these same principles, it's very easy to extend the functionalities of Tosca Commander and its interface by creating your own Add-On to the Tosca UI. You can create your own customized Ribbon item, Context menu item or Drop-down option in the Tosca Commander GUI.

This helps user to reduce the effort for repetitive tasks and also add extra functionalities. For example – if your team constantly needs to check the workspace summary of the

project, you can create a Ribbon AddOn to access this functionality faster. After customizing, you can simply click on the new button to accomplish the task.

If you want to see the properties of some specific Tosca Objects/Folders, the extension developer can create a Context Task so we can get to the information directly from right-clicking the item. For example, you can check what type of engines are used in a particular Module folder. If you want to drag and drop an object into a section which Tosca normally does not support the drop function, you can achieve it by using Drop AddOn.

An example for this might be dropping an entire Module folder on the TestCase section to create a TestCase from the contained Modules. With Tosca being highly and easily customizable, you can really set up your environment to reflect the best and most efficient processes for your teams.

You can find out more about how Tricentis products help you in your role, by watching another video or take a look at how the Tricentis Academy enables you in our where to go next video.

Extension Developer - Where to Go Next

Welcome to the Extension Developer Role Path video: Where to go next.

After the previous videos, you know have an idea about how you can customize Tosca to work with many specific requirements and environments. First and foremost, we know that, in case there are specific controls that Tosca cannot steer out of the box, customizations can easily be made.

Moreover, the customizations are easy to use and maintain. Besides, you can also customize the interface of Tosca itself to make it more comfortable and fitting to your testing needs. Another feature of Tosca is it can be steered by and integrated into a tool chain through various APIs. The Tricentis Academy offers you various courses that will help you to go further on your path as an extension developer.

We recommend you first to go through the Automation Specialist level 1 and 2 trainings, as they will give you the basic knowledge of how Tosca works and how control are recognized and steered. These courses are available both online and as instructor-led training.

The Automation Specialist Level 1 course is a 3 days course. The Automation Specialist Level 2 will take you 2 days to complete. Next is the Automation Engineer Level 1 course. With this course, you will familiarize yourself with more difficult controls as well as how Custom Controls are deployed and used by an Automation Specialist. After completion of AE1, you will then learn how to actually create the customizations in the 4-day Automation Engineer Level 2 course.

This course will enable you to build customized controls that Tosca can scan and steer, enabling you to steer specific custom controls in your HTML application using the existing TBox framework. You will also learn how to develop special execution tasks, for example verifying existing Buffer values. This training is available online as well as in classroom with an instructor.

As this course requires object-oriented programming knowledge, you will need to partake in an exam that checks your understanding of the C# programming language. If you want to build on the customization skills you learned in AE2, you can take the Extension Developer for Desktop Application Course.

In this course, you will learn about customizing Tosca to steer .NET, Java and UIA applications. This 2-day course is available as an instructor-led training course. If you need to integrate Tosca with other software in the DevOps toolchain, you will need the Tosca Integration Developer course. This 2-day course is also available as both an online self-based learning and in classroom.

The training course contains four lessons, that introduce the Tosca APIs and describe how to implement them in a real-world context. The topics covered: TC API, TCAddOn API, TC Shell and TC REST API. In addition to the courses I just mentioned, we occasionally offer public, instructor-led trainings on specific topics such as Custom

Controls (HTML, .NET, Java and UIA), Special Execution Tasks or Tosca APIs. Please check out the information on the courses on our website or send us an email at academy@tricentis.com.

For more learning and information about Customizations and Integrations, please check out our videos on the Tricentis Academy YouTube Channel. You can also find more information on our support portal. Furthermore, we provide constant webinars and user tutorials.

You can join or watch the on-demand recordings on the resources section of our website. Please also check out the Tricentis Academy website for more information on any Tricentis-related course.

We hope you have enjoyed this short overview and are now ready to dive into the Academy content. Good luck and happy testing!

TRICENTIS

TEST ARCHITECT

TEST ARCHITECT

Facilitate Effective Multi-User Collaboration

Welcome to the Test Architect Role Path video: Facilitate effective multiuser collaboration.

As a Test Architect you have to deal with a number of issues. One of them is repository growth. Your repository starts growing when teams and members start adding more artifacts to it. If this is not handled well, repositories can become unwieldy, disorganized and hard to work with.

One has to keep in mind that artifacts might be organized differently by teams. That can cause confusion or duplication of work if someone cannot find an existing artifact and has to re-create it.

