

Capgemini



sogeti
Part of Capgemini



Continuous Testing Report

In association with

ca A Broadcom
technologies Company



Contents

Introduction	4
<hr/>	
Executive summary	6
<hr/>	
Current trends in continuous testing	
<hr/>	
Test design	9
Functional and performance testing	13
Test data management	17
Test environment management	21
Test orchestration in the agile enterprise	24
Continuous testing: the road ahead	27
About the study	31
<hr/>	
About the sponsors	37
<hr/>	

Introduction

Welcome dear readers.

Quality and testing approaches, methods, and expertise have undergone radical changes over the last few years. Every organization today aspires to deliver faster and more valuable IT solutions to business and customers. To do this, they have been leveraging agile and DevOps methodologies and using smarter automation technologies and as-a-Service solutions to deliver IT faster and with greater flexibility.

At the same time, the IT landscape has also been growing in complexity. There is an increased

dependency on IT solutions today, with the integration of front-office and consumer-facing apps with back-office core systems, the leveraging of cloud and microservices and the integration and use of IoT. And, on top of that, AI is emerging to make these solutions autonomous and self learning.

All this technology is delivered by different teams, many of which may not even be part of a single company.

As we scramble to deliver innovative solutions for the newer, more complex IT landscape, there is, of course, a risk of failure. While some failures are inevitable and often provide a valuable learning opportunity (given a quick feedback loop), there are others that we must prevent from happening. Failures in core systems that seriously disrupt the business operations of an enterprise, failures that seriously impact a large number of clients and therefore jeopardize an organization's reliability and brand perception, or failures in systems that cannot easily be rolled back all demand good testing of these systems before being deployed.

In reality, many organizations today are struggling to adapt their QA and testing processes to meet all these changes and needs. The core questions today are: how to achieve quality faster, how to do testing smarter, and what should actually be covered in a test?

The Continuous Testing Report 2019 gives you insight into what is state of the art today, a benchmark on quality approaches in Agile/DevOps, some use cases, and above all, some clear recommendations on what you can do to improve your QA and testing activities.

I wish you happy reading and many good insights that help you to achieve your continuous quality goals.



Mark Buenen

Global leader, Digital Assurance & Testing, Capgemini Group

Welcome to the Continuous Testing Report 2019 from Capgemini, Sogeti, and Broadcom Enterprise Software Division. This report is a comprehensive review of continuous testing challenges, methodologies, and adoption paths in global enterprises. Paired with commentary from subject matter experts with decades of real-world experience, this report is a signpost for the current state of cross-industry continuous testing, and a roadmap to further adoption and realization of continuous testing benefits.

If your organization is hitting roadblocks on the continuous testing journey – you are not alone! While the report indicates that organizations understand the benefits of continuous testing and are keen to adopt the practice, we also see organizations struggling with various aspects of adoption. For example, we see some degree of testing automation across testing activities in the software development lifecycle. However, there is remarkable consistency in the issues that companies face. From getting the right coverage of the end - user journey and requirements in the test set, to automating functional testing, to accessing test environments on demand – most companies are dealing with significant challenges in successfully transforming their current testing practices to a true continuous testing paradigm.

The Continuous Testing Report 2019 illustrates, as did many other surveys before it, that the rate of new software deployments is generally growing in line with organizational expectations. In the report, 58% of respondents say they are doing deployments daily or even more frequently. But testing still remains the biggest bottleneck to getting new, quality functionality into the hands of customers with the desired volume, velocity, and quality. For example, respondents are spending an enormous amount of time maintaining test environments, doing tasks that can be automated and provided “on demand” with today’s testing practices.

Progress is happening. We’re seeing notable success and ROI when enterprises implement a new continuous testing mindset and tools that increase quality, lower cycle times, and use resources more

efficiently. These include tools for test modeling, test data management, service virtualization, in-sprint test automation, test orchestration, and easy-to-adopt, SaaS-based tools. We’re shifting left with “as code” and developer-friendly tools and shifting right to use monitoring data from production to continuously improve test sets and optimize test cycles.

The future is bright. There are exciting things on the horizon for continuous testing, leveraging new technologies such as machine learning, predictive analytics, and more. We’re excited about where we are and where we’re going.

We hope you enjoy the report.

Sushil Kumar

*Head of Continuous Testing Business,
Broadcom Enterprise Software Division*



Executive summary



Antoine Aymer
*Group Offer Director,
Sogeti*



Ravikumar Sengodan
*Senior Director, Financial
Services, Capgemini*



Christine Bentsen
*Head of Product Marketing,
Continuous Testing Business,
Broadcom Enterprise
Software Division*

This Continuous Testing Report (CTR) 2019

brings together survey data and subject matter expert contributions to outline the challenges and potential approaches to transforming test practices in the age of Agile and DevOps. It is based on the considered opinions of several subject matter experts from Capgemini, Sogeti, and Broadcom, buttressed by the results of a global survey of 500 senior decision-makers in corporate IT functions, working for companies and public sector organizations across fourteen countries.

The report is targeted at QA and testing professionals as well as senior tech decision-makers. It is meant to help them answer the following questions:

1. What do their peers think about some of the operational issues related to continuous testing?
2. What's working, what needs to be fixed, and how to go about it.
3. How is continuous testing evolving over time and where is it headed

Some of the key findings captured in the report are:

– Agile, DevOps, and automation are the key ingredients of IT today

Organizations across various sectors are committed to getting their products to the market as quickly as possible. According to the CTR 2019 survey, 58% of respondents deploy daily, or even more frequently.

- **Continuous testing is required and is a priority, though most organizations are still far from achieving it**

While significant improvements have been made in individual areas of the software development lifecycle, testing is still the key bottleneck standing in the way of truly implementing DevOps and continuous delivery.

- **The adoption of continuous testing is increasing, but still has serious gaps**

One part of the move towards CT has been the adoption of test automation for various QA tasks along with the move towards Agile and DevOps. However, the test automation landscape in the enterprise is scattered, fragmented, and lacks orchestration. The CTR 2019 also reveals significant gaps in automation, which slows down the entire process. According to the survey, on average just 25% of test data is generated using test data tools, 24% of performance test cases are executed using test automation tools, and 24% of end-to-end business scenarios are executed with test automation tools.

- **There is increasing interest around in-sprint and model-based testing (MBT)**

Both in-sprint testing as well as model-based testing are crucial enablers of continuous testing. The idea behind in-sprint testing is simple – to complete all testing activities inside the development sprint. This means that you need to write and communicate the correct requirements and generate testing artifacts, all within the same development sprint. One of the critical components required to do this successfully is the ability to write complete, unambiguous, and accurate requirements. This has led to a rise in the popularity of model-based testing, which is based on the visual representation of requirements and automatic generation of test cases. Model-based testing also provides for the automatic generation of test artifacts (including test cases and test automation scripts), as well as the flexibility to adapt to changing requirements using automated requirements change processes. According to the survey, 30% of respondents said they foresaw using model-based testing in the coming year.

- **Requirements management, test data, and test environment provisioning are the key challenges for CT**

The survey results reveal that the biggest bottlenecks for CT are in the areas of test data, test environments, and writing requirements that provide adequate coverage. When asked about how challenging each topic was when it came to the

implementation of continuous testing, respondents gave high weightings to “getting the right test data on time” and “getting the right coverage of end-user expectations and requirements in the test set.” The CTR 2019 results also revealed that, on average, respondents were spending as much as 47% of their time in building, managing, maintaining, and decommissioning test environments. This is an inordinate amount of time and indicates that strategic improvements in these two areas can potentially bring about a significant reduction in the total time spent on QA activities.

All in all, it’s important to remember the theory of constraints, which states that “you can only go as fast as your slowest asset.” Today, while organizations have transformed to Agile and DevOps methodologies, testing remains the missing foundational ingredient for a complete transformation of the software development lifecycle. The Continuous Testing Report 2019 looks at some of the key trends in different areas of continuous testing, the key challenges or gaps, and some remedial measures that organizations can immediately take to improve.





Current trends in continuous testing

Test design

Deepika Mamnani

Senior Director, Financial Services, Capgemini

Marco Venzelaar

Managing Consultant, Sogeti UK

Alex Martins

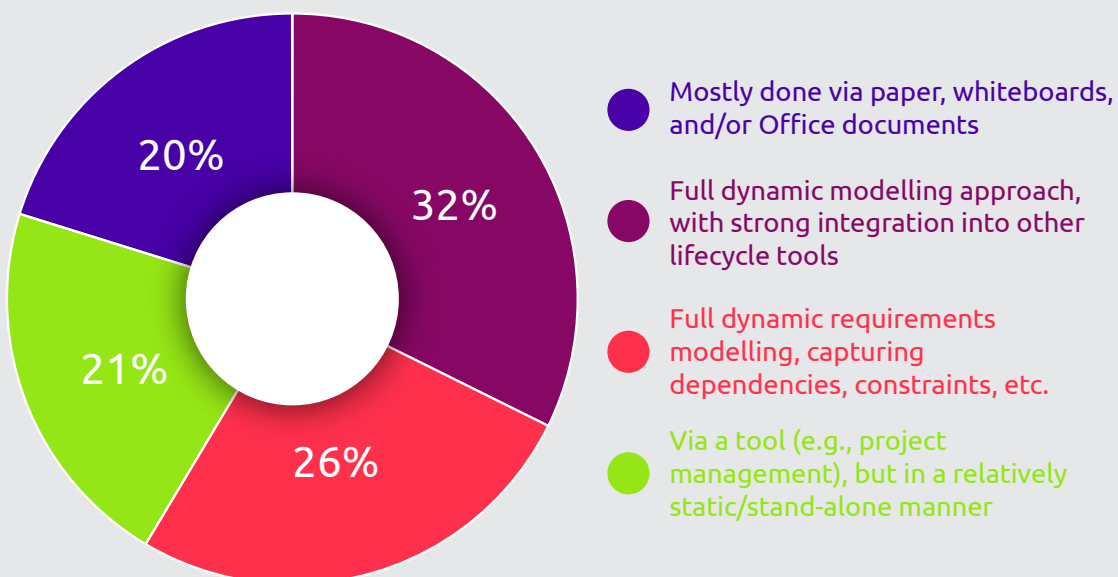
Head of Presales, Americas, Continuous Testing Business, Broadcom Enterprise Software Division

Model-based testing and AI solutions for better requirements management

Test design refers to the act of creating, writing, and updating test cases in response to changing requirements. It is a complex and intricate undertaking, and as illustrated by the Continuous Testing Report (CTR) 2019 survey results, there are several challenges that organizations are struggling with in this area. Some of the biggest challenges relate to poor test requirements capture, inadequate understanding of testable criteria, and lack of appropriate test coverage.



Current approach to gathering, analysis, and engineering



According to the Continuous Testing Report (CTR) 2019 survey, 40% of respondents said that their current approach to requirements gathering, analysis, and engineering was not automated. Some of these respondents reported using tools for rudimentary requirements capture in a “relatively static/stand-alone manner,” while others said they were using “paper, whiteboard, or word processing software.” Even the organizations that did practice dynamic requirements capture and modelling reported facing challenges in integrating requirements modelling with the existing lifecycle tools and processes.

Continuing from the survey, when asked about the degree to which testing is considered or incorporated into the requirements definition and change management phase, 62% respondents replied that although key application requirements were highlighted, full test case design and development remains a separate function. Some were very explicit in saying that testing has never been incorporated into a requirements definition process. For most

or stories through dynamic visual models, instead of text-based ones.

By automatically mapping requirements to a visual active flowchart model, teams have been able to accelerate requirements model creation and automate the test design effort with the appropriate test coverage. When tests match the requirements rather than the code, there is more collaboration and less rework, as there is cross-team understanding of the functionality required by the business. The acceptance criteria in this format is an excellent way to achieve in-sprint automation when business and/or product owners collaborate with developers and testers.

