

Test Environment Management: Optimizing Software Development

A point of view from Capgemini Group.

Performance driven. Quality assured.



Organizations can save significant amounts of software development time and costs if they address the vital area of **Test Environment Management (TEM)**.

The fifth edition of the World Quality Report¹ from Capgemini, Sogeti and HP shows that organizations have become more mature in testing and more aware of quality requirements. This is clearly demonstrated by the rapid growth in the number of companies centralizing their testing functions and building Testing Centers of Excellence (TCOEs).

As organizations improve on adapting their processes, operational models and testing tools to maximize Quality Assurance (QA) efficiency, a new area of concern emerges — the setup and management of test environments.

The World Quality Report 2013-14 (a global study of application quality and testing practices, based on a survey of over 1,500 CEOs, CFOs, CIOs, IT Directors/Managers and Quality Assurance Directors/Managers in 25 countries) confirms that a suitable test environment is one of the most fundamental requirements for effective software testing.

The latest research found that on average 40% of today's testing budgets are being allocated to test infrastructure and hardware, with an additional 28% spent on testing tools. Furthermore, managing test infrastructure is requiring increased resource allocation due to the growing complexity in architecture and the variety of demands made on test environments. Organizations typically require multiple separate instances of test environments — anywhere between one and 250.

In addition, the separation of responsibilities between the testing organization that defines and uses the test environments and an infrastructure team that provides those environments is making this even more challenging. Therefore, more organizations today are looking to transfer the responsibility for Test Environment Management to their testing organization.

Neglecting TEM — now a vital part of software delivery lifecycle (SDLC) management — can create the following issues:

- A need for higher investment in the organization's non-production (test) environment that could otherwise be used in more significant, business critical areas.
- Costly increases in storage requirements due to the rapid expansion of mobile, big data and social media.
- Higher levels of disruptive change in organizations' non-production (test) environments, versus their operational live production environment.
- The failure to provide the correct test environment in time and insufficient environment support to testing teams causes significant inefficiencies in testing teams — up to 20% and more.

Guidance for introducing Test Environment Management

At a time when organizations are seeking any advantage over their competition, particularly in key IT performance areas like total cost of ownership (TCO) and product time to market (PTTM), Test Environment Management can underpin these aims more than some senior managers realize.

The main reason for this is that while test organization maturity improves, companies' test environment capability is often still falling behind. The test environment provisioning then becomes a source of delay that negates many of the gains derived through organizations' functional and non-functional test and upstream team improvement programs.

Many companies spend an estimated 25% of their average production environment costs on their non-production systems (development, test, performance, user acceptance testing and staging). It is therefore fundamental for them to ensure that their investment is:

1. Fit for purpose.
2. Properly managed.
3. Closely controlled.
4. Fully optimized.

1. Fit for purpose.

All software projects and activities need defined requirements — and Test Environment Management is no exception. Unfortunately, many organizations fail at this fundamental first step.

When setting up their test environment, delivery teams often only attempt to replicate the production systems environment exactly within the non-production space. Whilst this makes sense — for some forms of testing — from a quality perspective, financially it is not always the most prudent approach.

Without a clear set of requirements from the outset, it is impossible to define the optimal set up of a test environment. Therefore, TEM teams often over-compensate by over-delivering on the environment components and accompanying documentation. This adds significant overheads (time, cost) to their delivery schedules, and introduces costly waste (re-work, maintenance, errors) into the SDLC process right from the beginning of a project.

Most software process improvement programs and test process improvement programs are now geared towards eliminating waste and error. As Test Environment Management is becoming a weak link, a first solution must be a proper definition of the 'real' requirements on the test environment.

¹ A full copy of the World Quality Report 2013-14 can be downloaded from <http://www.worldqualityreport.com>

The best TEM teams or processes deliver ‘just enough’ for the test teams to execute their required scope of work for that particular project phase.

A test environment is inherently different from production configurations. It does not always have to be the exact copy of a production environment, because it is being used differently from production. Test requirements have to be set up to accommodate frequent backup and restore procedures, and must be able to manage multiple versions of test data sets, as well as run specialized software for testing, such as stubs, drivers and data generation tools.

The degree to which the test environment represents the production conditions depends on the overarching test goals. For testing the application functionality early in the development process, a laboratory environment is usually sufficient. For user acceptance testing or performance testing, testers need a setup that more closely mirrors a production environment.