Therefore, you have to check if the overall repository structure is well organized. Another challenge Test Architects must master is the collaboration between teams. Teams need to know which artifact can or cannot be altered by which team.

To add more headaches to Test Architects, they need to think about the confidential data that has to be handled. For instance, HR or user data. A lot of times only certain teams should have access to sensitive data. Or even certain project components.

Tricentis understands these problems and works constantly to ease them. But let's explain how: Within Tosca Commander, there are several built-in functionalities that enable Test Architects to build and customize their test portfolio.

Let's go over some of these features: The User management section: Here you can create different types of users and groups with different rights and accessibility. So, Test architects can immediately manage their users, create and deactivate users and create passwords.

Additionally, they can see who belongs to a certain group or project. Even which rights they have. Another useful feature is Owning and Viewing rights: a user group can be set to own or be able to view certain objects. Here you can grant or restrict viewing or access rights for objects and folders toward a certain group of users.

Now let's check out Component Folders: This is really helpful when working in a multiuser environment. Each team or project can have their own folder structure consisting of component folders. But that is not all!

Check in, check out is another function where you can check a folder or an artifact out and no one else but you can work on it If you are done, you can check your work back in and others can work on it again

What is also vital is Versioning: Object versioning allows you to log and comment on any modifications that are made to objects in a multiuser environment. Additionally, you can switch back to any revision.

This is how you can trace and reverse any changes that were made to an object. Custom properties are a very powerful option: These allow you to easily define properties that your project or your team needs. That way you can attach more information to artifacts. For instance, CreatedBy or CreationDate could indicate if the artifact or folder has been reviewed or not.

This way teams can create any properties they need to keep track of their workflow. Your team members will need to know the status of the TestCase projects, for example: Which TestCases are being worked on, which ones need to be picked up, or which one is done. This can be Simply achieved by adjusting the setting “WorkState” of the TestCases!

And that’s how to facilitate effective multiuser collaboration! You can find out more about how Tricentis products help you in your role, by viewing another video or take a look at how the Tricentis Academy enables you in our where to go next video.

Increase Your Risk Coverage with Risk-Based Testing

Welcome to the Test Architect Role Path video: Increase your risk coverage with risk-based testing.

Software is never perfect—but how can you stop the most business-critical issues from evading your testing efforts? You could try to extensively test every possible path each time your application changes… But that's rarely realistic!

A much more feasible approach is to focus your limited testing time on building the most efficient set of tests that deliver the greatest risk coverage. What risk-based testing does is it aligns test planning and execution to business priorities.

Depending on business decisions, certain features get higher priorities in the testing process. For example, being able to pay for an item, is tested before being able to add a product review. It shifts the focus from Number of Test Cases to Risk Coverage.

With risk-based testing, we try to maximize risk coverage with the fewest test cases. Not only is it important to maximize risk coverage with the minimum number possible of Test Cases, is also important to know what tests to prioritize, and this can be very challenging.

Imagine there are hundreds of test requirements. Let's assume they are created and kept in Excel or other non-testing specific documenting software. In that case Test Architects will need a lot of time to see and work out if a requirement has high priority and whether it's already covered with a sufficient number of tests.

There are still many test projects out there fielding a big number of test cases, but they fail to cover the desired percentage of the total risks. Tricentis Tosca gives you the ability to track requirements right out of the box, thanks to the Requirement section.

With Tricentis Tosca each requirement is given a weight based on the function's potential frequency of use and the damage it might cause if there's an error. These values can be decided on very easily by business analysts within the organization without needing to understand anything about testing itself.

The test architect can then use the calculated weight to determine where to start testing to maximize risk coverage. Or as we say: test the right things first! Tricentis products give you an organized overview of all the requirements, each with their own weight.

Moreover, in Tricentis Tosca, everything is linked. The Execution section, where your tests are executed and logged, can be linked with the Requirement section, where your business requirements are defined and kept. With this ability, you will see immediately if a requirement is sufficiently covered and what the outcome of the TestCases that are related to said requirements are.

QTest also allows you to link your test cases to requirements.

In this way it is clear to see how much work has been dedicated toward covering higher risk areas Additionally, you can see the work progress: For instance, if the expected work was met or how much else is left to do.

To summarize: This risk-based approach to testing ensures that we have the minimal amount of risk that a release of our application will result in losses or damages. And we can achieve this with the minimum possible number of targeted test cases

You can find out more about how Tricentis products help you in your role, by viewing another video or take a look at how the Tricentis Academy enables you in our where to go next video.

Leverage a Single Platform for The Continuous Testing Process

Welcome to the Test Architect Role Path video: Leverage a single platform for the continuous testing process.

