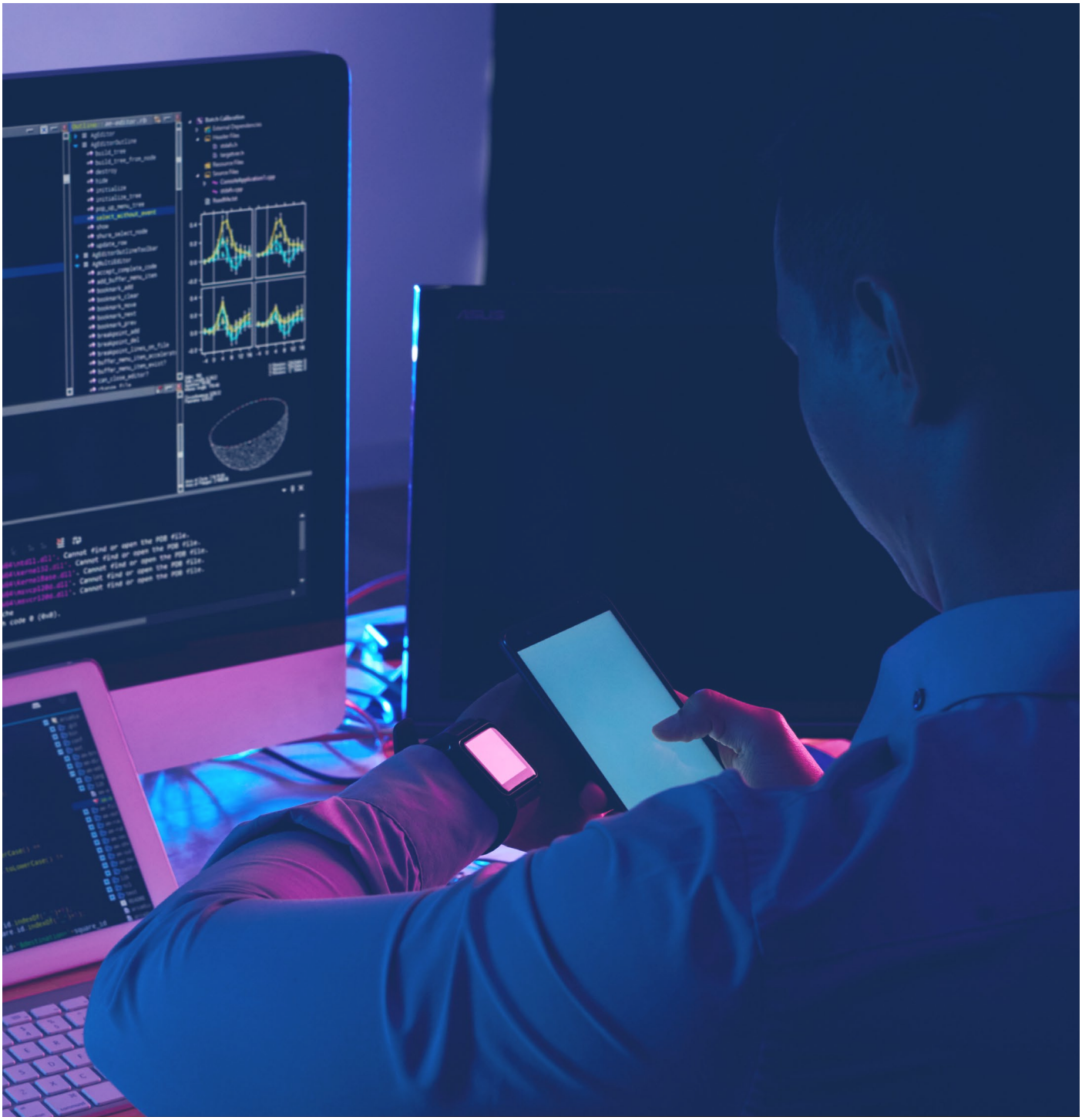

Build a Pragmatic, Future-ready Test Engineering Ecosystem

nagarro

Whitepaper: AQT

nagarro

Table of Contents

03 Executive Summary

04 Key Elements of a Future-ready QA Organization

04 Test Metrics Management

06 An Efficient Approach For Test Data Management

08 Flexible People and Governance Framework

10 Backbone Process Framework for Agile QA

11 Infrastructure and Tools

14 Test Value Chain

16 Conclusion

17 References

18 Author

nagarro

Executive Summary

According to an HBR survey among 250 business leaders, 66% of them agree that their company's future depends on the quality of their software. As per Nagarro's internal surveys, 78% of enterprises strongly believe that they need more efficient QA organizations to respond to disruption. Add to this, the fact that 25% of an application's development and operational costs can be saved through efficient operations. Quite evidently then, quality assurance is an essential part of the enterprise software lifecycle.