Designing test cases with appropriate requirements coverage

In an agile and DevOps environment where requirements change frequently, maintaining appropriate requirements coverage while testing at speed becomes crucial. This

How often clarification is required



organizations, neither requirements coverage plans, nor the automatic generation of test cases are the norm today.

This signals a big challenge for requirements management. Inordinate amounts of resources, time, and effort are being spent in clarifying and communicating requirements. According to the CTR 2019 survey, the vast majority of respondents said they spent between 40–70% of their time in clarifying requirements. To tackle such challenges, many project teams have moved to specifying requirements and/

is an area in which most organizations are struggling, as demonstrated by the survey results.

For instance, when asked about their challenges in ensuring test case coverage, 67% of respondents said “maintaining appropriate test coverage as requirements change is a big challenge,” followed by 61% who said “in-sprint requirements changes make test coverage maintenance particularly hard,” 60% who said “despite best efforts we still end up with gaps in our test case coverage,” and 56% who said, “we generally end up with far more test cases than

we need, with lots of overlap and redundancy.” All these factors are related. Often, when requirements change, it is difficult to identify the impact on the system, making it tougher to ensure sufficient test coverage. This can often lead to situations, where only a fraction of the feature functionality is covered at any given instance, while in others, a feature function may be redundantly tested several times over.

Leverage smart automation technologies for test design

The third challenge arises from the expanded IT landscape in today’s digital and integrated application economy. Today, a single application could be connected to thousands of other systems enterprise-wide, making it difficult to understand the full scope of testing required. Compounded by the pervasiveness of light-weight architectural choices, such as microservices and the complexity of developing for internet of things (IoT) connected devices, this has only accelerated the need for a test automation framework that does away with manual test creation and time spent on maintaining a growing library of test assets by hand.

To deal with this, testing professionals need a test automation platform that starts with the ability to

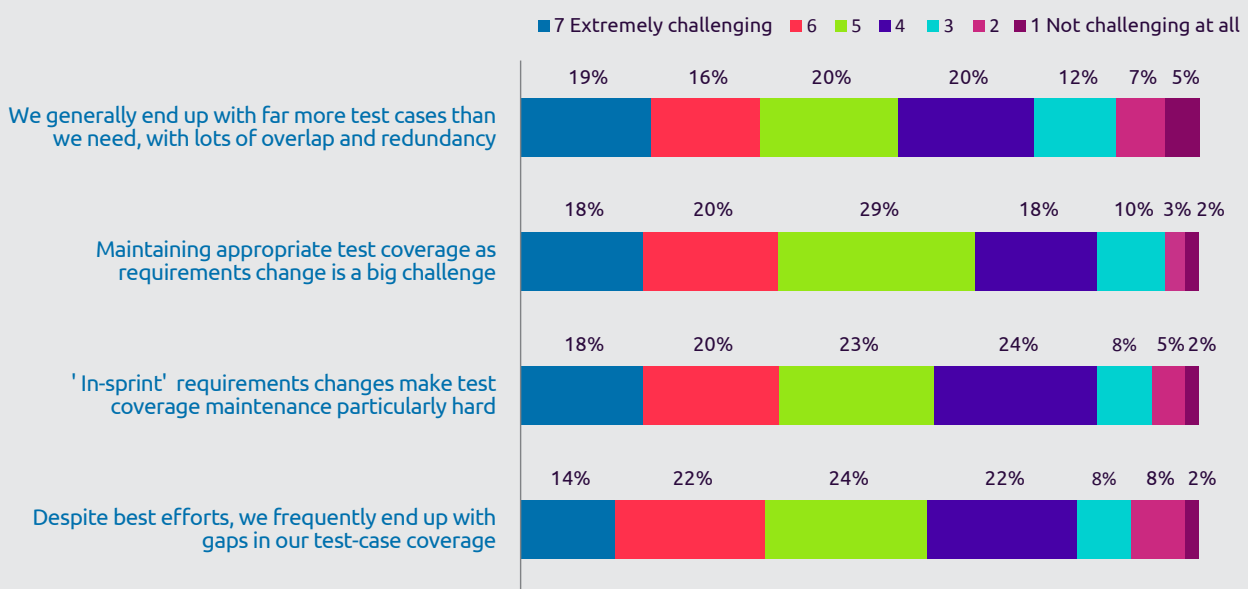
automatically generate maximum test coverage using the smallest number of test cases. The agile test automation platform ensures that the test cases are optimized and linked to the right data and expected results for the complete architecture of a system. These should be stored in a central library, to be shared and reused across distributed teams to fully test systems as they evolve. The test sets need to be open, agnostic, and easily imported-exported across third-party lifecycle tools and when requirements change, test sets need to be automatically maintained. This process also needs to include the finding and repairing of broken tests and the automatic creation of new tests to maintain appropriate test coverage.

Recommendations

To prepare for what is to come, we recommend the following:

- Embark on the test case design optimization journey by adopting model-based testing (MBT) techniques: In model-based testing, models are used to represent the desired behavior of a system under test (SUT). The model helps teams collaborate and get a full understanding of the expected behavior of a system, and the model can be adjusted as requirements change.

How challenging each topic is when ensuring test case coverage



- For agile frameworks, it is recommended to adopt user-story thinning techniques. This refers to techniques used to specify user stories through dynamic visual models that accurately and quickly convey technical testable requirements and define interdependencies in systems. This can help in designing light-weight, testable user stories. Once you have gained proficiency in designing testable user stories, adopt behavior-driven development (BDD) design techniques. This refers to documenting user stories in a common language that can be understood by all teams and specifying the acceptance criteria that can then be effectively automated.
- Once you have gained efficiencies from the above techniques, leverage predictive analytics for test case design. This refers to deriving analytics from production, orchestration, lifecycle, and agile planning tools to consistently improve testing efficiency and application quality.

Use case

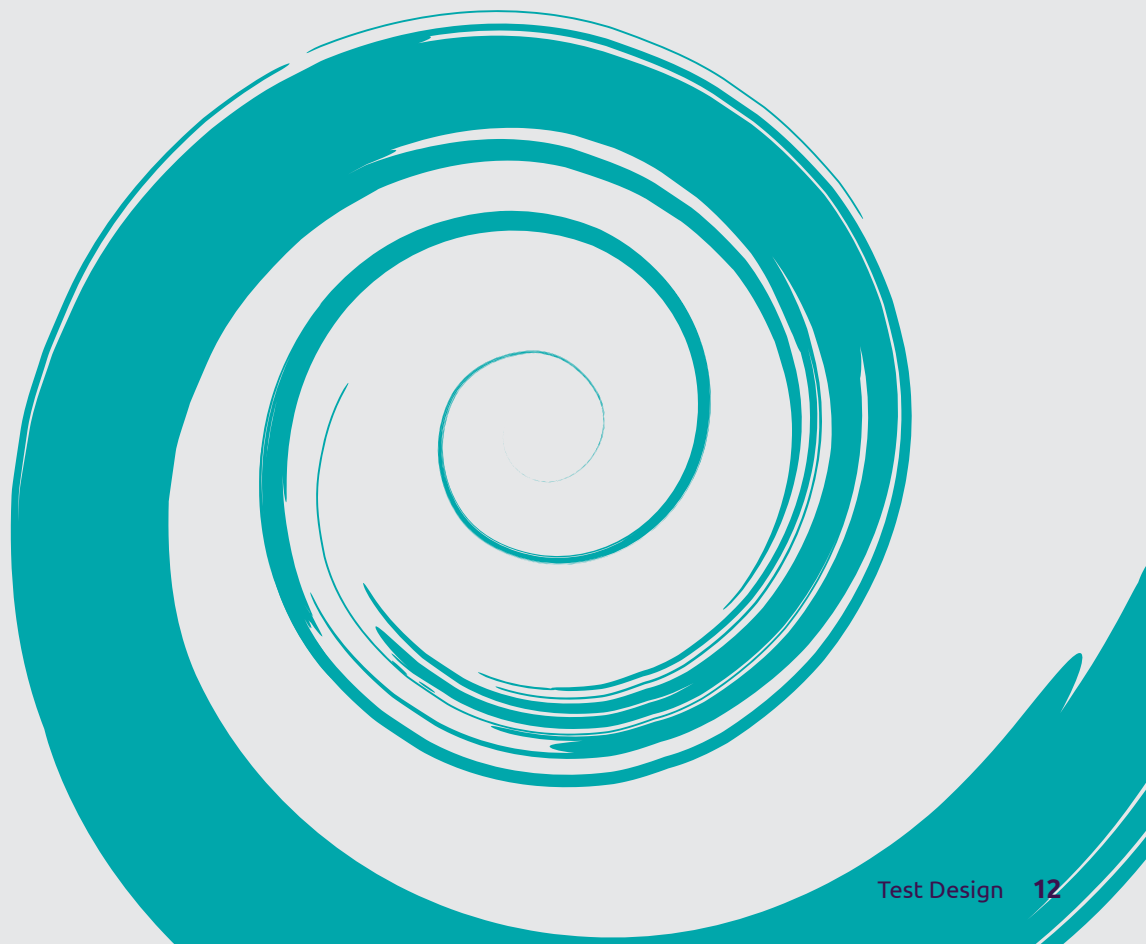
Recently, a global financial services leader went from doing manual requirements definition and test case design to a model-based testing and test design automation approach using a tool available on the market. They started building requirements models to represent the expected behavior of the application. This was followed by automatic test case generation and script creation.

The result? The firm dramatically improved their testing efficiency, creating new test cases 10x faster by automating the creation process in hours as opposed to days or weeks prior to test design automation adoption. Further, they went from 90+ days behind in new automation to full, in-sprint automation of new features (not just regression) with little to no technical debt.

Looking to the future

Organizations today are struggling with several challenges related to test design for continuous testing. Successfully negotiating these challenges requires serious integration of QA in the early requirements stage. We are already seeing business analyst teams getting integrated with development and QA so as to enable organizations to determine continuous requirements and then integrate them as part of the continuous delivery pipeline. This trend is only likely to strengthen in the future.

Over the next two to three years, we are likely to see a greater uptake of test design techniques such as model-based testing and behavior-driven development, the greater use of AI and ML elements to get the appropriate coverage, a reduction in testing time, as well as a greater ability to automatically generate tests in response to changed requirements within a single sprint.