To become a Test Architect of a platform you will need a lot of knowledge and experience in:

-The platform itself and

-the Infrastructure capabilities, that is, how every part and application integrate with each other.

This might be difficult for many reasons. For example, different scripting requirements for each platform will result in testers having to constantly upskill. Additionally, changing infrastructures require constant adaptation of your testing strategies and team. All this is obviously going to be very time and resource heavy.

To address these problems, Tricentis products leverage the same business readable interface across ALL TestCases, regardless of the technologies tested or methodologies used. Therefore, there is no demand anymore to upskill every time you need to test something new.

This, in turn, allows testers to dramatically cut the time that usually would be spent upskilling. Within the Tricentis platform, the intuitive business interface is used in conjunction with the Model Based Testing approach. This combination helps to reduce upskill times across all supported technologies.

Instead of having to learn to script tests for a platform, you can simply create Modules by scanning the applications. These Modules will then constitute the building blocks for your Test Cases. The process is the same, across all supported technologies and testing methodologies.

Now you have extra time to strategize the next steps in the testing projects. On top of all this, the Tricentis Testing Platform can also combine the power of model-based test automation and open source tools. This allows you to achieve continuous testing across DevTest & IT architectures. Overall, Tricentis products support more than 160 technologies and enterprise applications.

We also support many languages and technologies:

HTML5, php, Java and C# .NET and many more. The Tricentis Testing Platform also fully supports the testing of packaged applications such as SAP, Oracle Salesforce, ServiceNow, to name just a few. That means you can use Tricentis Tosca to test almost any available application. Regardless of the programming language they are built on.

Tricentis products can also be integrated with open source tools like Selenium, Appium, Gatling. Finally, The Tricentis Testing Platform fully integrates with ALM platforms or CI DevOps tools like JIRA, GitHub, Rally and more. That means you or your project or your team won't have to switch away from your proven workflow.

Tricentis offers you a full and comprehensive platform, fitting all your continuous testing needs.

As mentioned, Tricentis Tosca uses a model-based Test Automation approach. With this approach you can automate UI and non-UI tests, across browsers, on mobile devices as well as test the integrity of data for business intelligence!

With our Flood offering you can automate tests on the performance of a service as well. You can optimize your test projects even more by using the Risk-based testing approach, which is supported out-of-the-box with Tricentis Tosca.

For test data, we offer a Test Data Service to manage and connect your test data to your testing repository. You even have the ability to generate synthetic test data or to mask your production data to comply with privacy requirements using Tricentis' TDM Studio.

Of course, you can build all of your automated tests using Tosca. But, what if some of your services or application UIs will be unavailable during certain phases of the test process?

You can easily implement Tricentis' Orchestrated Service Virtualization and API testing, to help you test faster and earlier in the process. Tosca can be used as well in conjunction with Tricentis LiveCompare.

LiveCompare will help you analyze the impact of changes for your SAP system. To keep track of testing artifacts, across platforms and teams, Tricentis offers qTest Manager.

qTest Manager is an Agile Test Management suite created to help you track, organize, and report on testing activities across the enterprise. With scalable, in-sprint test management for open source test automation, the exploratory testing engine and extensive ALM real time integration, keeping track of your project has never been easier.

Tricentis also supports exploratory testing with qTest, through qTest Explorer. However, the tests you perform are only as good as the data you can gather from them.

For this Tricentis has introduced Analytics. Tricentis Analytics gives you a comprehensive yet intuitive dashboard view of all your test results. From there, you can also directly generate reports.

qTest Insights gives you extensive information on the Test you have performed in the Tricentis qTest Testing Suite. You will no longer need to be concerned with how your testing process stages flow into one another. Tricentis tools can be integrated to make up the entire continuous testing cycle.

The ability to set up risk-based requirements to provide maximum risk coverage and test automation are core features in Tricentis Tosca. Test Data Management and Service Virtualization are components that are designed to work seamlessly with Tosca.

Tricentis also offers the possibility to perform real user experience tests, fully integrated with Tosca, thanks to Tricentis Flood. To sum it up, with our products you get a full end to end continuous testing platform!

You can find out more about how Tricentis products help you in your role, by viewing another video or take a look at how the Tricentis Academy enables you in our where to go next video.

Improve Your Agility with CI in DevOps

Welcome to the Test Architect Role Path video: Improve your agility with CI in DevOps.

First, let's talk about DevOps, and why it is being widely adopted in many organizations' software development projects. Up until recently, software development had mostly been a linear process. A team would receive the requirements, then start on designing, implementation, building, then testing, and finally, releasing the software.