Despite such prominence to QA, there are still some major hurdles:

1. In many enterprises, QA is people-dependent. This leads to knowledge concentration in the hands of a few individuals. Knowledge transfer becomes another area that needs improvement when these individuals move on to take up other roles. Crucial information might get lost as such knowledge is shared more at an interpersonal level than at a process level.
2. Perhaps the biggest hurdle in modern QA teams is the lack of standards and transparency. To deliver high-quality software, a continuously monitored quality assurance system backed by a strong SOP is crucial. While this does exist in most enterprises, there is always room for improvement.
3. The number of bugs discovered by amateur hackers on popular enterprise software makes one wonder if QA teams are adding enough value. During the development lifecycle, software products spend too much time in the testing phase without enough tangible or measurable impact.
4. There is confusion on how to balance QA or Quality Assurance, which aims to prevent defects before they emerge, with QC or Quality Control, which focuses on identifying defects after they are produced.

The above points make one ask a few questions:

- Why does the end-product still have many defects despite spending a lot of money on testing?
- How can I always see the current proven progress and quality level?
- How can I expedite the testing and release cycles by integrating both the QA and development teams?

This whitepaper aims to answer the above questions while also providing the reader with a 360-degree overview of a future-ready QA team that is integral to every modern enterprise.

Key Elements of a Future-ready QA Organization

These elements, as illustrated in Figure 1, are permanent even as automation begins to gain more ground and QA tech stacks continue to evolve. However, the functions that come under them might undergo some modifications.

Whatever be its constituents, an effective QA strategy provides:

- A shift in the enterprise-wide QA mindset to provide synergistic services to all needs
- Scalable and flexible model skill management
- Enterprise Test Strategy covering functional and nonfunctional test requirements
- 360-degree knowledge management

Now, let us have a deeper look at each element of QA organization and explore the key elements, and the role that each plays. Let us start with Test Metrics Management.

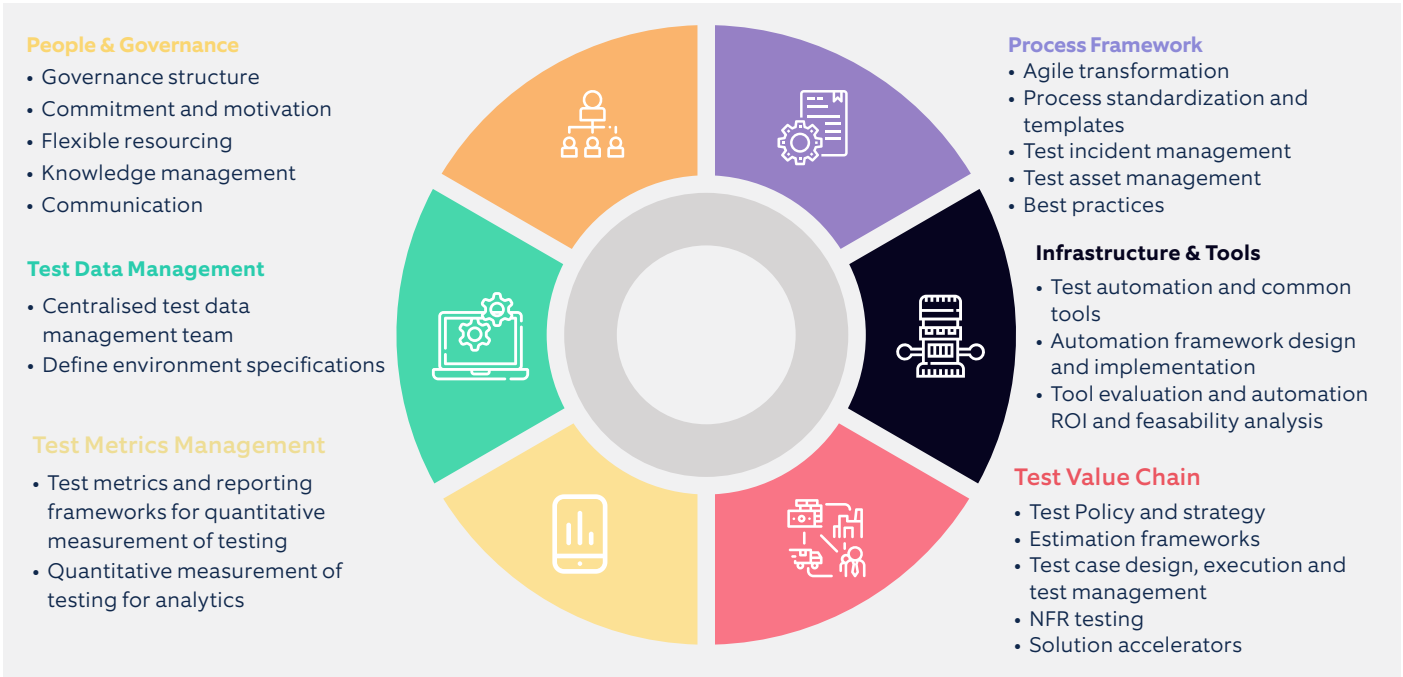

Figure 1: Six key elements to ensure a future-ready enterprise QA team

Test Metrics Management

Software Testing Metric is defined as a quantitative measure that helps to estimate the progress, quality, and health of a software testing effort. In other words, we are looking for a quantitative measurement of how well a testing process is doing.