Not all parts of testing require permanent environments. The WQR research shows that a quarter of respondents use only a temporary environment; and cloud-based testing platform-as-a-service solutions account for almost half of those temporary environments. However, hardware and software that is set up exclusively for testing purposes is a standard prerequisite for efficient, predictable and repeatable testing.

2. Properly managed.

The best environment managers are inquisitive by nature and have a high degree of common sense. They need to understand what it takes to make the environment work. For example, when assessing requirements compared to the solution architectural documentation, they need to be able to tell whether the requirements are excessive, sound, or if there are any areas that could be better qualified.

As the TMAP^{®2} methodology states, test managers need to begin the requirements scoping for a test environment well in advance of the start of the testing execution phase. QA organizations that fail to define the required test environment components and fail to engage the necessary Test Environment Management team early during the test planning stage may find that, by the time they are ready to execute test cases, the test environment setup is not yet complete. Any delays downstream in the solution delivery lifecycle have a compounding effect on

the overall project delivery schedules and can undermine any gains achieved upstream.

It is vital to have a solid grounding in the underlying business processes and access to the right people. Consider how well-documented the business processes are within your organization. In the absence of this knowledge, most environmentmanagement (and test) teams are flying blind, through no fault of their own.

Through effective management practices, projects can certainly co-exist and share the same development and test environments, without interfering with each other. This is a significant source of cost saving from a project SDLC delivery perspective. Two projects sharing an environment each pay half of the total cost of an environment, and three projects pay just a third.

To make environment-sharing work effectively, project pipeline management and project-to-application traceability is a must. The organization will also need to develop its control processes, co-ordination and communication, in order to ensure stakeholders' expectations are managed effectively.

One common risk management practice within organizations is maintaining multiple test environment instances of the same application, in order to facilitate parallel product development and to isolate changes from each stream of work. The challenge with this method is resolving how all of these instances can be maintained, retrieved and managed in an efficient way.

Over half (53%) of the 1,500 executives interviewed for the WQR 2013-14 study state that having to maintain multiple test environment versions, hardware, middleware and systems presents a significant challenge. Together with the fact that 45% of those interviewed report lack of clarity on efficient usage of available configurations as one of the main hurdles in provisioning a test environment, this shows that many organizations may still have a long road ahead of them before getting the balance right between risk, design, management, support and investment in their test environments.

Ultimately, projects may have no choice but to share resources (for example, solutions which are simply too costly to build new). Here, soft management skills in areas like negotiation and conflict resolution are required; the Test Environment Manager must be able to effectively manage multiple stakeholder groups, schedules and business priorities.

The best TEM team leaders have a strong project management background and a flair for the technical; their soft skills are well-developed and their communication and risk management abilities are of a high standard.

² TMap[®], or Test Management Approach, is Capgemini Group and Sogeti's world leading business-driven, risk-based methodology for structured software testing. The methodology offers a comprehensive and consistent approach to project and program test management, execution and quality assurance. www.tmap.net

As with many disciplines, Test Environment Management is about delivery, and having strong managers with a track record of technical delivery is desired.

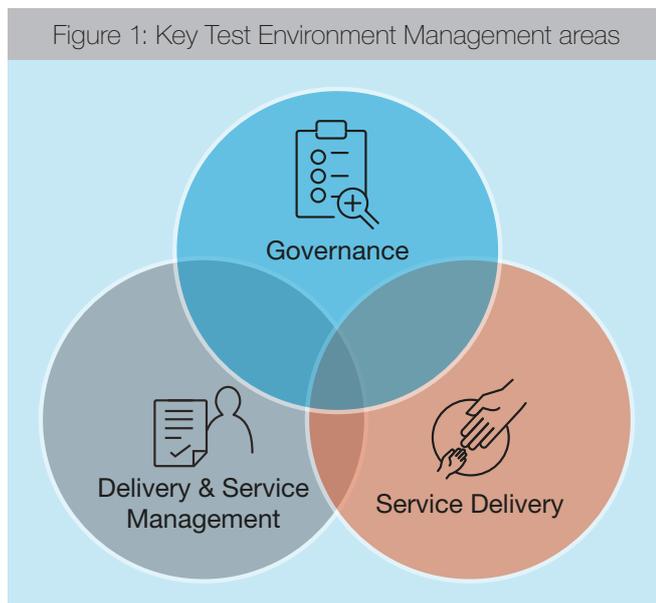
3. Closely controlled.