There are, however, many problems that can occur with this traditional method. First is the large time gap between when the team receive the requirements and the software release, since the team only gets the requirements at the beginning of development period.

Then, it's also not possible to revisit the design once the project is in a later phase, such as implementation or testing. Once a design is chosen, the project would need to stick with it with no flexibility for change.

Another problem is that testing a project late into development process can be very costly. Catching problems and bugs early on is desirable as the damage is minimized, and the time needed to fix bugs is low.

Fixing a bug late in development process takes more time, resource and can cause extra problems. The rigid development process also allows very little time and room for change. This is difficult because the market or stakeholders' needs might have changed a lot from the time development starts until it ends.

This is where the DevOps method comes in. Instead of having only one big linear process, teams split development processes into smaller cycles, releasing smaller features.

Each of those development cycles has their own planning, designing and testing phases, giving developers a constant source of feedback. With DevOps, operations and development engineers have an agile relationship. They participate together in the entire service lifecycle, from design through the development process to production support.

So, what does Continuous Integration have to do with DevOps?

Continuous integration is a practice, in which developers or members of a team need to integrate their work regularly, usually daily. After this, the whole repository is tested completely. In this way, problems can be detected and resolved early, saving time, effort and resources.

Continuous integration is essential to Dev Ops. Thanks to the ability to get immediate, quality feedback from the work that was completed, and how they integrate with the existing repository.

However, there are still some limitations:

First, the ability to test and integrate code into a repository several times a day. With manual testing, this requirement is almost impossible to meet. There is no possible way to test extensively every day, as the required resourcing would be too high. This might cause teams to integrate untested code into the repository.

The second limitation is the ability to verify quality by triggering tests after each build. With most test automation methods, it is difficult to do this. This is because systems tend not to be integrated together. Therefore, someone will have to do this integration or trigger the test runs manually.

Finally, there may be difficulties in collaboration between different teams during the testing process. This can happen due to the different specialties and skillsets of each team. Let's talk about how Tricentis can help you solve those problems.

As you know, it is essential to integrate your functional tests into your automated build, all within a continuous delivery environment. For this, Tricentis provides you with Tosca Continuous Integration, or Tosca CI.

With Tosca CI, you have the possibility to trigger your Tosca tests remotely via your build server and receive back a well-formed test result.

You can now integrate agile testing into an automated delivery pipeline through test automation and risk coverage (dev). Meanwhile, test environments (ops) can be made available on demand. Once the code is ready, with the help of CI servers such as Jenkins, we can start the Unit Tests, get results out and provide feedback immediately.

This process can happen at any point in the Software Development Life Cycle. This means that developers get the right feedback, at the right time, from the right stakeholder.

In a real-world scenario, the focus is to trigger execution without having any manual interventions and get the get results back. So, with the help of Tosca Continuous Integration, you can trigger the execution of your automated Tosca TestCases without user input.

Depending on your IT infrastructure, you can implement Tosca Continuous Integration in one of the following ways: Run the test execution via Remote Service, i.e. on a separate execution workspace. Use Tosca Continuous Integration together with Tosca Distributed Execution. In this case, several Tosca Distribution Agents will execute your tests, which speeds up the process.

This helps to empower your Continuous Integration & Delivery journey by minimizing testing time and effort. Moreover, Tosca CI supports many different tools in the DevOps toolchain.

This Flow Chart Explains the Continuous Integration pipeline from planning, Coding, to deploying and monitoring. As you see, Tricentis fits right into the whole process by being integrated into every step of the way.

This is how testing is no longer a roadblock for continuous integration. Of course, it is also essential for DevOps that everyone is able to collaborate effectively and efficiently. If there is a problem in communication, then the concept of bringing development and operation together has already failed.

Tricentis makes it easier for various groups in a project to collaborate on testing. Let's talk about how we can help teams collaborate during different stages of development.

First, once the code is deployed to the code repository, developers can use the API Scan to directly do unit testing. They can also simply create API Modules and send them over to QAs to create Test Cases in Tosca.

In the next stage, the Testers will do functional testing with the help of Tricentis Tosca. If the Test results pass, the build goes to deployment or a production environment. If they fail, the build is pushed back to the developers to analyze the problems. For the Business users and product owners, it's important to have a business Readable UI.

Thanks to this, they can see exactly what is being tested without the need for any coding knowledge.

This is how they can contribute valuable inputs and feedback at any stage of product development and testing. Of course, we also offer detailed test reports with our reporting and analysis tools.