If you can't measure it, you can't improve it

To ensure better management of test metrics, QA teams should segregate their reports into 3 categories as per their target audience: strategic, tactical, and operational.

This ensures that each report has a different goal and measures different KPIs. QA teams can then build a separate dashboard for each category.

1. Strategic

These reports are meant for decision-makers and C-level executives. They offer a bird's eye view of testing efforts. They also empower them to strategize for big goals like reduction in severity 1 and severity 2 incidents or to gauge the return on investment from automation. For example, Mean Time to Repair and Mean Time to Detect, system outages and downtime, cost of bug fixes pre/post release.

2. Operational

This is also a high-level reporting framework; however, they are used to track project-wise metrics. This dashboard is useful for PMs who want a clear overview of the project that they are running. It is also real-time, thus helping managers to act as soon as something goes wrong. For example, in defect removal efficiency, testing and defect trends.

3. Tactical

A tactical dashboard analyzes large volumes of data collected over time and allows you to explore the data through filtering so that you can detect patterns and opportunities. This is more in-depth than strategic or operational dashboards. For example, requirements and requirement coverage, defect distribution, defect open and close rate, test execution trends.

Choosing the right metrics, following them and improving them, is key to a successful software testing operation. A high-level idea behind having the following test metrics approach is to analyze, communicate, evaluate, and report.

Figure 2: Four steps to help you choose the right metrics

Now that we have seen the role of quantitative measurement of quality, let us go deeper in another critical aspect - Test Data Management. Test data can be automatically generated using tools (synthetic test data) and often, a masked and curtailed version of Test Data is created from production. What can be an effective approach for Test Data Management? Let us explore.

An Efficient Approach for Test Data Management

Efficient management of the data used for testing is essential in complementing the testing efforts. This enables higher levels of coverage and maximum return on investment. If the testing data does not advocate ease of use or represents the sampled sources poorly or consumes excessive resources for construction and maintenance, a negative impact on the outcome quickly demonstrates and continues to degrade the quality of results.

Test data management is the process of planning, analyzing, designing, building, and maintaining software quality-testing processes and methodologies. It allows the software quality and testing team to have control over the data, files, rules, and policies produced during the entire software testing life cycle .

Implementing a test data management approach involves steps that can help streamline the testing process by applying five best practices to test data management before going live, after testing is done:

Determine and study the test data

Organizations should identify their test data requirements based on end-to-end business processes and the associated data for testing.

Everyone knows it and testers have suffered from it: insufficient test data ruins automated testing.

Extract a subset of production data from multiple data sources

Collecting a subset includes obtaining metadata from the subset to accommodate data model changes. This creates realistic test databases small enough to support rapid test runs but large enough to accurately reflect the variety of production data.

Mask or de-identify sensitive data

Masking helps secure sensitive information and helps in ensuring compliance with industry and government regulations. De-identifying confidential data must provide a realistic look and feel and should mask complete business objects such as purchase orders across the system.

Automate actual and expected outcome comparisons

Identifying data anomalies and inconsistencies during testing is essential for measuring the overall quality of the application. Automating these comparisons help save time and identify problems.

Revise test data

Revising and refreshing test data helps improve testing efficiencies and streamline the testing process while maintaining a consistent, manageable test environment. To do this, there must be:

- Automated modular data creation and provisioning across multiple test environments
- Automated SOPs (Standard Operating Procedures) for test data creation (synthetic test data)
- Automatic data masking and copying of production data
- Integration of test data creation scripts with test automation framework and with the build and deployment process.

