Shaping the Future of Quality

The five game-changers for your business

nagarro

nagarro

Table of Contents

Executive Summary

An HBR survey revealed that a resounding 66% of 250 business leaders agreed that their company's future depends on the quality of their software[1]. According to Gartner, more than 90% of executives face increased pressure to release products faster as more than 50% of companies compete using advanced analytics algorithms[2]. You might also know that 25% of an application's development and operations costs can be saved through efficient operations. All these facts point in only one direction: Since quality assurance (QA) is a vital cog in the enterprise software lifecycle, it must always be efficient.

But despite all these facts, QA still faces some significant hurdles:

- Most organizations view test automation as too traditional, without looking into the aspects of intelligent and advanced test automation.

- Perhaps the biggest hurdle in modern QA teams is that they are not lean. To successfully deliver high-quality software, it is crucial to have a continuously monitored QA system, backed by a robust and adaptive process. Although this does exist in most enterprises, there is still a lot of room for improvement.

- While the speed of delivery is vital in the modern world, it will always need quality to be effective. Most QA organizations do not implement Shift-left, continuous integration, and deployment correctly, causing instability in production.

These challenges raise some important questions:

1. How can you implement advanced and intelligent test automation?

2. How do you make QA organizations "lean"?

3. How to increase efficiency and optimize the cost of quality engineering for systems with dramatically growing complexity without impacting product quality?

4. How to expedite the testing and release cycles by integrating both the QA and development teams?

5. How can you efficiently implement DevOps and Shift-left so that only minimal manual testing is required towards the end of the product life cycle?

6. How do you remove everything that is not adding value to the testing life cycle?

So here's the tricky part: How can we ensure QA can overcome these challenges, and what does the future of testing look like?

This whitepaper will answer these questions and provide you with a comprehensive overview of a future-ready QA team that is integral to every modern enterprise.

Five Key Elements which are shaping the Future of Quality

In today's technology-driven world, many enterprises are catching on to the wave of disruption. There are many drivers of this disruption. They could be about the changing technological environment, evolving business dynamics, increasingly intense competition, and suchlike. Disruption is a widespread concern for enterprises, as the technological competition is on the verge of disrupting business models. Polaroid Corporation is a classic example: A pioneer in digital imaging in the 1960s, the company did not make the necessary investments to maintain its dominance in the 1990s, when digital photography overtook film. On the same lines, we can see how printed newspapers, magazines, and letters stand increasingly marginalized in today's digitized world.

Although no one can predict the exact impact and extent of disruption in the next five years, a recent study found that it is vital for enterprises to continuously adapt to new ways of working. The world of quality engineering is no exception to this golden rule and is continuously evolving with new tools, approaches, and methods. Based on Gartner[3] and our point of view, here are the key elements that can ensure that your quality engineering is future-proof and can be well-prepared to face any disruption.

Hyper-automation	Total User Experience	Agile & Lean	AI & Big Data	IoT
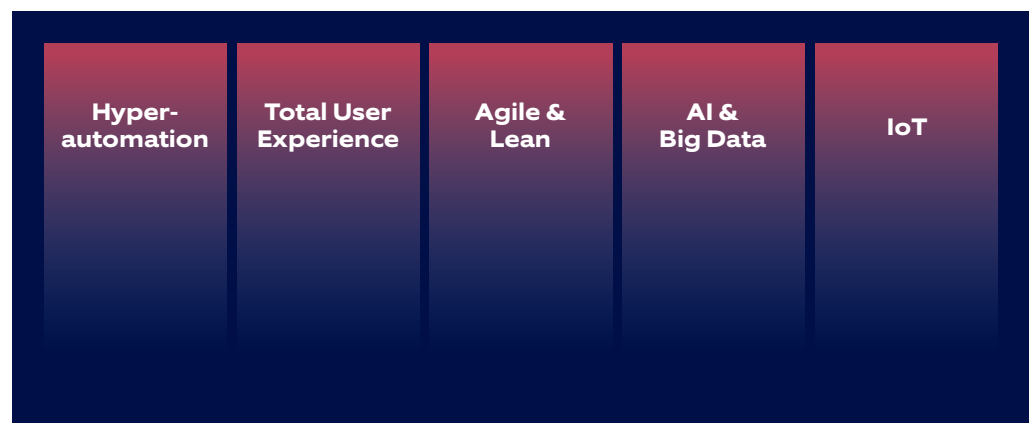
Five key elements, shaping the future of quality

We will show how each of these elements is central for the future of QA, how it will develop and what effects it has on product quality, costs and speed.

1. Hyperautomation

Create business value faster with quality and minimal risk accumulation. Reduce and balance efforts and costs in a world of dramatically growing complexity.

Automate the automation

In software, 'Automate the automation' refers to automating tasks using tools and technologies. This typically involves automating repetitive, complex processes, high-volume data processing, or routine time-sensitive tasks without human intervention. With time-to-market becoming a differentiator in software development, we need automation to achieve further acceleration and meet quality standards. The Automation team and test engineers will play a new role in tuning and monitoring the test results. Enterprise-level teams re-examine the existing lifecycle and identify gaps and potential areas for automation in the end-to-end lifecycle.