Functional and performance testing

Sheetal Ghag
Consultant, Sogeti Sweden

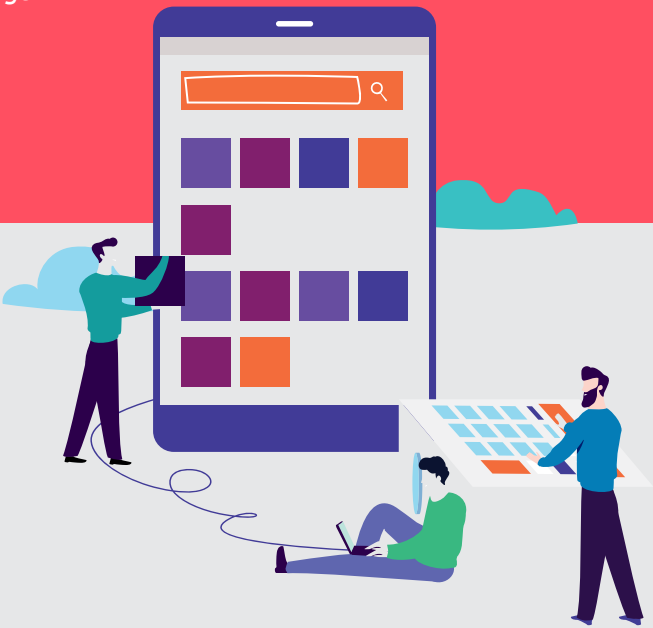
Jitender Chandani
Director, Financial Services, Capgemini

Kiran Kumar Makarla
Technical Marketing Manager, Continuous Testing Business,
Broadcom Enterprise Software Division

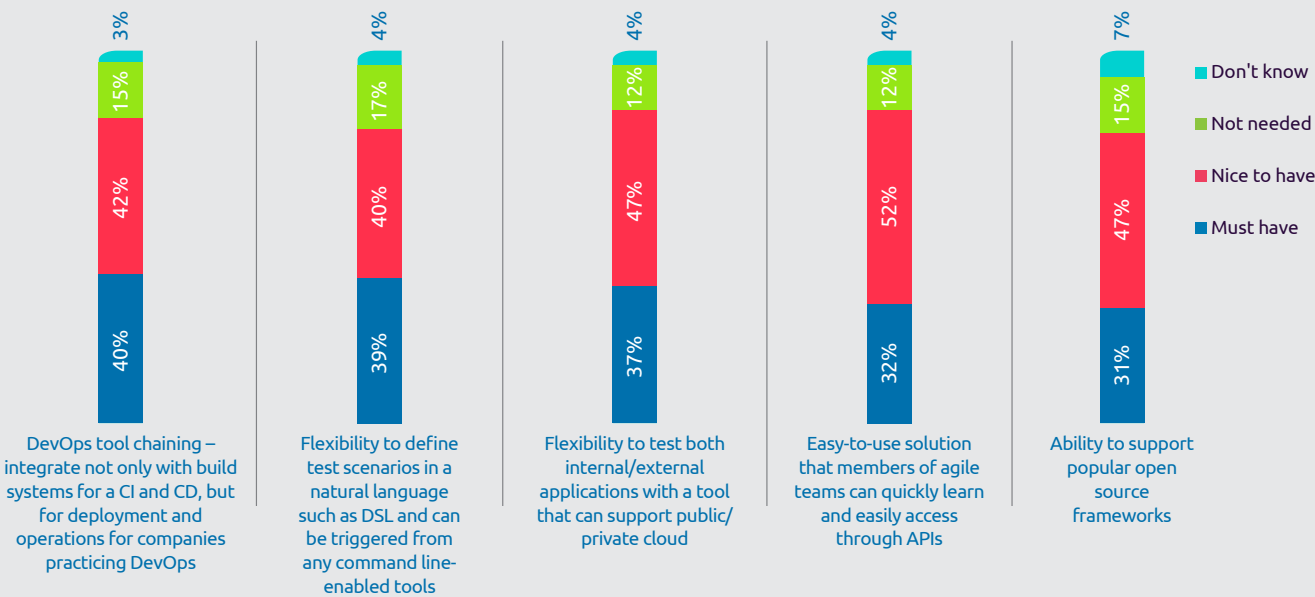
A change in mentality, culture, and organization is required

Overview

This chapter describes the impact of functional and performance testing on teams practicing DevOps and Agile and the way companies are coping with these changes. How can developers enjoy autonomy while working closely with business teams? How difficult is it for testers to step out of traditional waterfall models and be a part of fast-paced agile teams? How are open-source tools changing the face of



Critical components Agile teams look for in modernizing load and performance testing



testing as we know it? These are a few questions this chapter will address.

The transition to autonomous development teams

The rise of the application economy has brought developers into the thick of the action. They need to collaborate closely with business teams to improve the user experience (UX). They are also trying to replicate real-life customer experiences as closely as possible, using design-thinking principles. For many developers, this can be an incredibly exciting experience as well as an opportunity to contribute to the business. It is also a positive for DevOps team structure as the silos between the business and development teams are dissolved.

However, to work freely, developers need a robust infrastructure to support them. This includes the freedom to

asked about the critical components that agile teams look for in modernizing load and performance testing practices, 31% of respondents said the ability to support popular open source frameworks, was a “must-have,” while 48% said it was “nice to have.” This allows developers to be a part of a community and access required tools simply by downloading – something that is in sharp contrast to the proprietary tools where developers must purchase from vendors.

Today, developers and companies are also increasingly turning to cloud-based Platform-as-a-Service (PaaS) solutions for setting up test environments. This is because PaaS platforms can handle huge virtual user loads at comparatively low cost. The virtual user loads can also be scaled up and down almost instantaneously in response to requirements. This feature becomes especially important in the case of A/B testing, in which a developer might try to simulate a web application for 100 users to start

Frequency of undertaking activities



choose their own set of tools, provision an environment, and even scale up or down as per the requirements.

with and then scale it up to thousands of users across various geographies.

As far as tools are concerned, the popularity of open-source tools has been rising over the last few years. According to the Continuous Testing Report (CTR) 2019 survey, when

The changing role of the tester

The fast pace of functional and performance testing has hugely impacted testing teams. A decade ago, testers were part of large teams with fixed schedules and predictable loads. But with agile teams, the old structure is no longer relevant. In most organizations, testers are now working in parallel with developers and business teams and testing early in the project as part of the “shift-left” movement.

“Shift left” involves the testers much earlier in the software development life-cycle, thus allowing them to understand the requirements, software design, architecture, coding, and functionality. This enables them to ask questions, seek clarifications, and provide feedback wherever possible to support the team. The mixed nature of teams and the expanded scope of responsibilities has also led to a greater demand for Software Development Engineer in Test (SDET) professionals who can participate both in development and tests.

The waterfall approach of “big-bang” testing is not enough to handle the rapid iterations that functional and performance testing demands. Experts say that testers should now focus on breaking down tests into small components and functions with the purpose of testing only the changes made in an application. This will help in reducing the feedback time to minutes from hours or days.

For successful functional and performance testing, testers need to focus on the big picture. For instance, if a website is slow to load because of a high definition images or multimedia, testers should evaluate network bandwidth, latency, and the platform on which it is saved. This is a more focused way of testing that will help companies pinpoint the source of the problem and move towards identifying bottlenecks effectively.

According to the CTR survey, 32% of respondents stated that little or no performance and load testing is done in their company. Experts think that the reason for this is insufficient budget allocation to performance testing in the early stages of the project. Many companies suddenly wake up to the reality of performance testing just before the release when they fear something might be wrong with an application.

Experts say that forecasting the workloads, network simulation, and measuring the response times using the appropriate software tools are vital for performance testing of websites or mobile apps. This should also be coupled with synthetic data from production environment with the help of the same tools.

Adapting the test organization

Today’s test COEs (Test Centers of Excellence) face a dilemma as functional and performance testing have become integral to continuous testing and yet they haven’t kept up with agile and continuous software delivery trends. Instead of facilitating speed, testing has become a real bottleneck – forcing a tough trade-off between speed and quality.

Rather than having large groups of testers who are centrally organized, there is a need for people with quality focus embedded within agile teams. Their new roles require them to understand the business context and come up with test strategies that minimize the risk in applications under development. The following recommendations outline some of the key steps to make this happen.

Recommendations

We recommend the following actions to create teams that can provide adequate quality coverage, without compromising on time to market:

- Empower individual teams to manage and scale their quality resources with best practices and know-how, so they can better respond to the new business challenges.
- Build quality into the development process by ensuring developers adopt quality practices, such as keeping adequate requirements documentation, and perform automated unit testing.
- Ensure test automation is leveraged as the key tool for test coverage rather than being limited to just regression testing. Instead, automation needs to be a part of every code commit in order to ensure that quality is baked into every step of the software development lifecycle.
- Foster an environment in which teams can leverage their core strengths, such as using the programming language of their choice for automating the test scripts and keeping the right talent engaged.
- Standardize activities such as provisioning of test environments, generating, gathering, or masking/scrubbing test data, creation of workspaces, handing of API keys, reviewing the dashboards, and sharing of test metrics with the entire team.

Use case

The client was a daily deal recommendation service for consumers. Every 24 hours, they broadcast electronic coupons for restaurant and stores across different cities, recommending local services while also offering deep discounts on purchases of those services.

The main challenge was that the organization used multiple open source testing tools for CI testing across various teams that were located globally.

- They needed to streamline their internal processes and methodologies around testing by standardizing on one tool set.
- They chose and implemented an open source-based performance testing tool that allowed them to execute all their different open source testing scripts because it was 100% compatible with popular testing tools that teams in different regions were working with.
- The tool was both enterprise-scale and offered support, upgrades, and the reliability the organization required.
- The tool provided a way to execute the various open source scripts currently in use to be used on one platform.

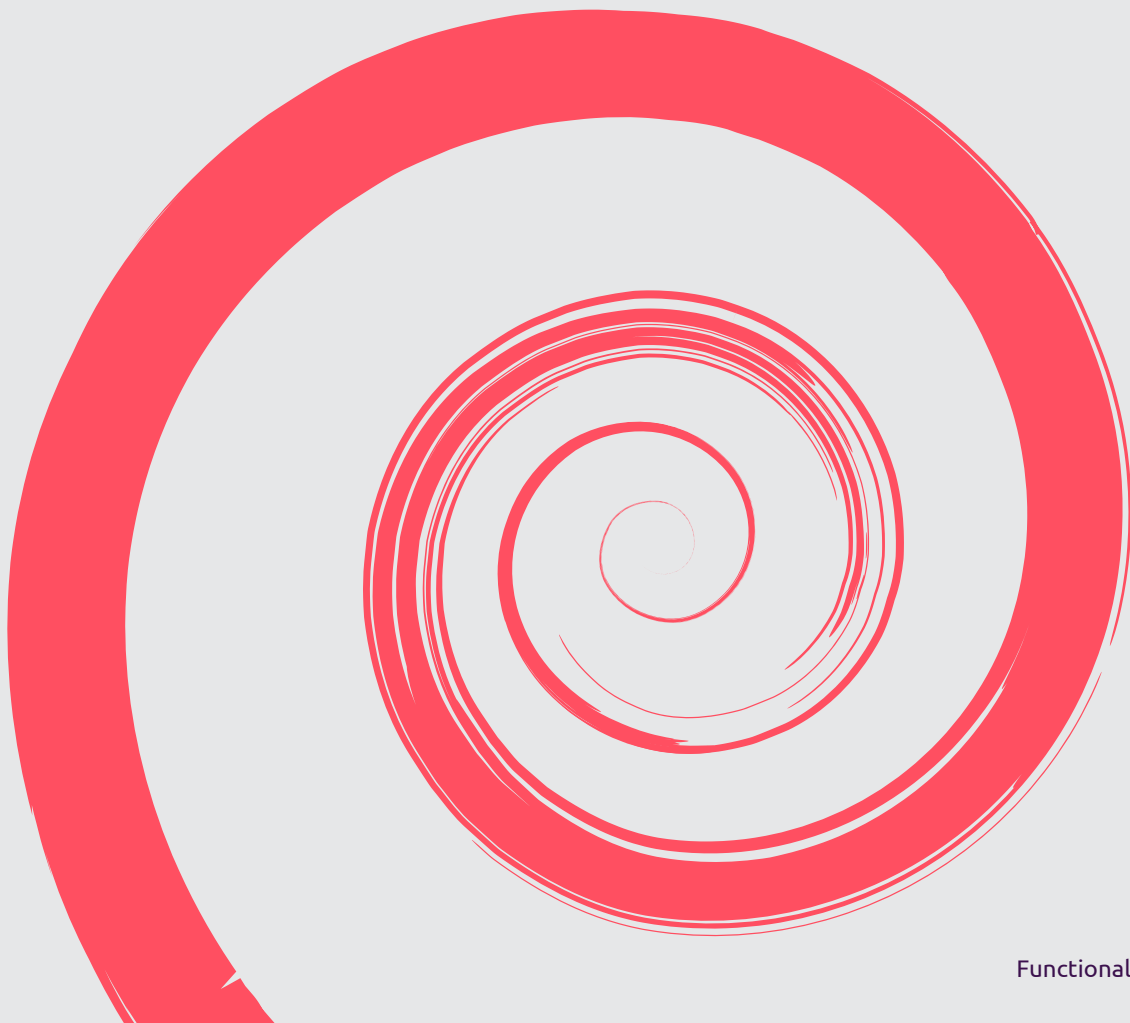
Results

- 80% reduction in maintenance of test infrastructure by streamlining and standardizing on a single platform
- 90% productivity gains in reusing various open source scripts from various teams and executing from a single platform.