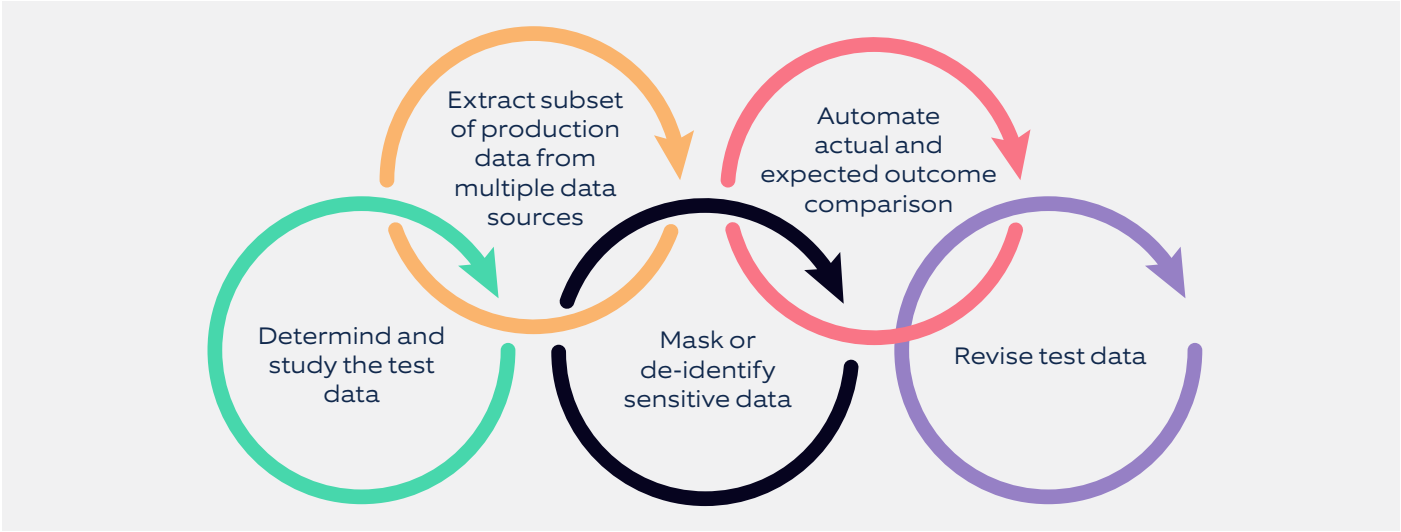

Figure 3: Test data management strategy

70-80% of test data requirements should be automated.

However, every continuous test data management strategy eventually faces certain challenges, such as:

- Diminishing access to a wider landscape of systems, including legacy systems such as mainframe and point of sale systems
- Creation and access to production-like data
- Prevention of test data depletion
- Identification of data anomalies
- Data requests poorly communicated, resulting in inadequate data returns
- Test priority confliction, and
- Timely data revisions.

70-80% of test data requirements should be automated.

By leveraging modern agile practices, QA teams can drive for the easy integration of TDM with the overall test management process and tools. This ensures a dramatic reduction in the time-to-market for a software without compromising on the effectiveness of the testing phase.

After understanding the significance of Test Data Management and Metrics, let us understand the role that people play to make a world-class QA organization. It is very important that people adapt, are flexible and believe in processes. Let us see how we can put the right framework in place to ensure that people enjoy the freedom that they need and at the same time, work efficiently.

Flexible People and Governance Framework

Typically, organizations running multiple software development activities handled by different teams, have a common and coherent governance framework to oversee testing activities in a broader context. This leads to several vestigial artifacts that may not be relevant to the project needs.

The solution then, is to shift the focus towards building an agile testing team. By their very nature, agile teams encourage collaboration and transparency. Flexibility to accommodate the requirements of an evolving project are built into the ethos of agile teams.

It's 2020 and in today's day and age, agile teams are commonplace. However, there are old enterprises and some newer startups too, that have only just begun their journey of agile transformation.

Agile Adoption Challenges

Enterprises still following the linear sequential Waterfall model, or the V-model will find that there is a disconnect between their development and testing teams. In turn, these teams are out of sync with the end-user requirements. A common complaint is that both the Waterfall and the V-model are rigid, meaning there is little flexibility and the scope to adjust is difficult and expensive.

Developers build a product and throw it over to the testing team. The QA team then writes elaborate testing scenarios to test the product. They also need to keep space for new testing scenarios to check for regressions in existing features.

As the product grows, this method becomes unsustainable and project managers are often forced to choose between not implementing new features or skipping on testing.

We all know the obvious answer.

Developing an organization-wide agile culture and mindset is the solution to these problems.