Teams do not just concentrate on automating UI/API test cases to achieve hyper-automation. Their main goal is to increase efficiency by automating operations.

Businesses are now transforming into intelligent, data-driven organizations, relying heavily on artificial intelligence (AI) and digital technologies to optimize operations and change business models. New-age technologies can manage end-to-end digital operations, enhance faster delivery of projects, and reduce time constraints.

Building an automated knowledge management repository to give users accurate information via chatbots helps offer round-the-clock customer support. No wonder, ChatGPT gained such massive traction, and as you can expect, this is just the beginning of many such tools. With AI/ML algorithms, chatbots constantly learn and automate different scenarios to improve customer services and reduce costs. **Robotic Process Automation** (RPA) mimics the actions of a human user. It helps automate pre-defined tasks like workflow management (routing tasks, invoicing to correct departments) or decision-making on pre-defined rules. RPA can also be integrated with other technologies, such as AI/ML, to make the process more intelligent and adaptable.

Teams can use **Natural Language Processing** (NLP) techniques to automate tasks such as text summarization, language translation, test case generation, test case execution, and test case optimization. They need to identify and prioritize the most important test cases to be executed, to save time, effort, and cost. Machine learning (ML) algorithms are used to analyze test results and identify patterns, trends, and outliers in data, which can be used to improve the testing process.

Automating the automation process allows us to achieve continuous testing with more accurate results. By using AI algorithms and techniques, businesses can enhance their product offerings, improve the customer experience, and increase the quality of their services.

What should we do? In today's ever-changing landscape, our focus should be to become an end-to-end assurance partner with a deep understanding of business processes and outcomes. We must leverage AI algorithms and techniques for self-healing test automation and automate tasks on reporting and analysis. At Nagarro, we have started a research program, that aims to "Automate the automation", where our automation architects have developed many cool AI –based utilities such as self-healing, code risk indicator, automated log file analysis, etc. **Click here** to read more about it.

Our visionary goal is to look beyond automated test execution and work on automated maintenance of the framework through self-healing, automated log file analysis, automated test design, automated code refactoring, and much more. The diagram below reflects some key areaswhere Nagarro research team is working on the topics of Artificial Intelligence for Testing.
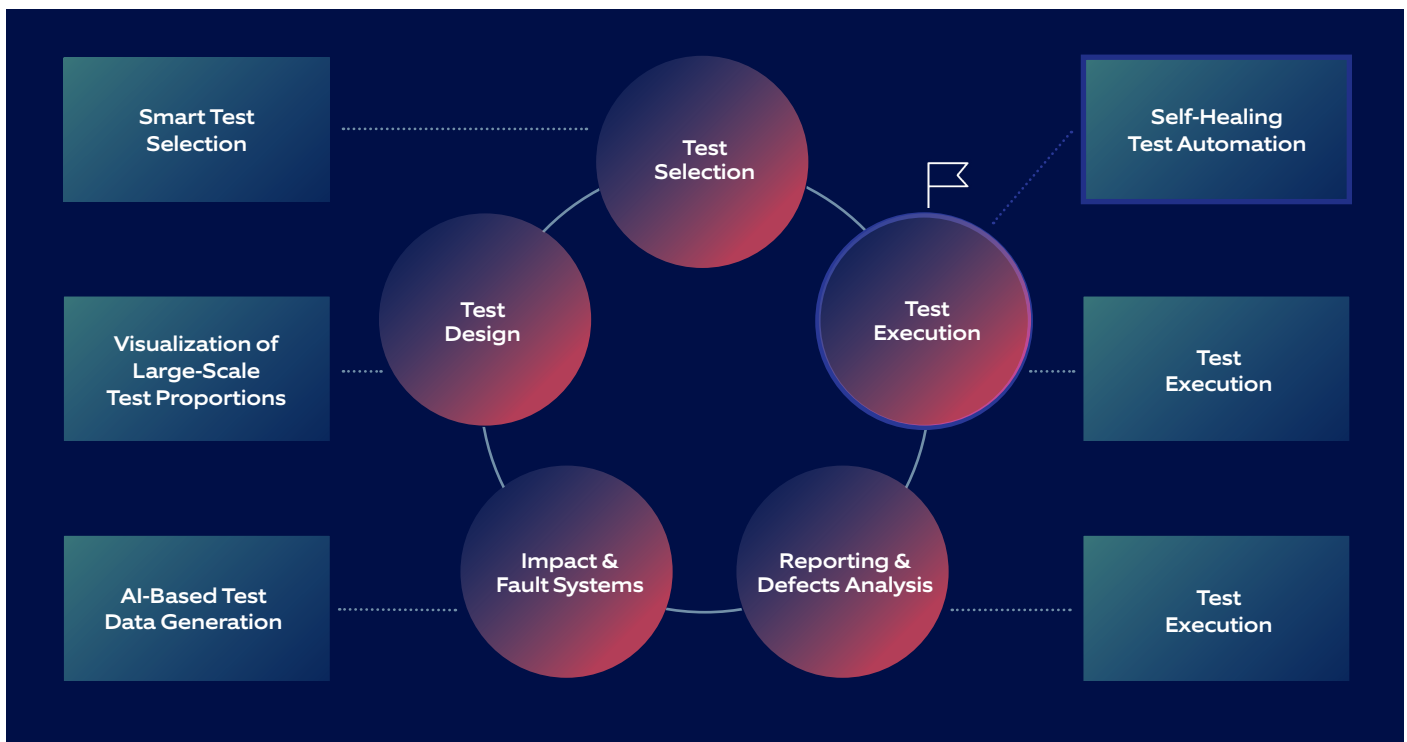
Key elements of intelligent test automation at Nagarro