Conclusion

Enterprise testing teams are in constant struggle not only to keep pace with the evolution of new technology stacks for the modern applications being built, but also with the new testing requirements that they demand. In the past, massive-scale load tests were only possible with expensive testing environments. Today however, SaaS-based solutions provide simple, self-service capabilities to generate test traffic from cloud-based locations around the world.

Projects teams are advised to run larger number of small-scoped tests in parallel. This will enable them to ensure that applications perform as expected before promoting code to the next stage in software delivery pipeline. Tools that have intuitive dashboards and rich, interactive timeline graphs can help teams identify bottlenecks quickly, providing the best returns on investment (ROIs).



Test data management

Jeffrey Hughes

Senior Product Marketer, TDM, Continuous Testing Business, Broadcom Enterprise Software Division

Prashant Pandey

TDM Product Manager, Broadcom Enterprise Software Division

Ashish Behanan

Director, Financial Services, Capgemini

Peter Haberl

Member of Management Board, Practice Leader Digital Assurance & Testing, Sogeti Deutschland

Josep Robert

TDM Service Manager, Sogeti

Regulations and the complexity of solution architectures make test data a priority

Overview

Test data management is in focus today, due to regulations such as the General Data Protection Regulation (GDPR), increasing concerns around privacy and data security, and the increasing availability of various off-the-shelf test data management (TDM) tools. Locating proper test data is also one of the main bottlenecks to achieving continuous testing in DevOps. According to the Continuous Testing Report (CTR) 2019 survey, 55% of respondents are currently spending between 30 to 60 percent of their total testing time on test data management activities. This is an inordinate amount of time and there are indications that most organizations have realized that addressing this one

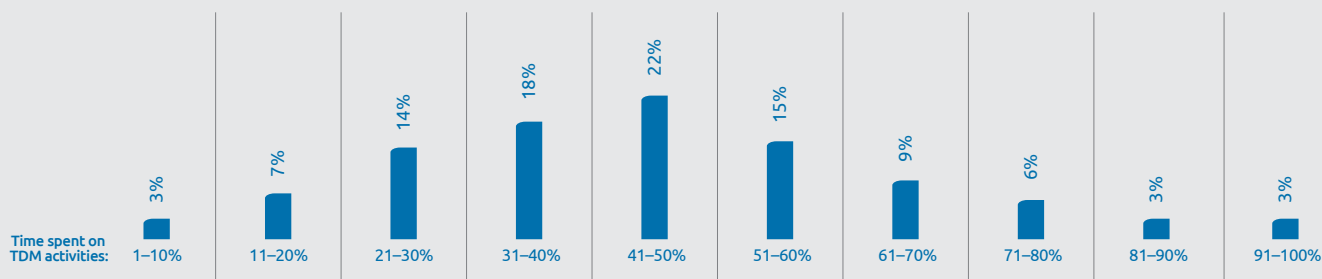
area will dramatically improve the speed and efficiency of the entire software development lifecycle (SDLC).

At Broadcom and Capgemini, we have seen a clear increase in the number of organizations performing assessments of their test data management practices and tooling over the last couple of years. Such assessments are always the first



Time spent searching, managing, and generating test data

■ Percentage of respondents



step in the transformation of the test data management function – and such transformation is needed, as demonstrated by the CTR 2019 survey results.

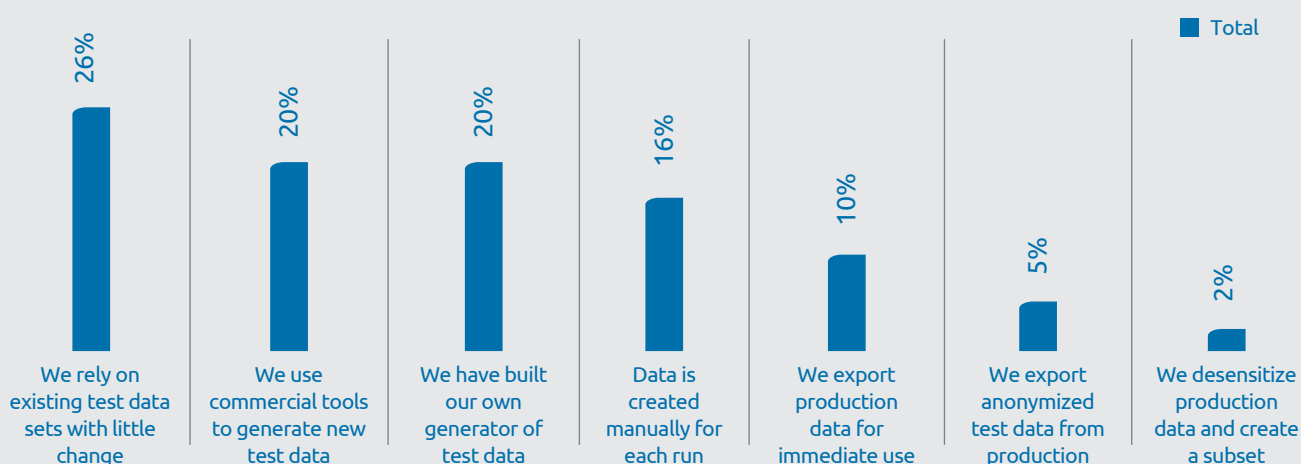
For instance, when asked about how they generated test data for continuous testing, 26% of the respondents said they use existing test data without any changes, making this the most popular method of provisioning data. The problem with this approach is that it often leads to inadequate test coverage, testing inefficiency, and mounting compliance and security risks. The reuse of test data sets also leads to issues such as the aging of timestamps and date data fields. Another approach to generating test data is to directly copy

limitations, ranging from continued development needs to maintenance over time. Many of these organizations, which currently depend upon such in-house tools, will usually need to move to commercial tools for test data generation and other automation capabilities.

The three key test data challenges: data knowledge, data access, and data management

The greatest challenge as indicated by 33% of respondents was the difficulty and time involved in extracting data spread across multiple databases. This is a reflection of the lack of

Provisioning and generation of test data



data from production environments. According to the survey, 10% of the respondents said they followed this practice. Of course, this reliance on production data has been decreasing over time, with GDPR acting as an important catalyst for the switch to masking, sub setting, and synthesizing of data.

These trends have led to the increasing automation of test data generation and the adoption of test data management tools. Twenty percent of the CTR 2019 survey respondents said that they used commercial tools to generate test data and another 20% said they had built their own generator of test data. A key factor holding back the greater adoption of professional, automated TDM solutions is cost. Organizations also struggle to find a single tool that will support all their applications and provide end-to-end functionality. Unwilling to invest such amounts (anywhere between EUR100,000–500,000 for an end-to-end automated test data solution), many organizations have built homegrown tools, which they rely on for their test data needs. However, homegrown solutions have a number of

knowledge and documented information on where data resides in the system. The reality is that every organization, whether large or small, relies on multiple, interconnected systems such that the data required for almost every test is spread across multiple databases. This makes extraction of the required test data a time-consuming process, as organizations typically do not have their data model well documented. The problem becomes even more acute for larger organizations with more complex infrastructures.

To deal with this challenge, we recommend the creation of an accurate metadata model, containing up-to-date information on various databases spread across different systems. However, for such a solution to be practical, the metadata model must be used in conjunction with a tool that is synchronized with all of the organization's applications, so that the model is automatically updated in real time as and when there are changes to any of the databases.

The second-highest challenge cited by 28% of respondents is that testing teams have limited access to production systems and are dependent on database administrators to get the data they need. This is, of course, as it should be! Production systems are business critical and should not be widely accessible. A related issue is having to maintain multiple test data set versions for different tests; the third-most important challenge in the survey, with 27% of respondents reporting it as a challenge.

One recommendation to deal with all three of the previous challenges is to create a centralized team to manage test data provisioning for all project teams. In fact, several organizations we know of have already started creating teams that look at test data from the perspective of tooling and from a process and operations stand-point. Such teams have a view of the production data and can keep track of the test data required by various teams, bringing in multiple efficiencies by centralizing data expertise and standardizing tools and processes. Organizations can also work towards creating self-service portals (backed up by a small test data team) to help test teams generate data whenever required. Such initiatives are expected to drive both innovation as well as greater automation of test data management. The use of such tools can also help with the third challenge, that of maintaining multiple test data set versions for different tests.

In addition to the previously mentioned challenges, 26% of respondents cited “creating and maintaining test data which are not copies of production data” as a significant concern.

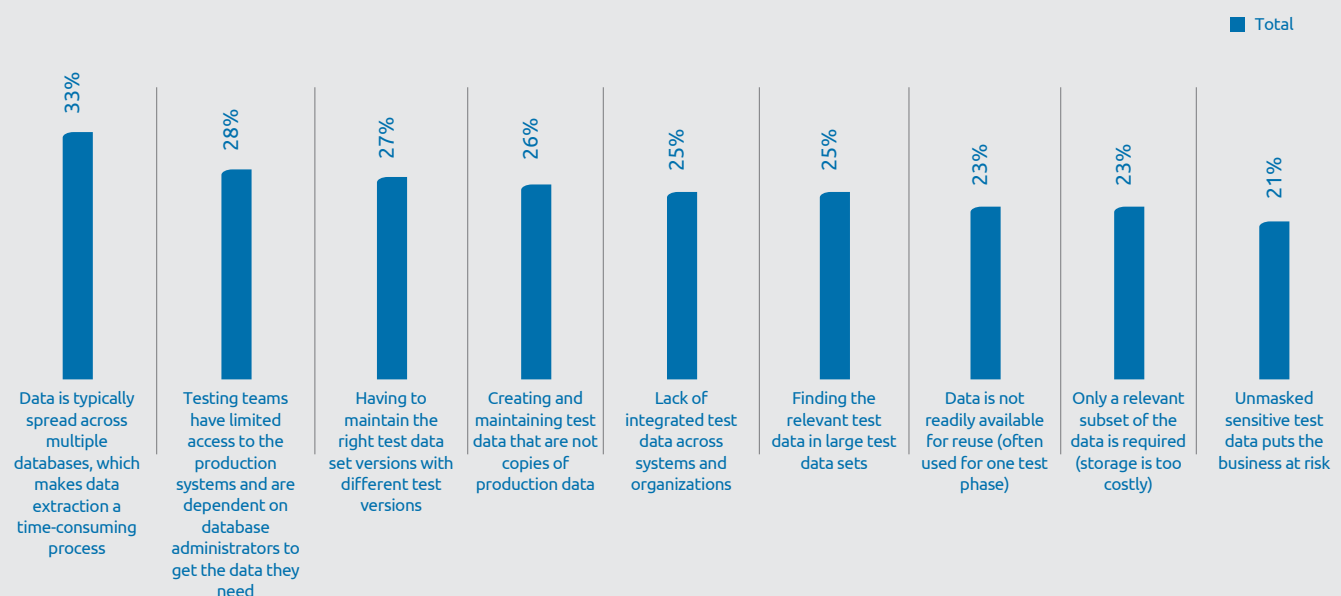
This is a considerable challenge as organizations need to keep updating their test data sets (which might consist of masked data as well as synthesized data) every time there is a change in any of the associated databases. As mentioned, the adoption of tools synchronized to various applications and databases, coupled with the right processes, can help organizations capture these database changes in real time and automatically update the relevant test data sets.

Key recommendations

To deal with the challenges identified in this survey, there are a few practical steps that organizations can take to achieve better test data management.