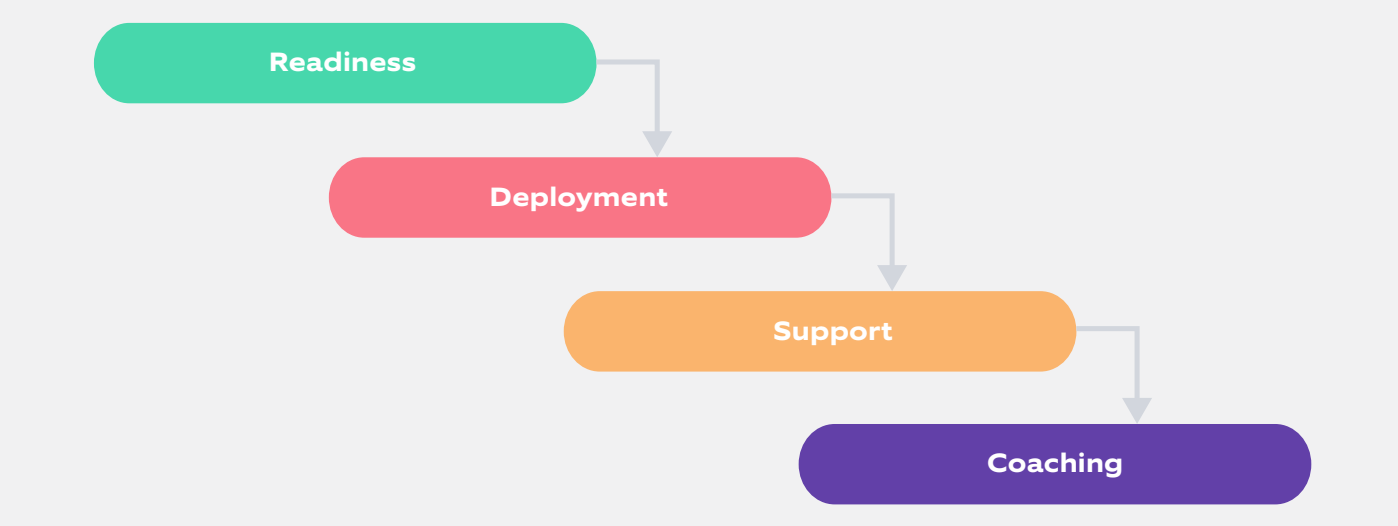

Figure 4: Agile adoption process

To actualize this shift, the transformation strategy team should:

Build a leadership team

Agile transformation requires changes to every part of the business, which requires support from the top management. Make sure executives are on board and know what is happening.

Define an end state vision

This includes a working premise for structure, governance, and metrics which we will progressively elaborate throughout the transformation process.

Build a transformation roadmap

We must give the organization some idea of what we're going to do, how long it's going to take, and what value we expect to harness from the investment.

Maintain a rolling plan

This plan will enumerate all the things in the organization that will be impacted over the transformation period.

Checkpoints

Periodically assessing the progress of the work of the transformation based on which one should retrospect and adjust the plan of action.

Adapt and learn

Re-assess the end-state vision based on how our understanding has evolved during the transformation.

Communicate

Regular, transparent communication about progress and impediments from the leadership will create excitement and energy.

The same transformation strategy takes different hues when seen from the management perspective.

Here, the focus should be on:

- Identifying risks and preparing a mitigation plan
- Ensuring the usage of agile planning and management tools even if it means having a dependency on a 3rd party
- Cultivating agile champions within each LOB who also acts as the Agile Coach
- Promoting a culture of active agile coaching and training, and
- Creating self-organizing teams that are cross-functional and multi-skilled.

The modern QA organizations are not strictly bind to processes. They are expected to be a smart mix of reactive, proactive, and adaptive maturities. We should have a backbone process run an ecosystem, but it should not be too rigid that it stops innovation and free thinking. Let's understand what a modern process framework would look like.

Backbone Process Framework for Agile QA

As mentioned earlier, enterprises are mainly flocking towards adopting agile methodologies for their flexibility. This is driven by the fact that flexibility is a key feature that decision-makers look for in any solution nowadays. Gone are the days when enterprises developed a one-size-fits-all package and their clients were forced to pick such solutions whose parts were completely irrelevant to what they had in mind.

This flexibility is a key reason why it is being increasingly adopted in software testing.

But is too much flexibility a bad thing?

Agile process standardization

QA teams should find the right balance between flexible test methods and process optimization. The focus should be on developing an effective test methodology by defining and publishing agile test process handbooks before the testing phase gets underway.

Agile estimation techniques

Right at the sprint level where user requirements are broken into tasks and estimated hours are assigned, the Planning Poker technique should be used. This consensus-based approach for estimating, a mainstay from Scrum, will help break down the requirements into granular tasks.

The next step is to ensure that all these tasks and the features developed therein are of actual importance to the business. Continuous focus on business value and alignment of the final product to business goals will help the QA team focus on ensuring that the 'customer delight' aspect is taken care of.

Requirements management

Given the tasks that are derived from user requirements, it is essential to prioritize their testing by adopting a parallel sprint and ensuring frequent capability releases to geographies/production divisions.

Risk Poker – a lightweight risk-based testing methodology in which risk analysis is performed through group discussion that outperforms the individual analyst's estimation – will be useful here.