Putting non-technical project managers (PMs) in TEM delivery roles is a risky strategy. TEM requires additional skills to be used in conjunction with traditional project management competencies, in order to allow for the successful delivery and support of test environments. These are:

- Technology knowledge.
- IT Service Management knowledge.
- Configuration knowledge.
- Lean processes and controls governance.
- Tooling knowledge.

Management of test environments can be divided into three key overlapping areas:

- Governance.
- Project Delivery and Service Management.
- Service Delivery.



Governance

The Test Environment Management Governance team is responsible for the TEM delivery framework, which includes:

- Policies.
- Standards.
- Processes (workflows and controls).
- Productivity (Service Level Agreements).
- Measurement.
- Compliance.

Using a services-based delivery model allows the activities in which the TEM team performs to be simplified. By defining a service catalogue, standardized requirements and engagement models can be defined, including the relevant processes and workflows. The effectiveness of the services ‘front door’ can be industrialized and controlled via adequate tooling, including the measurement of Service Level Agreements to ensure productivity and throughput meet expectations.

It is important to agree which services will be provided at an organizational level. This will allow the enterprise to better focus its processes and controls on those services that the TEM team is accountable for. The services that the TEM team manages are likely to grow quickly; and having highly attuned and innovative managers makes the work effort more manageable.

On average TEM teams manage up to 10 times as much change as their operations (live production) counterparts. Therefore, implementing laborious ‘heavy’ processes and controls simply negates any wins made in provisioning and test efficiencies.

Instead, enterprises need to introduce lean (simplified and light) processes and controls based on ITSM and ITIL industry best practice. TEM teams need to maintain control whilst maximizing throughput; this is where your Test Environment Manager’s risk management skills come to the fore.

As with all stages of the application delivery cycle, Test Environment Management needs to be closely aligned with project delivery and release schedules. If a QA organization maintains a permanent test environment, but lacks the right levels of controls and change movement policies, it will likely run into project delays. Getting the right balance between change agility and risk management, supported by the right tooling solutions, is crucial to the success of the test phase and the products’ time-to-market. This is the primary role of the Test Environment Management governance body.

Project Delivery & Service Management

The Test Environment Management Delivery and Service Management teams are responsible for planning, design, delivery and solution support management.

There are five options to consider when looking to deliver a test environment:

1. Use an existing environment.
2. Build a new environment.
3. Share an environment.
4. Virtualize the service.
5. Re-schedule the activity.

Each option has positives and negatives based on the circumstances at the time and should be evaluated before starting the test environment design activity.

When delivering a new environment, standard project management methodologies work best and integrate well into wider, existing organizational workflows and processes. Test Environment Managers must have effective PM skills, and this is greatly reliant on delivery experience and expertise.

One known way to keep costs down is to use innovative infrastructure solutions. The TEM team can add significant value to the organization's SDLC through their understanding of the needs of the testing organization, and recognizing what the lowest-cost solution is within organizational constraints and policies.

Service delivery

The Test Environment Management Service Delivery team is responsible for the execution of specialist support activities in the test environments.

Most organizations have specialist resources or teams of resource to provide services such as:

1. Database administration.
2. Systems (server) administration.
3. Desktop administration.
4. User access administration.

Test Environment Managers plan, design, schedule and control the delivery services of these resources or teams, in order to meet the entry criteria of the testing services team.

Optimizing the engagement, interactions (collaboration) and execution activities of these specialist teams can significantly improve the productivity and time-to-market of the overall product.

An important element of the team's work is agreeing a standard of services delivery control. This could be simply defining clear roles and boundaries of responsibility between teams, standardizing engagement processes, introducing common metrics or introducing more automation into execution activities.

Knowing where to obtain the greatest returns for the least investment becomes important from a business perspective and having a clear understanding of your present capability and areas of weakness is crucial to any business case proposals.

4. Fully optimized.

In Capgemini and Sogeti's view there are four main aspects to consider for achieving optimized TEM:

- Test asset rationalization.
- Lean process engineering.
- TEM automation.
- Test Data Management.