- First, identify your biggest test data management challenges. Are you currently using a homegrown test data management (TDM) system? What are the limitations of this tool? Are you currently employing some form of masking and subsetting? Are you concerned about how you are using production data for testing, as it pertains to data privacy and compliance? Are you looking to expand how you generate test data to utilize synthetic data creation?
- Next, review available test data management (TDM) solutions. Define your test data priorities and look for tools that will meet your needs now and, in the future. If you primarily need a solution for masking and subsetting, but you also want to move towards synthetic data generation, look for a solution that provides all of three of these data generation capabilities. If the primary

Challenges with test data



driver is mounting regulations for data privacy, look for a TDM solution that can actively address GDPR and other data privacy regulations by helping you locate, mask, or generate good test data. Or, if your company is struggling to find the right test data for multiple testing teams and keep the test data organized and “clean,” look for a TDM solution that provides a self-service portal where testers can find and reserve or generate synthetic test data whenever they need it.

- Then, adopt the appropriate commercial TDM tools and build your test data support team. Start small: pick a single team or process and implement the selected solution. Document the learnings and build up expertise: a certain level of expertise needs to be achieved before rolling out the solution across the entire organization. Create a change management team experienced with the solution to help roll it out across the entire organization. And be sure to create a metadata model coupled with an automated solution for real-time updates.

Use case

A large energy provider located in the EU had concerns about the General Data Protection Regulation and how non-compliance could adversely impact their revenues. Because they dealt with thousands of consumers and maintained large amounts of customer data for testing purposes, they decided it was time to find a solid test data management solution. They specifically searched for a solution that would allow them to discover test data across their existing databases, mask data, and generate new data from scratch for some applications. They have adopted a hybrid approach to test data management where they still use production data that is masked (for legacy application testing) and

use synthetic data for testing their new mobile and web applications.

The result has been faster generation of test data that is GDPR compliant and also of higher quality. Testing teams now have the test data when they need it and in the right format required for each application test. This saves the company time and cost through a streamlined test data management process that will accommodate their application development needs for years to come.

Conclusion

Having the right test data is undoubtedly a key consideration for the realization of significant time and efficiency gains in quality assurance (QA) and testing. Such efficiency considerations, along with regulations such as the GDPR and privacy and security concerns are driving important changes in test data management. Over the last few years, there has been an increase in the use of commercial TDM tools that provide data subsetting and synthetically generated data. There has also been a steady decrease in the use of production data for testing. These trends are only likely to gain in strength as we move forward.

Looking to the future, organizations will need to have a well-planned test data strategy, if they are to fully achieve continuous testing. Organizations will need to have their test data model documented and linked to a test data management tool. We are also likely to see an increase in test driven development (TDD) and a requirement for new skillsets and roles that merge the skills and mindsets of developers and testers.

Test environment management

Anish Behanan

*Director, Financial Services,
Capgemini*

Martin Salgado

*Advisor, Presales, Continuous
Testing Business, Broadcom
Enterprise Software Division*

Johan Torneheim

*Senior Consultant, Test Environment,
Sogeti Sweden*

Tools, service virtualization, and centralized teams for improved test environment management

Overview

The adoption of Agile and DevOps has pushed most work down to individual project teams and this has led to development, testing, and integration activities being carried out in parallel either within the same or across different teams. These parallel development streams necessitate a greater number of environments at different times. Organizations today are struggling to meet this need because they rarely have complete end-to-end environments. This lack of appropriate test environments when required, along with the challenges related to finding test data, form the biggest bottleneck to the implementation and evolution of continuous testing. These challenges also drive some of the major trends in test environments today.

For instance, a key trend that has grown over the last few years is the adoption of the cloud. Cloud-based testing environments remove the need for physical infrastructure and back-end systems to support those environments. Service virtualization further allows quick environment provisioning by simulating unavailable systems and services so that development and testing can continue in parallel. In addition, organizations are increasingly moving towards containerized environments. Instead of building complete virtual environments, project teams are spinning up containers and managing them, thus speeding up environment provisioning and deployment, saving costs and

resources, and improving the manageability of the entire process.

Another big trend is the move away from manual processes to test management tools. There is a multiplicity of such tools in the market today; all of which lend further support to cloud adoption, test environment virtualization, and containerization. Together, these trends enable organizations to fully reap the benefits of Agile and DevOps.

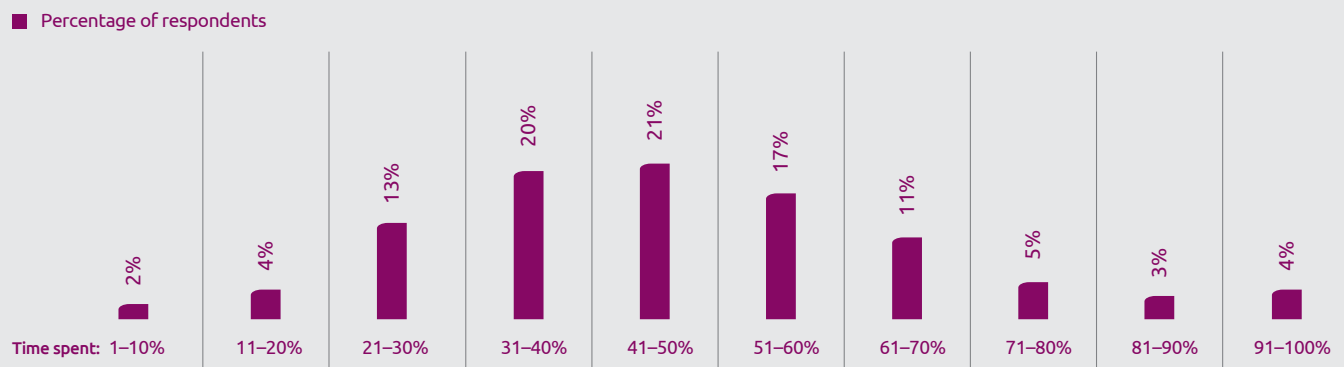
Challenges

There are still some roadblocks that stand in the way of greater adoption of Agile and DevOps. One of the biggest is the lack of centralized procedures and ways of working. With different project teams defining their own standards and deciding on tools and infrastructure, cascading best practices across the organization has proven to be difficult. Concerns around data privacy and security have also led to a cautious approach to the adoption of the cloud. Many organizations, especially in sectors dealing with sensitive personal data, are choosing private cloud solutions because of these concerns.

These issues with test environments were also confirmed by the Continuous Testing Report (CTR) 2019 survey results. For instance, when asked about the percentage



Time spent on test environments



of time allocated to building, managing, maintaining, and decommissioning test environments, as many as 40% of the respondents said they spent more than 50% of their time doing so.

Though the amount of time spent on test environments varies depending upon the maturity of the organization, CTR 2019 survey results indicate that, on average, the amount of time being spent on test environment activities is excessive and wasteful. To reduce this time, organizations are turning towards automation with the help of tools available on the market today. However, there is a learning curve with these tools and many organizations are still struggling to synchronize processes across different project teams. This is why experts are now suggesting the creation of centralized teams to select and adopt tools. In addition, the lack of centralized planning in terms of provisioning test data and environments is another factor that leads to significant delays in terms of scheduling, booking, and provisioning

environments, synchronizing processes, and cascading best practices across project teams.

Time came up again as a key issue when respondents were asked about – “test environment-related challenges that impeded efforts to improve the software development lifecycle (SDLC).” Participants gave the highest weighting to “wait times and cost for environment provisioning” (36% of respondents) and “complexity of needed applications” (36%), followed by “inability to identify defects early in the testing process” (33%). As already noted, both the time as well as costs of test environments can be reduced with the help of a centralized test environment team, use of automation tools, and containerization and virtualization of environments. The “complexity of needed applications,” arises due to the increasingly interconnected and interdependent application architecture that needs to be properly understood in order to build appropriate test environments. The statement could also relate to the complexity of the applications required

Test environment issues that impeded software development



to automate test environment provisioning and the lack of familiarity with such tools. Similarly, the “inability to identify defects early in the testing process” can be seen as a culture, mindset, and organizational issue. In particular, a greater focus on the applications integration phase would also help anticipate and reduce many unforeseen issues that come up late in the testing process.

Respondents were also asked about the test environment-related constraints that made the biggest impact in terms of restricting their testing. They gave the highest weighting to “scheduling constraints” (28%), followed by “restricted access to dependent services, components, or applications” (16%). Scheduling constraints are a challenge with the overall test environment management setup and workflows and as already mentioned can be tackled by creating a centralized test environment management team. The second challenge (restricted access to dependent services, components, or applications) arises due to the way testing is set up or organized. Project teams often need to access downstream applications or services through other teams that have access to that particular application, service, or component. This often becomes a tricky exercise, particularly when their requests are technical in nature, as they need to coordinate, coach, and hand-hold the gatekeepers to get what they want. Service virtualization can help in such cases, as it gives teams the ability to simulate downstream dependencies that are typically out of scope or managed by other teams.

Recommendations

The key recommendations for agile enterprises to overcome and remove the biggest test environment challenges for their agile teams are:

- Establish a dedicated test environment support team that provides solutions for provisioning, use, and maintenance of test environments to the agile teams.
- Strategize the transformation of test environments to cloud-based provisioning.
- Break down large complex test environments in virtualized components.
- Implement self-provisioning solutions based on containerization.

Use case

A financial company transitioning to Agile discovered that the move to parallel development and testing created the need to provision an increased number of test environments with the appropriate test data. They did not have the proper tools or resources to meet this challenge, so they explored the option of service virtualization and test data management solutions. Once they implemented these measures along with best practices and self-service solutions, they were able to reap the benefits of agile development and eliminate these challenges all together.

Conclusion

Test environments are one of the biggest bottlenecks to achieving continuous testing. The CTR 2019 survey results reveal the inordinate amount of time that organizations spend on test environment management as well as some of the key challenges in this area. We have also seen a few positive developments in terms of the adoption of virtualization, containerization, and tool-based automation. These trends are likely to strengthen in the future as organizations realize that virtualization and containerization are absolutely necessary to meet the demands of Agile and DevOps on a limited budget. The next two to three years are also likely to see organizations opting for increased levels of automation, particularly for solutions that automatically tell them about the impact that changes in functional requirements will have on test cases.

In the future, we are also likely to see organizations adopt a more integrated approach to testing with the help of tools or solutions that help unify the handling of test data, test environments, and test planning. In addition, as the shift left’ mentality gains greater ground in organizations, developers are increasingly going to be held accountable for testing activities. To pull this off, a large portion of the workforce might need to be re-trained with new skills, a culture shift, and the right systems. With the right tools and with the help of service virtualization, they might be able to spin up their own environments and test their code before moving it to the next phase. This would then be a truly agile way of working.

Test orchestration in the agile enterprise

Marco Venzelaar

Managing Consultant, Sogeti UK

Prabakaran Karuppiah

*Vice President, Financial Services,
Capgemini*

Uri Scheiner

*Senior Product Line Manager,
Continuous Delivery, Broadcom
Enterprise Software Division*

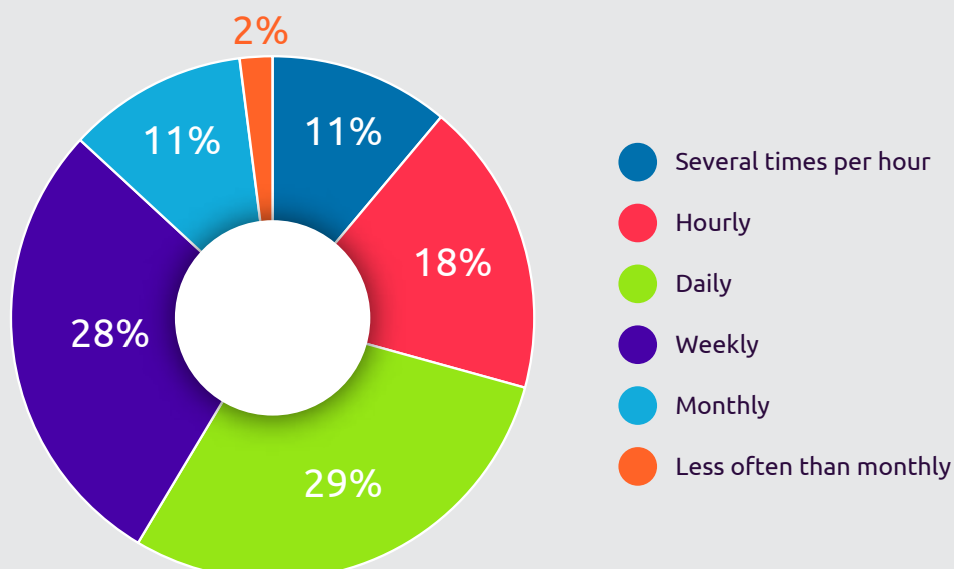


According to the Continuous Testing Report (CTR) 2019 survey, 32% of respondents said that they wanted to deploy new builds daily and another 29% said that they would like to do so on a weekly basis. Moreover, as many as 13% respondents reported that they wanted to deploy new builds several times per hour. Today, the biggest obstacle to achieving this kind of speed, lies in the quality assurance (QA) and testing function.