Change management process

A well-defined change management process must be published and incorporated into the team guidelines. Changes raised through the process must then be accommodated within sprints.

After looking at the backbone process framework, let's see the importance of having the right infrastructure and tool set. It is important for a modern QA organization to create value continuously and hence usage of right tools is critical. Let's go deeper in this topic.

Infrastructure and Tools

When it comes to building and delivering better software faster, you can no longer choose between quality, speed, and security if you expect to remain aggressive. The classical test skills, such as test planning, test management, and product risk analysis are regarded as obsolete because they curtail development too much. At the same time, we observe that the new skills and technological solutions, such as test automation, data analytics, and AI technologies, are in high demand as they help in delivering higher business value and improved software quality faster. The new ways of working offer numerous advantages. New features are delivered faster to end users. Multi-disciplined teams share a joint objective to deliver customer-focused solutions.

A future-ready QA organization can only be developed by leveraging the best tools available and building upon credible, futuristic infrastructure. There is no doubt that the future is automation. But how will software testing be affected and what is the recommended approach to integrating automated testing?

End-to-End Automation

Before implementing automation, there must be a thorough audit of the existing testing infrastructure to ensure that it is feasible to be driven through automated processes. This also involves test tool selection, defining the scope of automation and the framework for automation.

Part of the audit process is also determining which all test cases to automate. To optimize your ROI, you should automate:

- High risk and business critical test cases
- Test cases that are repeatedly executed
- Test cases that are very tedious or difficult to perform manually, and
- Test cases that are time-consuming

At this point, one might wonder whether 100% automation is possible. While it is certainly possible to get very close to 100%, it might not always be ideal to go this way. Automated testing mechanisms could fail if the application is frequently updated or if the environment is unstable. It is also recommended to do UI verification and testing manually.

Selecting the Right tool

Of course, the toolset that you use for automated testing is highly dependent on the application under development, the language being used to develop it and other technical considerations. Testing tools like Selenium and Puppeteer have been around for a while and several companies have adopted them into their QA practice.

Whatever tool your QA team decides to adopt, it should have the following features:

- Reliable object identification capabilities
- Shareable object repository
- Reusable code modules to reduce test maintenance
- Cross-browser and cross-platform testing
- Customizable test report, and
- The ability to easily integrate with tools like Jira, Jenkis, TravisCI, Git etc.

100% Regression Automation

Whenever developers make a change to the codebase, the QA team needs to perform a regression test to ensure that the new functionality did not break the software. Such changes are very common in the SDLC and manually performed regression testing is not feasible. The solution is to implement an automated regression testing suite. The only manual work needed here is an occasional upkeep of the suite.

QA teams need to adopt automated progressive regression methodology to ensure that incremental changes do not impact the earlier work by:

- Automating and executing from the beginning for regression within Sprints
- Automating regression for features built in Sprint N in Sprint N+1, or
- Automated smoke testing and functional/regression testing within Sprint N.

Connected Testing Ecosystem

The apparent current trends in QA are a progressively business-driven approach, an increase in demand for end-to-end testing, and a surge of interest in AI. Creating an integrated, perceptive, and comprehensive approach to testing then becomes critical. A bracketed testing ecosystem can provide continuous monitoring and delivery of system advancements, right across the product lifecycle. It can also feature perceptive and adaptive test scripts, responding to real-world use patterns.