Test asset rationalization is the fastest, simplest and most cost-effective way to get senior leaders to pay attention to Test Environment Management, and involves essentially cutting the number of servers and cost needed to support the organization's SDLC.

Lean process engineering is another mechanism which can be used, and which will enhance TEM team throughput whilst minimizing resourcing overheads.

TEM automation is another area that can produce tangible benefits, especially in incident reduction (error) and build effort. There are many different ways to introduce automation into Test Environment Management. These include:

- Automated code packaging and deployment or builds.
- Shake-down test automation (aka re-baseline testing of a test environment post build to ensure new defects have not been introduced).
- Systems monitoring and health check solutions (aka understanding the status of your applications' core services).
- Baseline comparisons of configuration files.
- Database integrity checks.
- Website hyper-link crawlers (to ensure there are no broken links of target websites).
- Workflow ticketing (to ensure approval control processes, audit data and metric generation is as seamless as possible).
- Reporting.

A key obstacle to introducing automation for many organizations is the investment and budget required. The TEM team must provide sound business cases with ROI analysis to enable management decisions on investing in automation.

Test Data Management (TDM) also comes into play here, in successfully implementing IT application-sharing whilst delivering substantial storage cost savings. This is especially beneficial at a time when organizations are seeing their storage requirements grow by up to 30% a year, due to the proliferation of 'big data' and the adoption of technologies like mobile devices and social media.

With businesses collecting ever-increasing volumes of data relating to their day-to-day operations and customer preferences, managers are under more pressure than ever to analyze this data, interpret how it can be used and share it with different people and groups within the business.

This same data is frequently being used by new and more complex business applications. At the same time, organizations can be targeted by external parties seeking to gain unlawful access to confidential and personal data for fraudulent criminal purposes. Since the same data set is frequently shared across multiple applications and used for different purposes simultaneously, it is difficult to test live data without the danger of disrupting production systems. There are also issues regarding compliance and data protection.

TDM addresses the following needs:

- Reducing risk and protecting reputation.
- Reducing cost – in many cases moving CAPEX off the balance sheet.
- Improving customer experience – fewer errors and greater data integrity.
- Faster speed to market/new revenue.

Backwards data migration is also a complex issue (e.g. banks updating their core systems and ensuring they don't lose records of existing loans and mortgages).

Test Data Management is an essential component of the software testing process. To accurately predict testing outcomes, QA needs reliable sources of well-defined and consistent data as input for the testing process.

However, today's organizations often struggle with the increasingly large volumes and configurations of data required for testing, which are exaggerated by the need to maintain multiple versions of data in the test environment. Accurate test results can only be attained with consistent data sets that contain a controlled variety of data instances that, in turn, represent various real-life situations in the production environment. An automated, repeatable testing process is unachievable without well-organized and predictable test data sets.

The operational need to obtain large quantities of representative test data of sufficient complexity drives many organizations to use production data as a source for their test data – especially when the testing process doesn't clearly define requirements for test data sets. Executives interviewed for the World Quality Report state that they use copies of production information for testing purposes as much as 26% of the time. This approach also makes it easier for QA to closely mimic production conditions and, if needed, volumes.

However, copies of production data are not always the best option – especially for functional testing. Pre-defined or specially crafted test data sets with appropriate variations work much better for validating application functionality.

The research for the latest World Quality Report found that many organizations don't apply any anonymization to their production information in half the instances when using production data for testing. Test data that is taken directly from production sources, without first being manipulated, can create worrying data privacy and security issues. Data masking and sub-setting should be employed to reduce security risks and maintain data quality. Alternatively, using test data that doesn't contain any personal or sensitive information can also be considered appropriate for cloud-based testing.