To summarize: DevOps is a new and exciting method of software development, in which Continuous Integration plays a big part. However, there are still roadblocks for continuous testing and collaboration.

Tricentis aims to solve those problems, so that teams can focus on what matters: the quality of the product.

You can find out more about how Tricentis products help you in your role, by viewing another video or take a look at how the Tricentis Academy enables you in our where to go next video.

Test Architect - Where to Go Next

Welcome to the Test Architect Role Path video: Where to go next.

In the previous videos, we discovered how Tricentis offerings can make your life as a Test Architect much easier.

First, we introduced you to how multi-user collaboration can be achieved within Tricentis Tosca. Then, we went through how Tricentis Tosca helps you to optimize your test coverage with the Risk-based testing methodology.

And finally, we learned how the Tricentis suite can fit into your continuous testing process and improve your agility in continuous integration in DevOps.

In order to become a Tricentis Test Architect you will have to achieve the highest skill status in automated testing. This proves that you have a firm grasp on Tricentis Products and their Best Practices. It also includes that you can lead testing teams using these products.

The required courses a Tricentis Test Architect fulfills are:

- Automation Specialist Level 1 and Level 2. Which are respectively 3 and 1.5 days long
- Test Design Level 1 and Level 2 these are respectively 3 and 1 days long
- Automation Engineer Level 1 which will need 2 days to complete.

All of them are available online as self-paced courses.

With those core courses, you will get all the basic skills needed to create and execute automated UI and non-UI TestCases using Tricentis Tosca based on the Risk-based testing methodology and more.

Automation Specialist for SAP or qTest Specialist are additional courses on other specific features of the Tricentis family.

Furthermore, being a Tricentis Test Architect requires that you have personal working experience with the Tricentis platform, as well as the Test Architect Certification.

Once you have the experience to apply the knowledge you have gained of Tricentis products in real life testing projects, you can go for the Test Architect certification.

Within the Test Architect course, the following materials are provided to you:

- Project Planning
- Continuous Integration
- Repository Configuration
- Tosca Best Practices
- Requirements and Risk
- and Analytics and Reporting

These materials will help you strengthen your existing knowledge.

This certification is also the highest endorsement of your skills and a proof of your achievement in testing with Tricentis products.

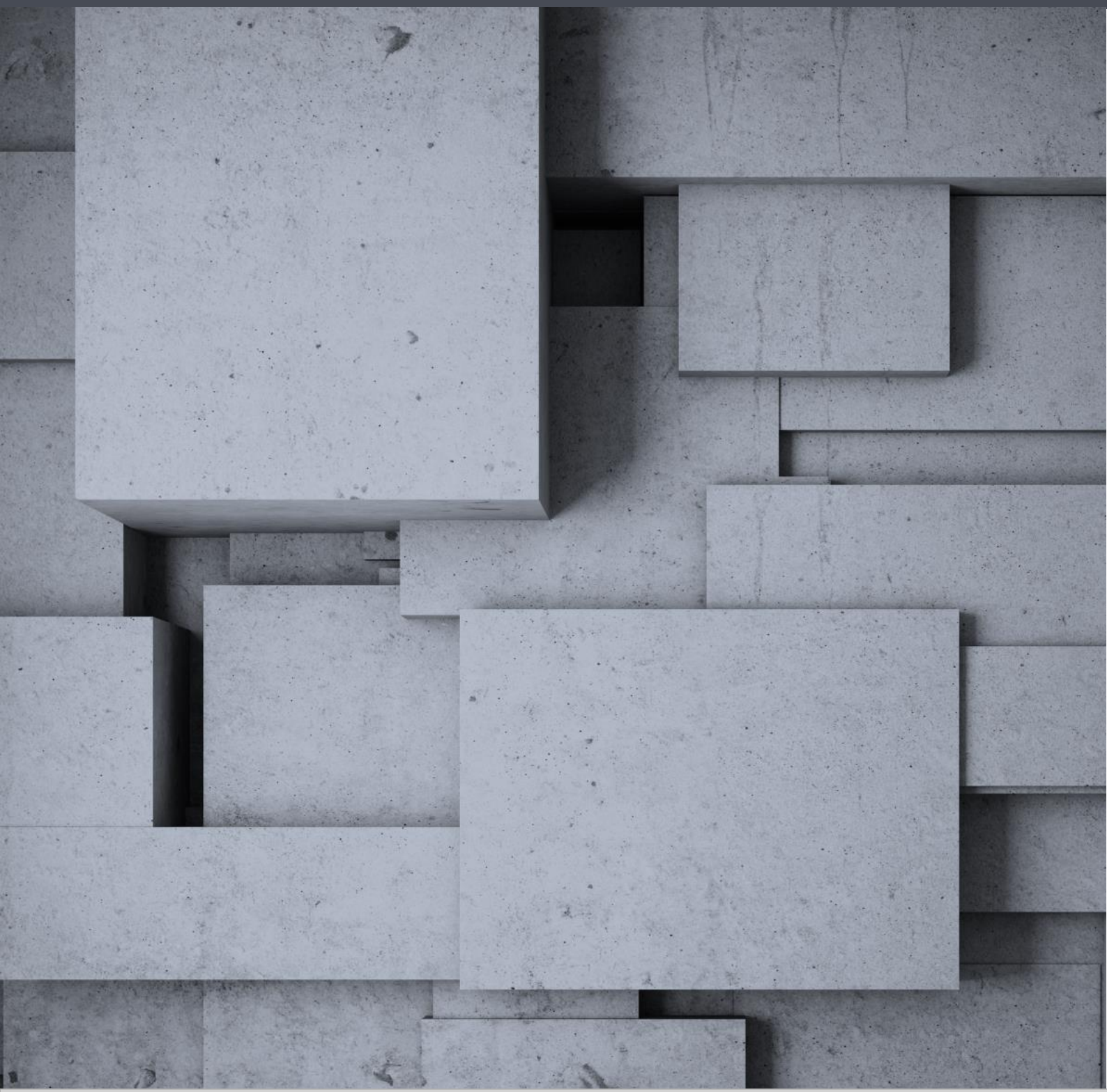
Besides the courses, Tricentis offers many additional ways to expand your knowledge and to keep up to date with our products. Check out our Marketing webinars, where the newest updates and functionalities are announced.