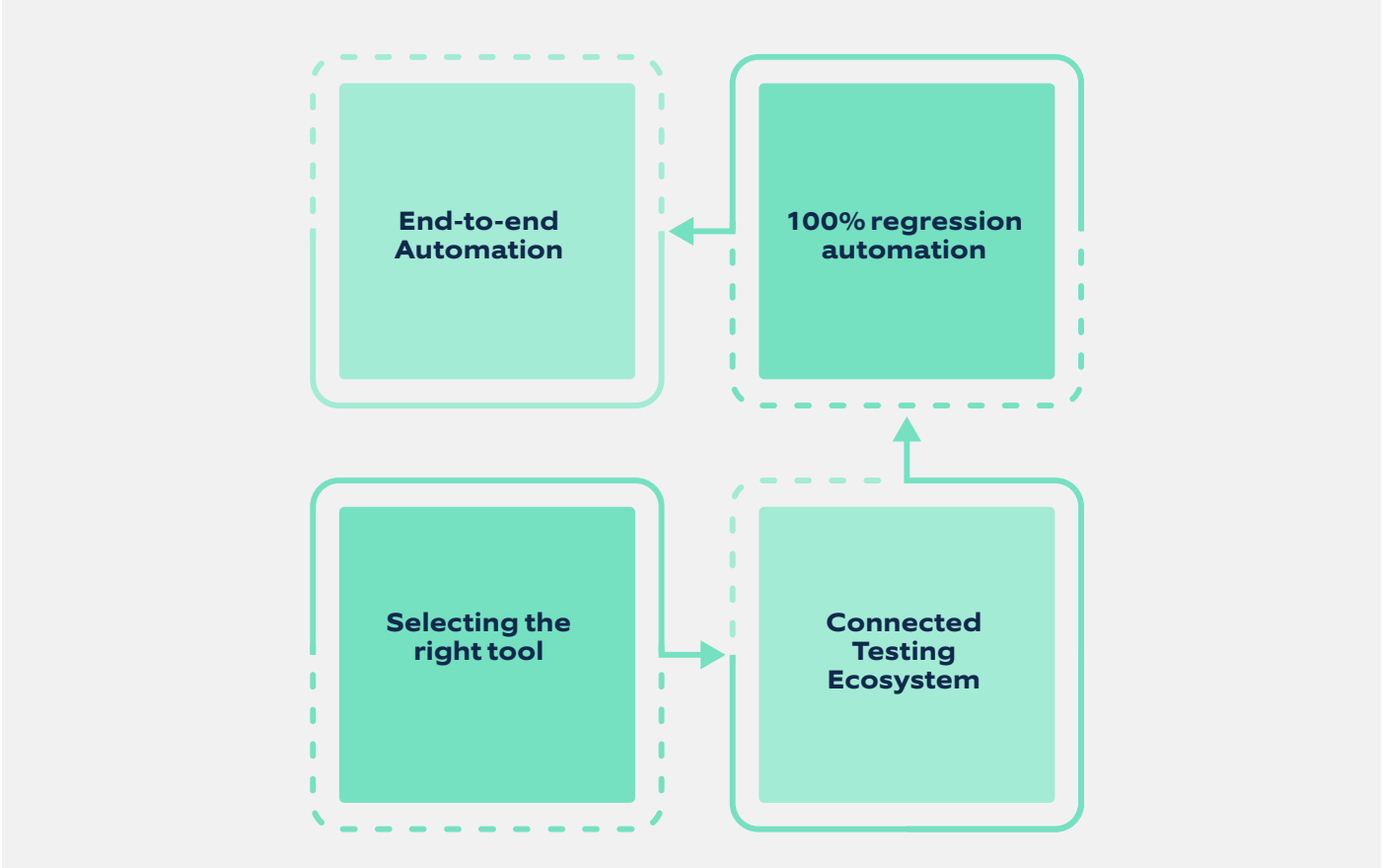

Figure 5: Infrastructure and Tools

Build a Team Around Emerging QA Skill Set

Core domain skills should be integrated with new strengths in automation, in the test environment, and test data. Growth of Artificial Intelligence in business is creating requirements: first, the need to tame smart developments for business advantage; and second, the need to use AI specifically within testing and QA. Skills in data science and mathematics, and in the understanding of cognitive processes are needed to achieve this.

AI-driven technologies such as chatbots have a compelling role to play. Chatbots, for example, have taken the human-computer interaction and experience to a different level. Since, organizations these days rely heavily on automation, building a team around chatbot development and testing helps in achieving the same. Training and fine tuning chatbots so that they behave like humans is a daunting task. For this, the QA team needs to consider the full set of potential scenarios so that the chatbot is well-trained.

Today, there is an increasing need to deliver superior and rapid services. There is an enormous demand to access, create, use, and share data from any device. The urge is to provide meaningful insights and control over various interconnected IoT devices. Securing the data over the internet and its privacy is very critical; authorizations play a very important role in IoT data streaming and transfer.

Considering the security challenges involved in IOT testing, QA professionals must assess a range of vulnerabilities in IoT.

Blockchain technology is picking up pace in various industries, domains and segments. Securing, storing and managing data and digital identity, organizations tend to look out for platforms which are secure as their data is being shared across the supply chain. Blockchain testing comes with its own set of challenges, which compels companies to have a core QA team for handling these challenges.

Test Automation as a Platform

Organizations should regard Test Automation as less of a capability, and more as a platform. It should be a broad arena shared by tools and functions that come together to fulfil a collective purpose – and driven by the objective of the business.

Continuous value creation is key for the success of QA teams. The testing must move up in the value creation cycle to support accelerated deliveries. Let's see how testing could be moved up in value chain to continuously minimize the business risk accumulation.

Test Value Chain

An increased adoption of Agile and DevOps approaches has resulted in QA to become more fundamental to development processes, thereby shrinking time-to-market and improving cost-effectiveness. This integration enables organizations to locate and address defects at initial stages, and implement updates sooner, thus helping with prioritization and increased throughput volume.

Smart automation helps in finding and fixing issues quickly. It also helps combined test and development teams decide which changes will deliver the best and fastest returns. Competitive demands and digital transformation increase the need for more apps and new functionalities in a shorter duration of time. As a result, it's highly likely that testing and QA will be an integral part of the end-to-end development process.