It is exactly to meet such demands for increased speed and quality that organizations have been turning towards automation tools over the last few years. This has led to a situation in which organizations have “islands of automation” in their software development life-cycle (SDLC) that are chained together with manual processes. Since the entire

system can move only as fast as it's slowest component, the whole process is unable to scale up to the demands of efficiency and speed being put upon it. This situation is further complicated by increasingly complex architectures and the lack of visibility for all teams into the development pipeline. The challenges arising from such complexity came through clearly in the CTR 2019 survey, as 62% of the respondents reported that “the releases are getting very complex, often involving multiple applications with

The frequency at which teams would like to deploy new builds



dependencies and different technologies with potentially conflicting resources.”

Test orchestration – a structured approach to continuous testing

The next step is to address silos of automation with orchestration. In the realm of testing, test orchestration eliminates “islands of automation” by combining manual and automated tasks in a holistic fashion. It links together individual, automated tasks, which helps organizations move from spending time on manual handoffs, dependencies, wait times, and cycle waste to one in which test generation and execution are fully integrated as part of a fully automated and optimized continuous delivery (CD) pipeline.

Testing today is more a sum of moving parts that need to be orchestrated perfectly together to achieve “quality at speed.” Organizations also want to orchestrate tests as part of a release so test results and quality data can serve as feedback to the quality and risk of the release, as well as the efficiency of the SDLC as a whole.

So how difficult is it to achieve this? According to the CTR survey, when asked to weigh the difficulties involved in achieving continuous testing (CT), respondents gave the highest weighting to “coordinating and sharing knowledge across the CT toolchain,” followed by “standardizing and automating the release processes,” and “identifying the bottlenecks by gathering data across the entire release pipeline.”

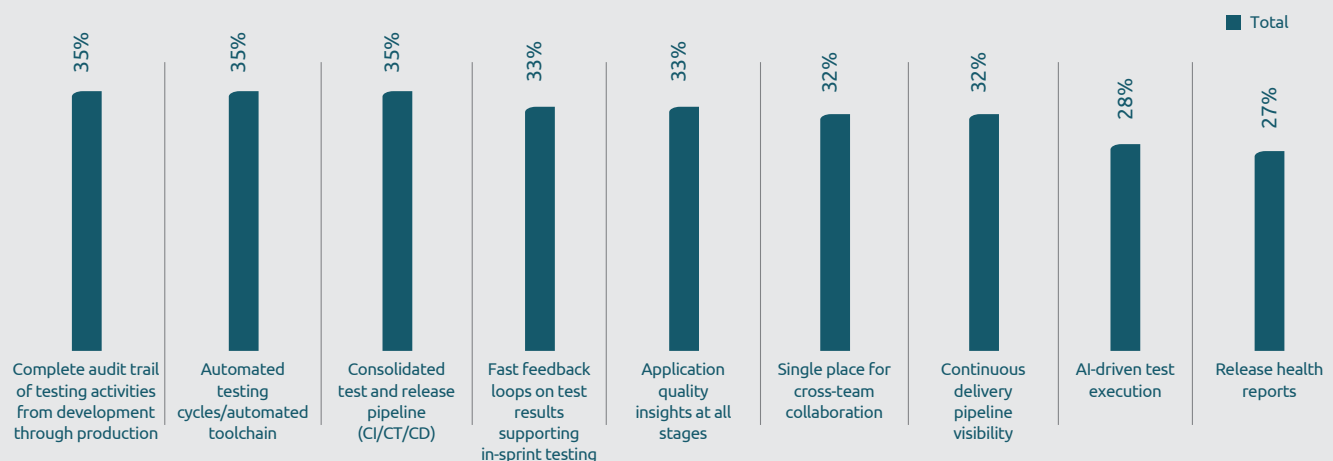
Similarly, when asked about the test orchestration capabilities that were most important to them, 35% of respondents pointed to “complete audit trail of testing

activities from development through production.” A similar number (35%) of respondents indicated “consolidated test and release pipeline,” and another 32% said that “visibility of the continuous delivery pipeline” is an important orchestration requirement for them. All three of these statements reflect the importance placed on the flow of information across the testing tool chain.

In this context, one of the ways to get better visibility into the continuous integration/continuous testing (CI/CT) pipeline is to use customized dashboards, a practice that is being adopted by many project teams and organizations. Another promising development is the rise of artificial intelligence (AI) technologies that provide “smart” test orchestration. With the addition of machine learning capabilities, systems will be able to automatically determine the tests that are required in the release and production cycles. According to the CTR survey, 28% of respondents said that they consider AI-driven test execution as an important orchestration capability.

The challenges mentioned above show the dilemma that many organizations find themselves in today. It is partly the result of how IT organizations have developed over time. Over the last few years, in an effort to be as agile as possible, many of these organizations moved towards autonomous and self-empowered teams. Functions such as QA and testing were integrated at the team level and such teams took independent decisions on things such as QA approaches and tools, etc. While this resulted in organizations becoming more agile, organizations also found that not only were they unable to leverage test results and test assets across teams, but that they were also losing control of the amount and cost of QA tools and test environments. At an organizational level, it also

Most important test orchestration capabilities



became difficult to identify misalignments in test coverage. As a result, it has now become increasingly clear that some form of central orchestration is required to overcome such challenges.

Good orchestration helps to create a coherent QA and test strategy across teams and optimize the usage of test automation and test environments. It also gives business owners a continuous view on the actual E2E state of quality in their core processes and applications. It is an approach that covers each phase and task in the release. These phases incorporate both dev and testing and allow testing tasks to be incorporated into each phase in the SDLC. Audit trails are compiled as a result of these fully tracked processes. As the release progresses through the orchestrated pipeline, real-time information is available to every team in the SDLC in a single source of truth. As each task is completed, it can be automatically moved to the next task if all requirements are met (i.e., if all tests are passed) or the person or team responsible for the next manual task can be automatically alerted, and reminders and alerts are provided and escalated if tasks are taking too long. This shortens wait times and encourages collaboration across teams.

Key recommendations

To address test orchestration challenges in the CI/CD pipeline, organizations need to:

- Create visibility of quality processes by implementing customized QA dashboards across the CI/CD pipelines
- Optimize test automation tools and test operations across agile teams
- Set up automated testing cycles, so as to shift left and identify defects early in the lifecycle, thereby enhancing product quality
- Automate the self-provisioning of test data
- Automate provisioning of test environments complete with virtual services and test data
- Leverage AI and machine learning (ML) technologies to optimize test cycles. This can include selecting the correct tests to run according to what is in the pipeline, through to provisioning test environments and running the tests. Using natural language programming (NLP), the test orchestrator can learn the pattern of test data usage and auto-generate test data with minimum manual intervention.

Use case

A leading Australian bank had over 5,000 builds for 100+ applications and 450+ QA resources. Their

biggest challenges revolved around manual build and deployment and inconsistent environments leading to frequent deployment failures.

The company implemented continuous delivery and continuous testing comprised of a single platform for automated build and deployment integrated with testing and automation tools. The solution had quality gates for auto-validation of environments leading to lower build deployment failures.

Benefits:

- Reduction in cost: the team size was reduced by 30%
- Build cycle time reduced by 40%
- Shift left and early defect detection
- Coordinated testing, development and environment teams
- Faster time to market and significant increase in environment uptime.

Conclusion

Agile and DevOps teams are driving quicker release and build cycles. However, it is imperative that companies set reasonable expectations from their continuous testing processes, as the focus on speed can impact quality. Instead of trying to outpace each other, companies should build their release pipeline based on the sector in which they operate. For instance, for a mobile application provider weekly and even daily builds are important, but for energy and utility company monthly or quarterly build would suffice.

Orchestrating tests as part of a continuous delivery pipeline will help teams to add quality measures to assess the risks associated with new releases or updates. A change in culture, mindset, and the greater uptake and standardization of tools will all have to come together to move us in this direction.

In sum, some of the key elements organizations are shooting for through test orchestration in the agile enterprise is based on the following elements:

- Continuous visibility on quality status across all CI/CD pipelines
- Optimized test operations and test automation tools across agile teams
- Automated provisioning of test data
- Automated provisioning of test environments
- AI and ML technologies to continuously and automatically optimize the test sets.

Continuous testing: the road ahead

Stephen Feloney

Head of Products, Continuous Testing Business, Broadcom Enterprise Software Division

Panchal Barua

Continuous Testing Architect and Director, Financial Services, Capgemini

Sathish Natarajan

Vice President, Digital Assurance & Testing Practice, Capgemini Group

The road to complete automation

Overview

Continuous testing enables continuous delivery with the goal being orchestrated automation of the entire pipeline with zero manual intervention during any part of the process. Such an approach implies an evaluation of quality at every stage of the continuous delivery (CD) process, starting right from the requirements-gathering phase (shift left) and continuing even after an upgrade or piece of code has been released (shift right). It also implies leveraging of artificial intelligence (AI) and machine learning (ML) elements to support such automation and improve results over time.

This is a transformation that takes time. While significant portions of the pipeline have been automated (typically regression test cases or test execution), there remain large areas, where manual intervention is required. These gaps in automation, are mostly seen in three areas:

- Moving from development to test environments
- Provisioning/configuring of required elements (test data, test environments, etc.)
- Integrating a feedback loop into the testing cycle after the application has been released.

Since a chain is only as strong as its weakest link, these gaps in automation defeat the very purpose of continuous delivery, which is to deliver better software faster and cheaper.

The next two to three years – a critical time for continuous testing

As technology advances, organizations have just started addressing some of these “gaps in automation.” Among these, there are certain low-lying fruits that can be tackled immediately, using technology currently available, in order to improve results with minimum investment. The Continuous Testing Report (CTR) 2019 survey results reveal some of the most important areas in which we are going to see significant progress over the next two to three years.

One of these areas, which has also been a long-time challenge for the industry, is test data management. According to the CTR 2019 survey results, 36% of respondents spend more than 50% of their testing time in searching, managing, maintaining, and generating test data. This is a significant amount of time, a lot of which can be saved with the help of test data automation, or the automated generation of compliant test data using software tools. Such automation is within reach for most organizations today, as it does not require companies to completely transform their testing capabilities and is a quick win to eliminate a common source of wait time in the development lifecycle.

Another step that will help immensely in moving towards continuous testing is the auto-generation of test cases. Today, the creation of test cases is a time-consuming,

manual process and auto-generation of test cases can cut down this time significantly while adding value to the testing process. Moreover, auto-generation techniques can focus on designing test cases from user requirements, rather than from the created code. This approach has proven to be helpful in improving product quality and reducing rework as the tests are designed keeping in mind the intended functionality.

Many organizations have already moved to using techniques such as model-based testing (MBT) with the help of tools commonly available on the market. Model-based testing is the foundation that must be in place for continuous testing and we are likely to see many more organizations going in for such auto-generation of test cases over the next couple of years.

Another area, that is a key bottleneck today, is test environment management. According to the CTR 2019 survey, 40% of respondents said they were spending more than 50% of their time in building, managing, maintaining, and de-commissioning test environments. Cloud adoption can help drastically cut down on this time by virtualizing the entire test environment.