Another source is our User Tutorials, where we go in-depth about features or functionalities. And finally, you can check out the Tricentis Academy channel on YouTube, where we have a large number of short videos meant to address more specific single topics.

Please visit us on our website to find out more information about the offerings from Tricentis Academy. We also recommend that you attend Tricentis events such as Accelerate, Users Conferences and others. These events serve as opportunity for you to meet the people behind the products and trainings and learn more about all the current, as well as future offerings.

We hope you have enjoyed this short overview and are now ready to dive into the Academy content! Good luck and happy testing!

TRICENTIS



TEST MANAGER

TEST MANAGER

Centralize Data Gathering and Improve Visualization of Reporting Data

Welcome to the Test Manager Role Path video: Centralize Data Gathering and Improve visualization of reporting data.

As you probably know, a Test Manager's life is never easy. Let's take a medium complexity project for an example. A project of this scale is most likely using a variety of tools to track, execute and report on the different phases and types of testing.

Your team is probably using an ALM to keep track of the artifacts. A CI or DevOps tool to execute and track the tests. And maybe even a third tool to manually merge together all results. Matching all the data together and providing an accurate report is a challenging task. This sometimes can result in gaps in the project. The data might be overlooked or not addressed quickly enough because it was lost among the noise. And even if the data is in the right place, other problems may come into play. Many data gathering and management engines lack the capability to visually display information and reports in a clear and understandable way. But you need a clear and "to the point" visualization of a situation or problem if you are talking to higher management or business-oriented people. They need to see the pros and cons at one glance.

Tricentis has addressed these problems in both qTest and Tosca suites. Both solutions fully integrate requirements with all other elements of a testing project. Such as test cases, execution data and defects. A complete coverage of your whole testing project is ensured. However, the data you gather is only as good as the reports you use to display them. For this reason, Tricentis offers qTest Insights as well as Tricentis Analytics.

qTest Insights is a powerful, built-in, reporting engine for the qTest Suite. qTest Insights is a self-service business intelligence tool. It consolidates, manages and analyzes all your testing data along with your integrated ALM issues within qTest. This, united with the capability of fully customizing Dashboards, filters, charts and visual heat maps will allow you to paint a comprehensive picture of your testing project in qTest.

Tricentis Analytics brings together all test results from Tricentis Products. It not only integrates with qTest and Tosca, but also Tosca Flood and all satellite features. Results are merged into a single view of test progress across manual steps, functional and API automation, and performance testing. Tricentis Analytics is particularly targeted for those projects who use both qTest and Tosca to fully test their products. Analytics provides portfolio level visibility through a predictive, BI reporting engine. It consolidates, manages, and analyzes activities across the Continuous Testing Platform for your enterprise agile and DevOps initiatives.