Organizations often use a combination of methods for obtaining test data – the most popular approach being to

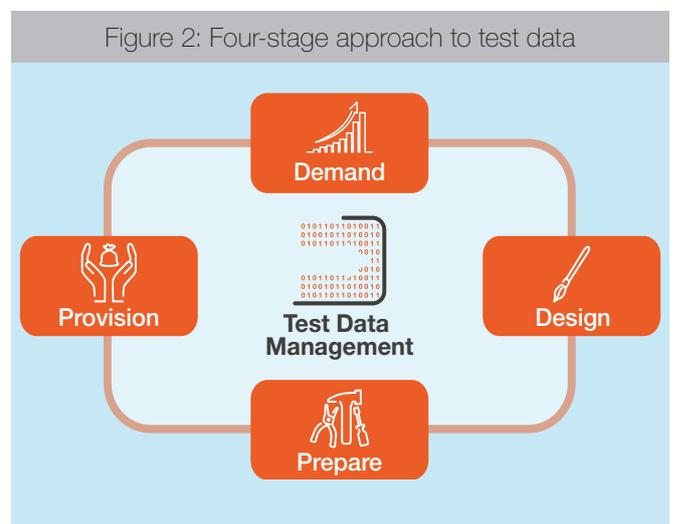
restore back-end data with each iteration and then allow its re-use for the same tests (21%). The use of managed test data sets and restoring back-end methods allows testers to create a repeatable process and therefore this is ideally suited for automated testing.

Research suggests that over a quarter of all methods used involve automated test generation – employing both custom-built and commercial test generation tools. Automatically created test data also lends itself well to repeatability and automation, provided that it is properly stored and maintained.

The World Quality Report research questions covering test generation methods also found that many organizations still prefer to create new test data, as opposed to re-using and adapting existing data sets. In order to deliver accurate results, QA organizations need to carefully consider and apply knowledge of application functionality and design, to ensure they are building the right sets of data to properly validate the features for each release.

The research results also show that, increasingly, organizations struggle to source test data that is fresh, accurate, comprehensive and sufficiently complex to deliver accurate results for testing today's complex application landscape. Managing data in a test environment that spans multiple applications and even crosses organizational boundaries, in business-to-business processes, requires extensive knowledge of not only applications, but their interconnections and the data structures of each individual part of the landscape.

In Capgemini and Sogeti's view, successful Test Data Management (TDM) is based on a four-stage approach – understanding the **Demand** for test data in the particular client situation; turning the demand into a **Design** of the test data; a **Prepare** stage of creating the scenarios for selecting, masking and generating the test data; and finally **Provision** of the right test data into the right environment at the right time.



To share test environments effectively, you need to have an integrated project management, service management, Test Data Management and knowledge management capability.

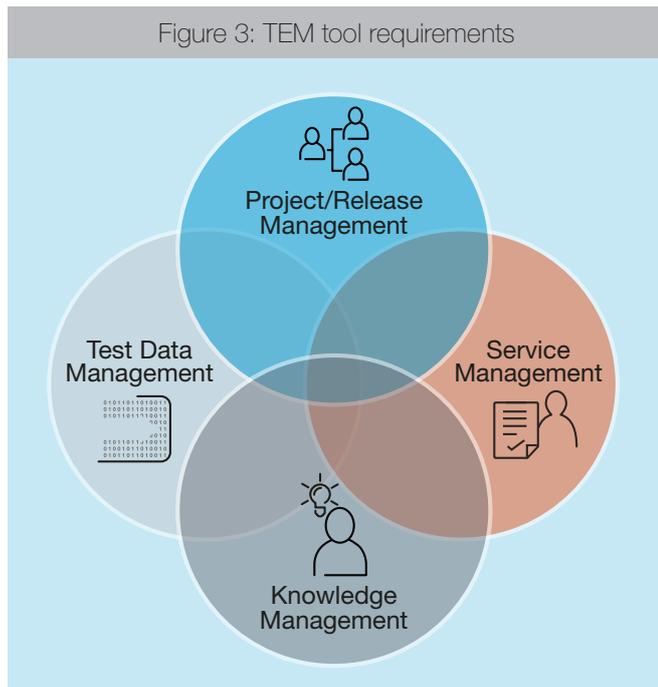
This integration proves challenging for many organizations because:

- They get distracted by software already in use within their organization, and try to solve their problems by patching these systems together.
- A tool is in use but has grown organically over time in a reactive way, based on specific needs at a particular point in time or for a particular project/deliverable.
- Tools have been developed in silos within teams, and processes are entrenched in the operations of that particular team or division.

Whilst it is possible to administer testing processes manually, using a good, lightweight integrated TEM tool will make things much easier for the TEM team. This will also maximize the productivity of much of the SDLC team – from project managers and business analysts through to developers, testers and operations staff. In this way, you will shorten the release cycles of your products, resulting in faster product time to market (PTTM).