Additionally, developers and testers do not need to provision entire environments as they can use service virtualization (SV) tools to virtualize parts of the environment that are either unavailable for testing, are still in development, or are third-party services. With automation, complete, ephemeral test environments can be created on demand and decommissioned when testing is complete, reducing both the time as well as the expenses associated with maintaining environments. Cloud technology can help with this and the adoption of cloud for the creation of virtual test environments is a logical step to take for automating the delivery pipeline.

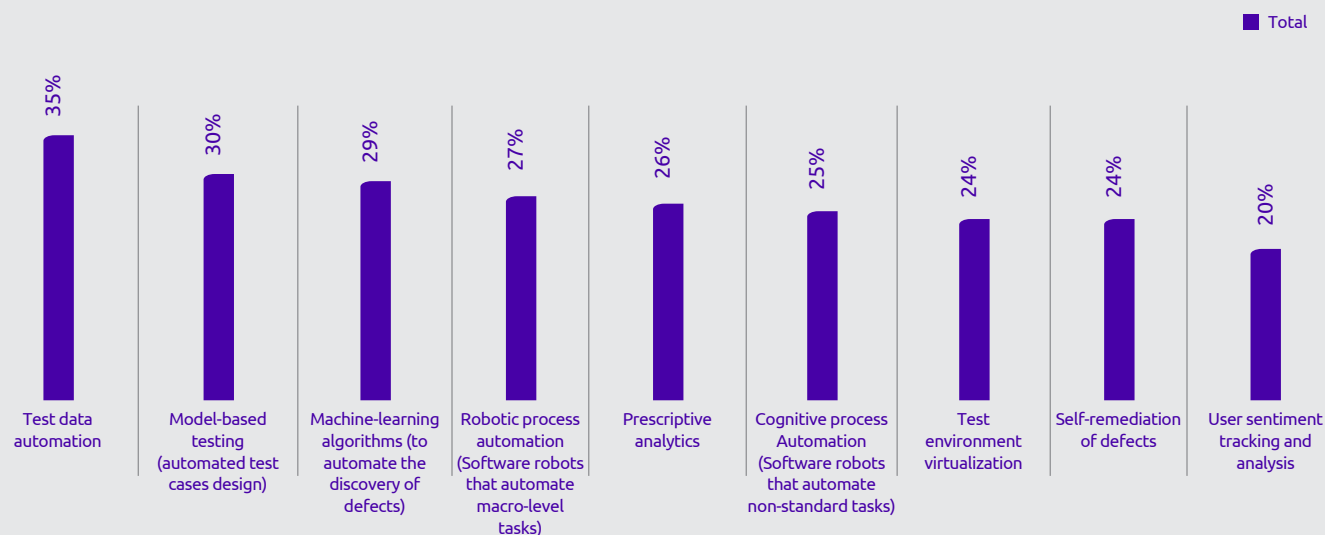
The urgency of automating these parts of the SDLC was also brought out by the CTR survey. For instance, when asked about the automation techniques they foresaw using in the coming year, respondents gave the highest ranking to test data automation (35% of respondents), followed by MBT (30%) and machine learning (ML) algorithms to automate the discovery of defects (29%). Machine learning technologies will help identify the risk level of new code;

Challenges

Despite the existence of such “low-lying fruits” or “areas of opportunity,” the move towards continuous testing is not going to be easy. The industry has evolved so quickly that most QA organizations are still struggling to figure out how to adapt to innovation. In this context, the most fundamental change seen over the last three to five years, is that processes which used to operate in silos are now connected and integrated within teams.

Such integrated, inter-disciplinary teams require not just a change in workflows and organization but also in the skill sets of team members. None of the functions are now working in a silo and thus every team member needs to have a holistic understanding of the entire process, all the way from creating user stories to putting the code in production. For QA and testing professionals, this also means that they need to upgrade their technical skills, as a lot of activities which they did not need to bother about earlier, now fall under their purview. For instance, QA and

Automation techniques expected for the following year



suggest, orchestrate, and execute tests, and incorporate data into a complete feedback loop to improve testing along the development lifecycle.

testing professionals who earlier used to be completely dependent on data base administrators or other external providers to furnish them with the test data, now need to

have both test data as well as test environment virtualization skills themselves.

This is an area most organizations are struggling with as there is a major shortage of skills required. To tackle this a number of companies have set up exhaustive training programs, in which resources are trained in new technologies and tools.

Another big roadblock is the legacy mindset. A lot of testers still believe that they need to have the entire application before they test. This slows down the process and goes against the very grain of the agile and DevOps philosophy. In addition, there are also significant challenges with regard to test data and test environment that have been touched upon in the earlier section.

Recommendations

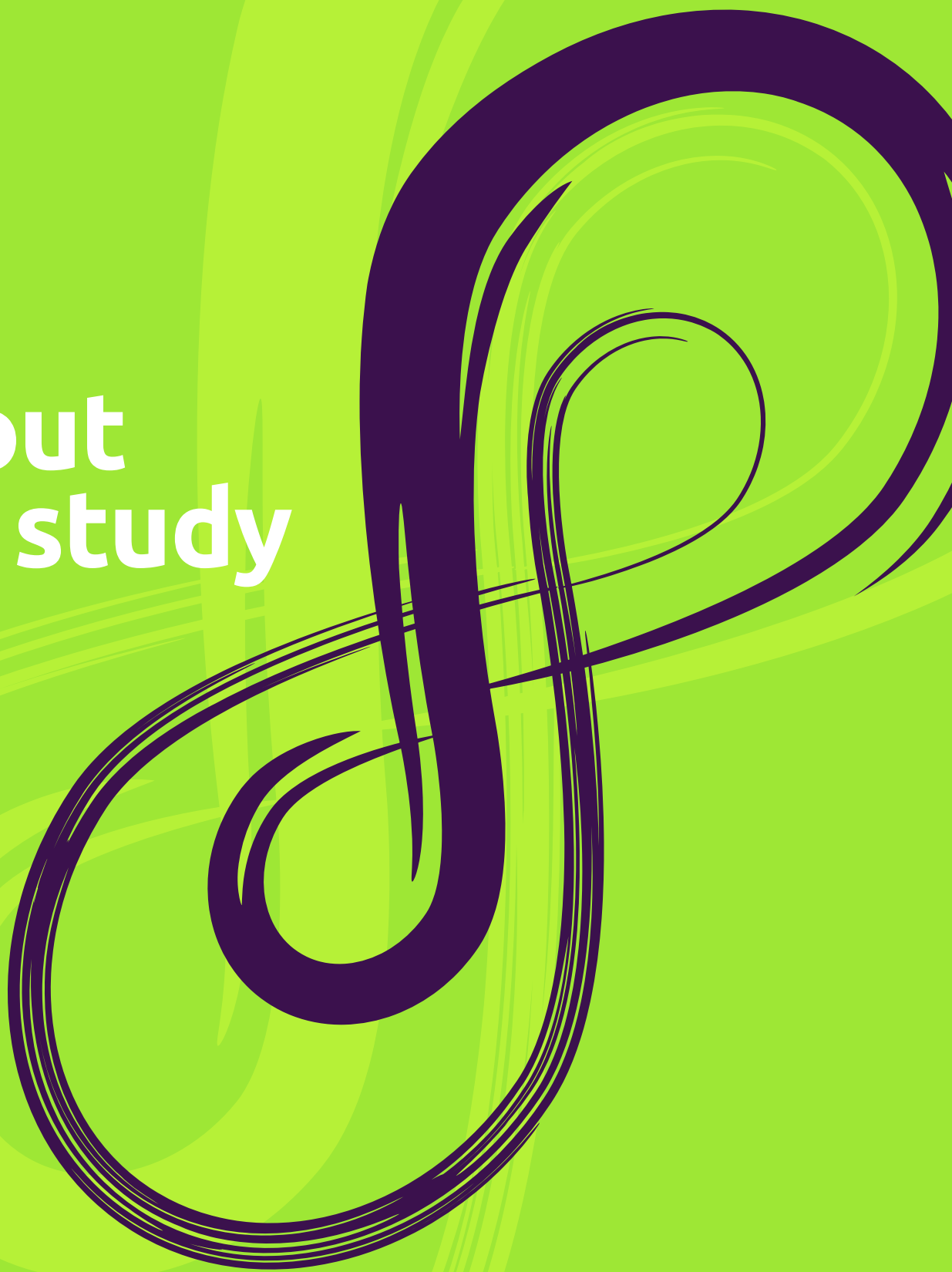
- Start small – Instead of applying continuous testing across a business, enterprises should look at identifying a single product or group that is willing to adopt the practice and serve as an example of success to the rest of the organization.
- Get baselines – If you do not have something to base things on you won't know how to improve. Once you have your baselines, you will know where your biggest pains are, and you can resolve them.
- Start to automate at a lower level.
- Then, use the data you have collected to improve over time. Add automation where it will have the biggest impact. Eliminate manual handoffs. Over time, you will see how much you have improved and which areas are right for additional attention.

Conclusion

To truly achieve continuous testing, there are certain prerequisites that must come into place. Enterprises need to have the infrastructure that supports the ability to better manage their test data, the ability to stand up environments quickly and remove them if not required, the ability to populate the environments with the right data, and the ability to auto-generate and maintain test cases and integration of the automation test suites so that the test can be run. In addition, automation needs to happen at the API level and agile teams need to be involved with testing. Moving forward, it is also becoming increasingly important to make sure that the tool choices being made are developer friendly and to orchestrate the entire process, i.e. "automate the automation."

We have already seen the challenges in terms of culture, mindset, and skill sets that organizations will need to tackle before continuous testing can be achieved. Solving these challenges can be tricky, but at the same time we have also seen a number of easily achievable changes that can be addressed immediately. Over the next two to three years, we expect to see a general move towards model-based testing, test-data automation, and test environment virtualization in the immediate future.

About the study



About the study

The Continuous Testing Report is based on research findings from 500 interviews carried out during June–July 2018 using an online survey approach with some telephone interviews where required and preferred by respondents. The average length of each interview was approximately 20 minutes and the interviewees were all senior decision-makers in corporate IT management functions, working for companies and public sector organizations across eight countries.

The interviews were based on a questionnaire of 30 questions, with the actual interview consisting of a subset of these questions depending on the interviewee's role in the organization. Quality measures were put in place to ensure that the questionnaire was understood, answered accurately, and completed in a timely manner by the interviewee.

For this research, we selected only organizations with more than 1,000 employees (in the respondent's national market).

Research participants were selected so as to ensure sufficient coverage of different regions and vertical markets to provide industry-specific insight into the testing and data extraction practices within each sector.

The research sample consists mainly of senior-level IT executives as shown in Figure 31.

To ensure a robust and substantive market research study, the recruited sample must be statistically representative of the population in terms of its size and demographic profile.

The required sample size varies depending on the population it represents – usually expressed as a ratio or incidence rate. In a business-to-business (B2B) market research study, the average recommended sample size is 100 companies. This is lower than the average sample size used for business-to-consumer (B2C) market research because whole organizations are being researched, rather than individuals.

As mentioned above, the B2B market research conducted for the Continuous Testing Report is based on a sample of 500 interviews from enterprises with more than 1,000 employees (40%), organizations with more than 5,000 employees (40%) and companies with more than 10,000 employees (20%).