Tricentis Analytics is built to accommodate more complex reporting needs for enterprise users thanks to its ability to handle huge volumes of testing data -- everything from running SAP test automation to executing popular open source frameworks at

enterprise scale. Thanks to Tricentis analytics, you will be able to realistically assess your release readiness by instant evaluation of current and future business risks. Risks that are associated with the health of your application -- so you can release quickly and with confidence.

You can generate reports on qTest and Tosca and also simultaneously on all the other integrated systems. Additionally, you can build traceability and coverage reports from your integrated DevOps tools that relate to the Tricentis Platform (e.g. Jenkins, Jira Software, Selenium). All the reports are easy to create thanks to an intuitive drag and drop interface. In addition to the pre-built reports, Tricentis Analytics gives you the ability to build custom reports. This way you can track testing KPIs and forecast trends to measure team performance. To measure testing across distributed teams you can combine as many sprints, releases and projects as you wish.

The advanced data visualization includes distribution plots, Pie charts, Histograms, Tree maps, Interactive gauges, Combo charts, and much more! Thanks to Tricentis Analytics you will be able to accurately assess all key factors that might influence a software release like Risk reduction. You can see if teams can increase coverage and reduce risks. You can also view time to Resolution: That is understand how developers and testers are working together to reduce defect turnaround time. There is Speed to Market: To Determine how testing efficiency is improving from release to release. Additionally, Test Automation Progress: so you can assess and predict test automation rates across teams.

Now you have seen how Tricentis Analytics helps you to test smarter! You can find out more about how Tricentis products help you in your role, by watching another video or take a look at how the Tricentis Academy enables you, in our where to go next video.

Keep Track of Artifact Versions

Welcome to the Test Manager Role Path video: Keep track of artifact versions.

As a Test Manager you need to have a correct and clear overview of your Test projects. Therefore, it is very important to keep track of the Execution history for your Testing objects. Otherwise it will be nearly impossible to track the project progress. And it quite likely increases the redundant work. If the application you use does not support versioning, it can be challenging to keep track of the changes that occurred over time. This also may result in a loss of data. Additionally, the lack of test execution version history might lead to generating incorrect reports with incorrect test results.

And finally, not having access to test artifacts older versions might hinder your ability to revise your Test data or Test artifacts if needed. With Tricentis Tosca and Tricentis qTest, all of these issues are taken care of. Both suites enable access to older versions of all the test objects. Tricentis Tosca helps you to keep track of your test Execution history and updated Requirement sets. This way you will be able to achieve Risk-Based Test automation. You can track modification history information on Modified attribute values or Newly created objects. You can even access objects after they have been deleted or moved between and within two clusters. Additionally, you can also track modifications to a sequence of ordered objects or a list.

Every time an object is modified between check out and check in, these modifications are given unique revision numbers and can be easily accessed. This option is always enabled in a multiuser environment, so there is no chance to accidentally lose old versions. The modifications can be checked in with added comments. So that means you can recover deleted objects, revert to old versions, or export older versions as well.

Another advantage is that Versioning can be tracked over the whole project in the repository, with checked in changes, as well as the local modifications. As mentioned before, versioning is a central part of Tricentis Tosca and Tricentis qTest. By using qTest manager you can keep track of each Test object in your testing project. From requirements to the test execution and Defects. After a single change, qTest will automatically update the test object version number. Additionally, qTest automatically creates new versions of artifacts after each iteration and approval. At any time, previous approved versions can be viewed, so that it is possible to keep track of the past changes.

To sum it up: both Tricentis Tosca and Tricentis qTest are built to support versioning. This means that you will be always able to access test execution version histories. Thanks to that, you can now generate correct reports based on correct test results! With Tricentis products you will access correct and effective revisions of data.

You can find out more about how Tricentis products help you in your role, by watching another video or take a look at how the Tricentis Academy enables you in our where to go next video.

SAP Update Impact Analysis

Welcome to the Test Manager Role Path video: SAP Update Impact Analysis!

As you may know SAP is the most used application in larger enterprises. According to the Oxford Economics SAP analysis, 77% of the world's transaction revenue passes an SAP system. So, if it comes to testing updates of an SAP application, you can probably imagine the difficulties that come along. Not only do you have to determine where the changes were. But you also have to consider on how they affect your testing. Many of these changes occur via slow and risky processes that you will encounter, such as the migration of support packs, enhancement packs and SAP HANA or S/4HANA transformation. At times you might even face Emergency fixes or custom releases. In these cases, many companies will adopt the test everything approach.