Finally, holding the Test Environment Management data in a single repository (source of truth) adds significant weight to the underlying data and business information (reports), especially if you are using a central tool across many distributed and diverse technology stacks and geographical locations or time-zones.

Figure 3: TEM tool requirements



How we can help

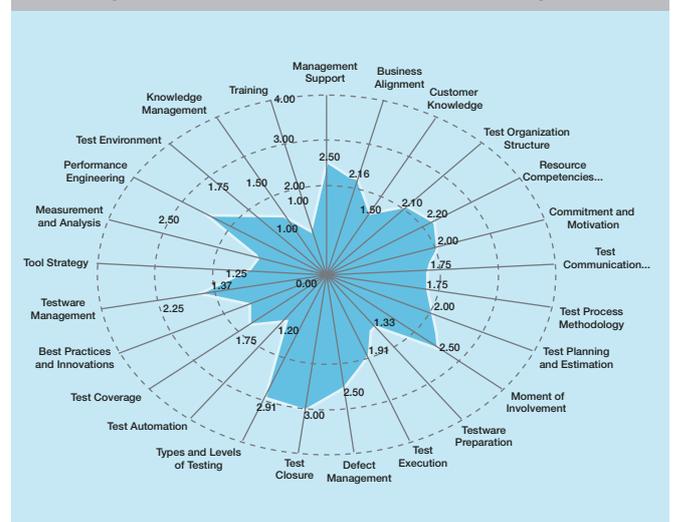
Capgemini and Sogeti can help organizations overcome the challenges outlined here.

Our detailed, robust, flexible, proprietary TEM and TDM offerings include operating models, a services body of knowledge (workflows, processes, standards, guidelines) and delivery frameworks based on the lean implementation of industry best-practice project management (PMI, PRINCE2) and service management (ITSM/ITIL) methodologies. These methods are adaptable for any of the solution delivery methodologies and techniques popular in organizations today – waterfall, iterative, agile, CI, ADD and TDD.

We use our TPI/Quality Blue Print (QBP) methodology, combined with global project management and ITIL methodologies, to perform Test Environment Management, Test Data Management and Development Operations capability maturity assessments (a global first).

This allows industry comparisons across all sectors and geographical locations – ask your local Capgemini and Sogeti representative for more information.

Figure 4: Example QBP assessment diagram



Capgemini and Sogeti also offer Managed Test Environment Management Services (MTEMS) which are globally scalable.

To access our TEM, TDM and DevOps fundamentals checklists go to:

<http://www.capgemini.com/testing-services/test-environment-management>

<http://www.uk.sogeti.com/Our-Services/Software-Testing-Services/Application-Testing-Services/Test-Environment-Management/>

About Capgemini and Sogeti

With almost 140,000 people in over 40 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2013 global revenues of EUR 10.1 billion. Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they want. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

Sogeti is a leading provider of technology and software testing, specializing in Application, Infrastructure and Engineering Services. Sogeti offers cutting-edge solutions around Testing, Business Intelligence & Analytics, Mobile, Cloud and Cyber Security, combining world class methodologies and its global delivery model, Rightshore®. Sogeti brings together more than 20,000 professionals in 15 countries and has a strong local presence in over 100 locations in Europe, USA and India. Sogeti is a wholly-owned subsidiary of Cap Gemini S.A., listed on the Paris Stock Exchange.

Together Capgemini and Sogeti have developed innovative, business-driven quality assurance (QA) and Testing services, combining best-in-class testing methodologies (TMap® and TPI®) to help organizations achieve their testing and QA goals. The Capgemini Group has created one of the largest dedicated testing practices in the world, with over 12,300 test professionals and a further 14,500 application specialists with Testing experience, notably through common centers of excellence with testing specialists developed in India and elsewhere.

For more information, please visit
www.capgemini.com/testing or
www.sogeti.com/testing

Contact

To find out how Capgemini and Sogeti's Testing Services can help your organization achieve its Testing and QA business goals, please contact your local Capgemini or Sogeti testing representative or our Global Testing Services Sales Team:

Mark Buenen
Vice President,
Business Development,
Testing Global Service Line
mark.buenen@sogeti.nl