The QA process and teams should be aligned in such a way that helps in fulfilling the following purpose:

Business case perspective

Understanding the business case helps in understanding the business perspective of the application, thus helping in identifying which defects are critical, and designing the most suitable testing methodologies.

Collaborative testing

Involving various stakeholders at early stages and communicating with them makes it easy to understand the business case. It also helps in making quick and correct decisions about the functionalities and features of the application.

Lean approach

Focus on faster go-to-market by eliminating inefficiencies and delays in the system.

Early validation

Apart from the Left shift that we talked about a while ago, this ensures that business requirements are fulfilled at the system level, before integration of the whole. This reduces the cost of testing as defects are identified at an early stage.

Customer centricity

Shifting your way of thinking and continuously measuring the customer experience identifies gaps between what the customer thinks is an excellent experience and what you think is excellent. Ensuring that the engineering team is collectively delivering high-quality applications that fulfill customer needs and broadly identifying performance, trends and issues faced by customers are some of the primary objectives of customer eccentricity.

The non-functional aspects of a software (performance, reliability, UX, etc.) can and should be done by human testers.

In an agile project, automated testing takes on the brunt of the testing responsibilities. This includes testing new user stories and performing regression testing.

From a Project Manager's perspective, to extract maximum value, they need to ensure that the QA organization needs to be cost-effective while also:

- Prioritizing tasks accurately
- Regularly analyzing the QA team productivity and continuously optimizing their processes, and
- Strongly suggesting and implementing an agile cooperation model throughout the SDLC.

The QA team should also be strongly use-case oriented and understand the purpose of the software, its main goals, and the problems that it should solve.

Conclusion

There is no magic wand for becoming a world-class quality organization, neither does it occurs in isolation. There is no single path to take toward organizational culture and improvement.

The age of automated testing is here, and the onus is on key decision-makers and stakeholders to take the bull by the horns. However, this does not mean that traditional testing roles have become redundant. They still have their uses and PMs need to take a call on what strategy is ideal on a case-by-case basis.

But QA teams across enterprises need to adopt the latest in agile if they are to stay ahead of the testing curve. The currency in testing circles has changed from "number of test cases" to the "risk coverage" that the testing process achieves.

This redefined KPI, combined with superior speed-to-market powered by test automation, is essential to support the service virtualization seen in the IT sector.

At Nagarro, we have defined a roadmap for the state-of-the-art test organization:

- Identify the details of the current situation
- Define the areas of improvement in QA practices
- Determine a detailed roadmap to create world-class QA practice
- Recommend a test organization set-up, practices, tools, technology, restructuring, KPIs, templates, end-to-end automated solution, and
- Leverage industry best practice accelerators and our Innovation Labs tools to reduce costs and improve productivity.

This modern 360-degree model combined with our industry-standard TPI® methodology provides step-by-step guidance and recommendation roadmap to achieve the adaptive stage in test practices.

nagarro

References

1. https://www.techopedia.com/definition/30193/test-data-management

2. https://www.researchgate.net/publication/329356928_A_risk_poker_based_testing_model_for_scrum

nagarro

About the Author

Khimanand Upreti
He has more than 16 years of experience in software testing and holds a university degree in Computer Science. Since 2011, Khimanand has been working for the software house Nagarro and together with his Austrian colleague, Hannes Färberböck, heads the global business unit Accelerated Quality and Test Engineering (AQT) and Center of Quality Excellence (CQE). His passion in testing is focused on new digitalization topics, such as testing of artificial intelligence, machine learning or quality assurance of chatbots, and blockchain.

Anuj Verma
He is an experienced Quality Analyst with a demonstrated history of working in the information technology and services industry. Having an experience of more than 8 years, he is skilled in Automation Testing, Test Planning and Release Management. He has a keen interest in Consulting, Process Management, Agile Methodologies and Project Management. Anuj holds an MBA focused on Information Management and Strategic Management.

Marketing team

Shailesh Dhaundiyal
Editor

Lisa Pradhan
Content Marketer

Nitish Singh
Designer

About Nagarro
Nagarro is a global digital engineering leader with a full-service offering. Nagarro specializes in "change the business" technology services, including digital product engineering, e-commerce and customer experience services, Artificial Intelligence and Machine Learning capabilities, cloud and IoT solutions, and consulting on next-generation ERP. The company has a broad and long-standing international customer base, primarily in Central Europe, particularly Germany, and in North America. This includes a number of global blue-chip companies and leading independent software vendors (ISVs), other market and industry leaders, and public sector clients. In total, the group employs over 8,400 people worldwide (as of June 30, 2020).

Further information is available at www.nagarro.com.