This is, however, not only inefficient but ineffective. If you test everything in the execution testing suites, some artifacts are lost during testing. And redundancy will occur. This approach might even become outright unfeasible on larger systems. In these cases, traditional analysis solution as well becomes too complex and time consuming.

To overcome these problems, Tricentis offers LiveCompare. When using LiveCompare, the testing time of enterprise SAP updates is reduced by up to 85%. While achieving 100% risk coverage! (How?) Well, Tricentis LiveCompare provides AI-powered impact analysis for SAP updates. LiveCompare exposes the risks to your critical business processes lurking in any standard or custom change, identifying the most-at-risk areas that need to be tested. LiveCompare's AI-powered impact analysis, compares your current SAP system with a proposed custom release or standard SAP update. It analyzes all the differences between the versions that impact your business processes, system integrations, custom code, security and governance. This means that with LiveCompare Enterprises no longer need to waste time and resources running their entire regression test suite for each update to their standard or custom SAP code.

But how does Tricentis LiveCompare fit into the Tricentis suite? It starts with the Enterprise Sap update where each regression suites needs to be tested. Then with the help of Live Compare you can do the Smart impact value Analysis on the existing Testing Suite. This will provide 85% of the risk Coverage and the most impacted use cases.

Now, thanks to the seamless integration with Tricentis TOSCA you can perform the execution of the most impacted use cases and get the results and reports out of it. Then, again, with Tricentis LiveCompare, you can perform impact analysis on the most impacted use cases, regardless of them passing or failing. Then, you can generate the reports out of it. Additionally, thanks to the integration with Tosca, you will be able to identify the test gaps and fill them in for a solid base for future regression SAP tests. Now you are ready to release your SAP update with ZERO defects.

Let's quickly summarize the benefits of LiveCompare: LiveCompare INCREASES RELEASE VELOCITY

You can ship SAP updates faster by testing 85% less and achieving 100% test coverage.

LiveCompare:

REDUCES COSTS

You will reduce testing costs and resources by automatically generating the most-at-risk areas that need attention. And this without the need to test everything.

DECREASE BUSINESS RISK

Through LiveCompare's AI powered impact analysis, you can decrease the level of risk. The risk that is associated with SAP development changes.

ACHIEVE HIGHER QUALITY OUTCOMES

Generate higher quality software and start your path to zero production defects with focused test automation.

Now that you know how Tricentis helps you test SAP applications updates efficiently watch another video on how the Tricentis platform meets your needs, as a Test Manager. Or jump into the where to go next video for a guide on how we enable you in all these features.

Test Manager - Where to Go Next

Welcome to the Test Manager where to go next video.

In the previous videos we have seen how thanks to the Tricentis Tosca and qTest suites you can now access environments where all testing artifacts are linked. Additionally, we have discovered how both, Tosca and qTest, fully support versioning of artifacts. This obviously makes your life easier when trying to keep track of changes. We have also covered how, thanks to Tricentis Insights and Analytics, you have access to fully comprehensive data reports to help you drive your project forward. Finally, we have seen how, thanks to Tricentis LiveCompare, you are able to speed up your SAP update testing by 85%. All this while still achieving 100% risk coverage.

The Tricentis Academy offers you a variety of online Self-paced courses, to help you getting started on both Tosca and qTest.

Thanks to Automation Specialist Level 1 and Level 2 you will learn how to automate your TestCases, deal with testing results and result versions. These courses are respectively 3 and 1.5 days long.

Test Design (Specialist) Level 1 and Level 2 will show you how to manage your test data and use Tosca's capabilities to its fullest. You will learn how to generate hundreds of TestCases with just a mouse click. These courses are respectively 2 and 1 days long.

Additionally, thanks to qTest Specialist Level 1 course you will learn how to deal with artifacts versioning in qTest. You will also learn how to set up a project in order for it to fit in your development methodology. This course is approximately 1 day long.

To get enrolled for online courses visit the Academy website here:

But courses are only the beginning, the Tricentis Academy offers a wide variety of user Tutorials and Micro content on both the Tricentis Website and our YouTube Channel. Here you will find in depth guides on how to best configure your Tricentis Analytics and qTest Insights. Additionally, we offer countless videos exploring best practices as well as configuration and administration of all products that make up the Tricentis offering.

Remember to subscribe to our YouTube Channel to be always up to date with our weekly video releases. And please - contact us, if you have any suggestions or requests for new videos!!

We hope you have enjoyed this short overview and are ready to dive into the Academy content. Good luck and happy testing!