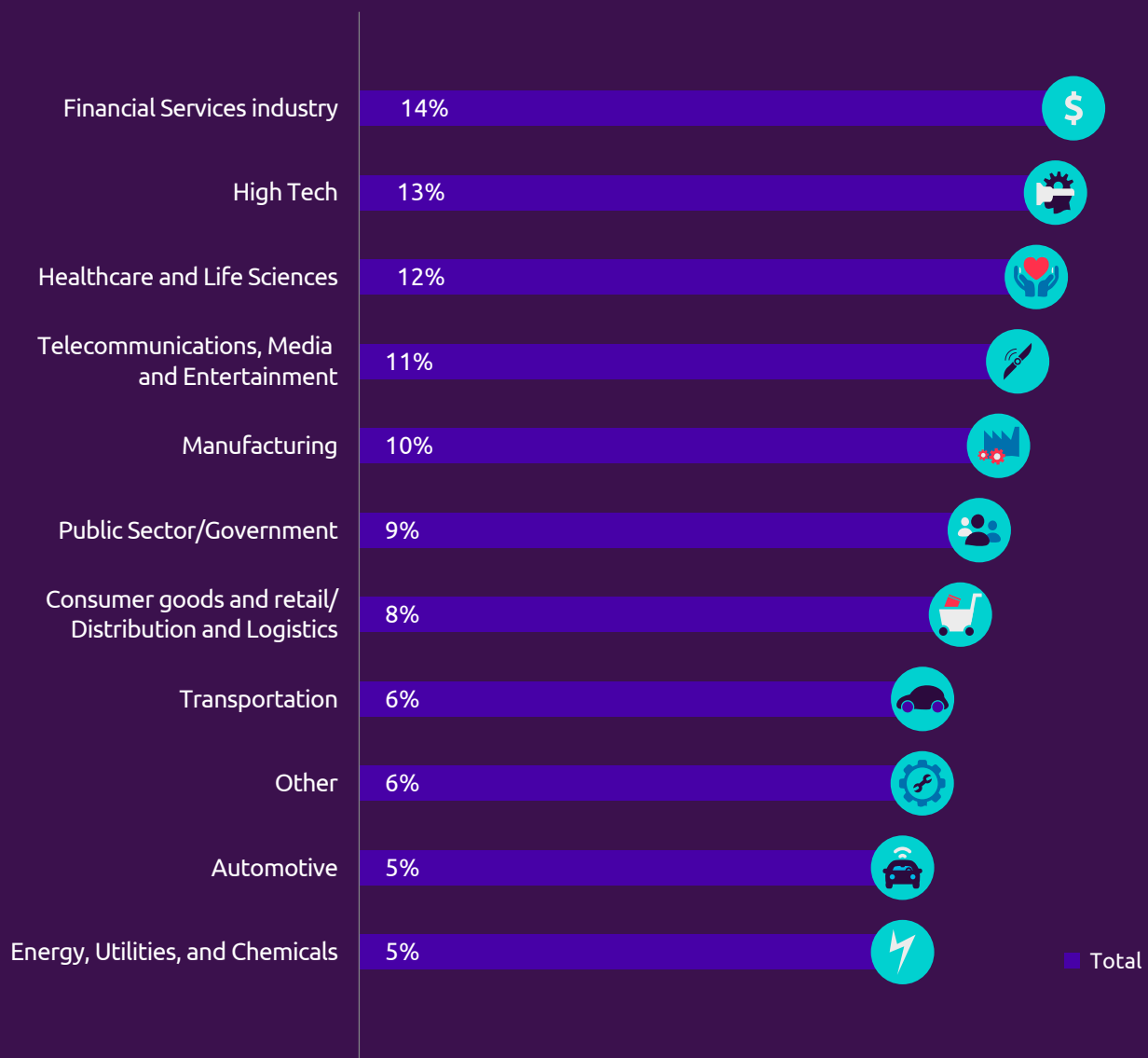
During the interviews, the research questions asked of each participant were linked to the respondent's job title and the answers he/she provided to previous questions where applicable. For this reason, the base number of respondents for each survey question shown in the graphs is not always the full 500 sample size.

The survey questionnaire was devised by QA and testing experts in Capgemini, in consultation with Coleman Parkes Research. The 30-question survey covered a range of testing and data extraction subjects.

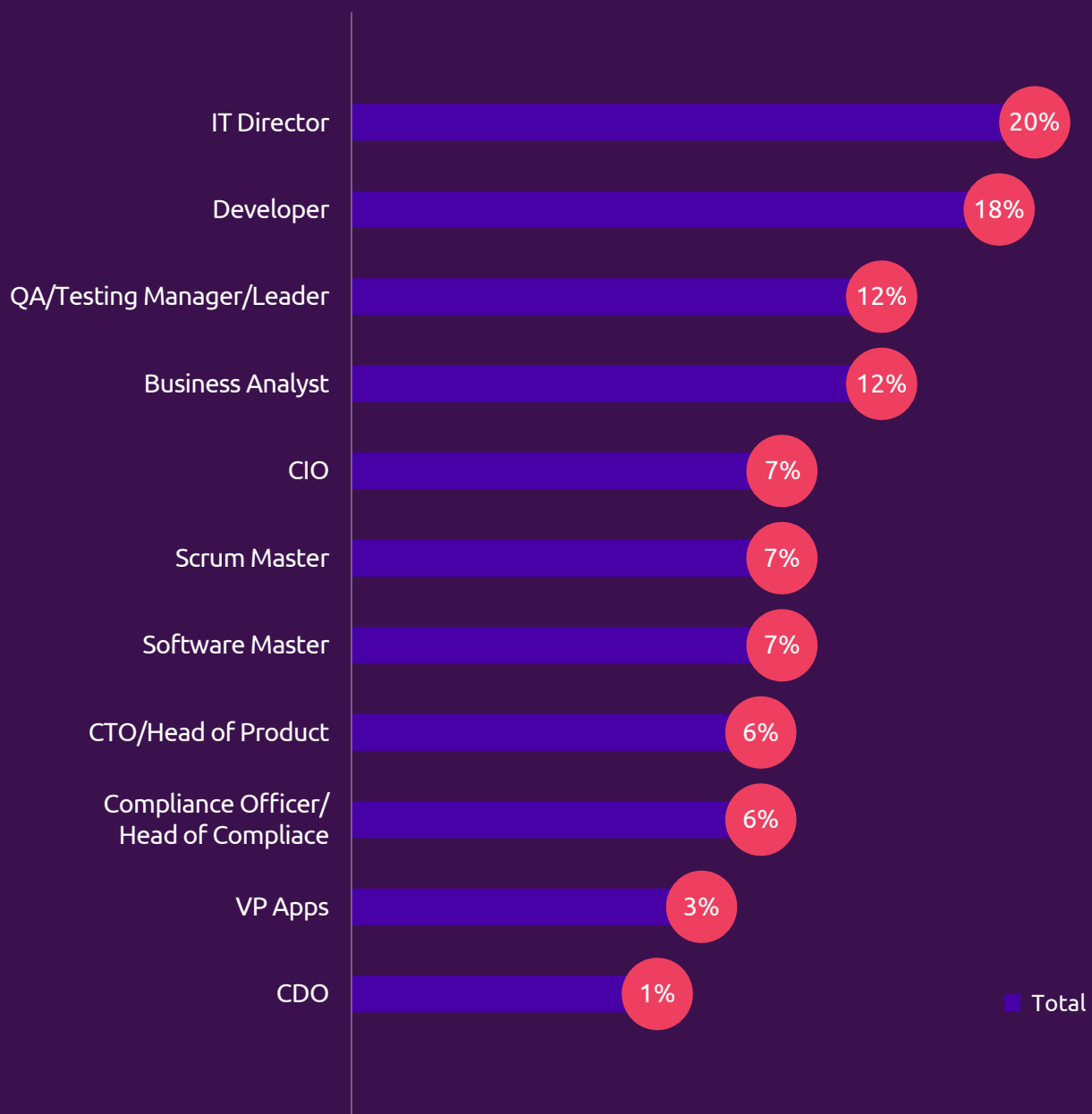
Interviews by region



Interviews by sectors



Interviews by job title



Thank you

Capgemini, Sogeti, and Broadcom would like to thank

The 1,700 IT executives who took part in the research study this year for their time and contribution to the report. In accordance with the UK Market Research Society (MRS) Code of Conduct (under which this survey was carried out) the identity of the participants in the research study and their responses remain confidential and are not available to the sponsors.

All the business leaders and subject matter experts who provided valuable insight into their respective areas of expertise and market experience, including the authors of country and industry sections and subject-matter experts from Capgemini, Sogeti and Broadcom.

Main Report Authors

Christine Bensten, Antoine Aymer and Ravikumar Sengodan

Writer for Main Chapters

Rahul Mitra, Jayant Kumar and Sayantani Das

Creative Design

Priyanka Tipnis, Manas Kar

Global Marketers

Anubhav Gulani, Archit Revandkar

Partner Management

Marianne Kantor

Content Proof Reading

Monica Kwiecinski

Market Research

*Stephen Saw, Rachel Leafe and Ian Parkes
(Coleman Parkes Research)**

*Ian Parkes, CEO and co-founder of Coleman Parkes Research, is a full member of the Market Research Society. All research carried out by Coleman Parkes Research is conducted in compliance with the Code of Conduct and guidelines set out by the MRS in the UK, as well as the legal obligations under the Data Protection Act 1998.

Capgemini and CA, a Broadcom Company, and their respective marks and logos used herein, are trademarks or registered trademarks of their respective companies. All other company, product and service names mentioned are the trademarks of their respective owners and are used herein with no intention of trademark infringement. Rightshore® is a trademark belonging to Capgemini. TMap®, TMap NEXT®, TPI®, and TPI NEXT® are registered trademarks of Sogeti, part of the Capgemini Group.

No part of this document may be reproduced or copied in any form or by any means without written permission from Capgemini and Broadcom.

About the sponsors

About Capgemini and Sogeti

A global leader in consulting, technology services, and digital transformation, Capgemini is at the forefront of innovation to address the entire breadth of clients' opportunities in the evolving world of cloud, digital, and platforms. Building on its strong 50-year heritage and deep industry-specific expertise, Capgemini enables organizations to realize their business ambitions through an array of services from strategy to operations. Capgemini is driven by the conviction that the business value of technology comes from and through people. It is a multicultural company of 200,000 team members in over 40 countries. The Group reported 2017 global revenues of EUR 12.8 billion.

Sogeti is a leading provider of technology and engineering services. Sogeti delivers solutions that enable Digital Transformation and offers cutting-edge expertise in Cloud, Cybersecurity, Digital Manufacturing, Digital Assurance & Testing, and emerging technologies. Sogeti combines agility and speed of implementation with strong technology supplier partnerships, world-class methodologies and its global delivery model, Rightshore®. Sogeti brings together more than 25,000 professionals in 15 countries, based in over 100 locations in Europe, USA, and India. Sogeti is a wholly-owned subsidiary of Capgemini SE, listed on the Paris Stock Exchange.

Learn more about us at:

www.capgemini.com/continuous-testing

www.sogeti.com/testing

About Broadcom

Broadcom Inc. (NASDAQ: AVGO) is a global technology leader that designs, develops and supplies a broad range of semiconductor and infrastructure software solutions. Broadcom's category-leading product portfolio serves critical markets including data center, networking, enterprise software, broadband, wireless, storage, and industrial. Our solutions include data center networking and storage, enterprise and mainframe software focused on automation, monitoring and security, smartphone components, telecoms and factory automation. For more information, go to

www.broadcom.com.



Broadcom

Christine Bensten

Head of Product Marketing,
Continuous Testing Business,
Broadcom Enterprise
Software Division

Capgemini

Anand Moorthy

Vice President, Financial
Services, Capgemini

Ajay Walgude

Vice President, Financial
Services, Testing Leader,
Capgemini

Dhiraj Sinha

Vice President, Financial
Services- Testing, Asia Pacific

Deepika Mamnani

Senior Director, Financial
Services, Capgemini

Sanjeev Deshmukh

Vice President, Apps Services,
North America, Capgemini

Ramesh Mahadevan

Senior Director, Testing,
Capgemini

Sogeti

Antoine Aymer

Group Offer Director,
Sogeti

Sathish Natarajan

Vice President, Digital
Assurance & Testing Practice,
Capgemini Group

Maheshwar Kanitkar

Senior Director,
Sogeti

In association with



A Broadcom
Company