Intelligent automation & self-healing

Traditional automation cannot identify and resolve issues independently, leading to costly downtime and vast maintenance efforts. Intelligent automation and self-healing leverages AI and ML techniques to address these problems by allowing automation to adapt to changes in the system. With the ability to detect and diagnose problems, Intelligent automation can take corrective action to resolve the issue without manual intervention. This not only makes the automation more resilient, but also leads to more efficient and sustainable automation processes. Here is a high-level snapshot of how self-healing works.
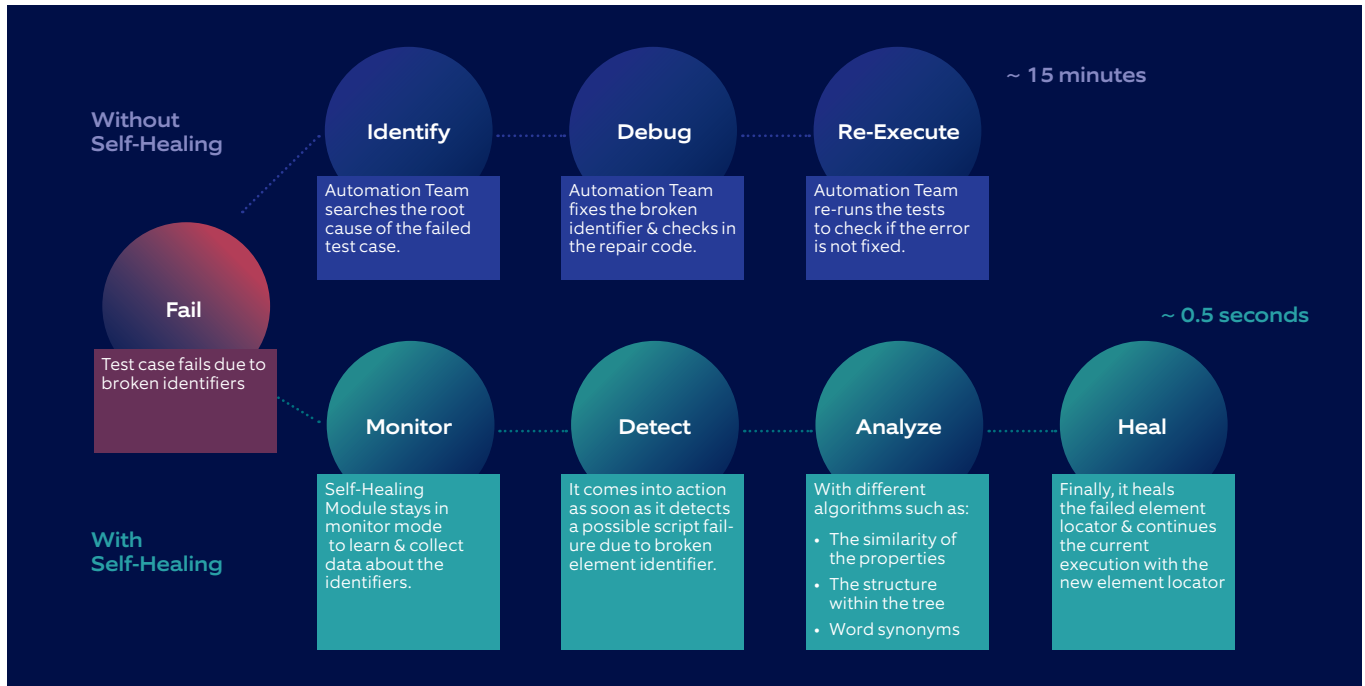
Algorithmic flow of how a self-healing algorithm works

Sustainable & tool-agnostic automation

Another way to make automation solutions more sustainable is by designing them as tool-agnostic. Tying automation solutions to specific tools and platforms makes it difficult and expensive to maintain and update them over time. **Click here** to know more about Nagarro's test automation approaches and how we make automation more useful for you.

Sustainable and tool-agnostic automation approaches prioritize flexibility and versatility to ensure that automation efforts remain relevant and effective over time, even as technology evolves.

Advanced visualization

Advanced visualization is crucial in test automation, providing a comprehensive and intuitive representation of the testing results. It allows teams to quickly identify areas of improvement, track progress, and make informed decisions that drive continuous testing and quality assurance efforts forward. With advanced visualization, testers can gain a deeper understanding of the impact of their work and make data-driven decisions that enhance the speed and efficiency of the testing process. By visualizing test data in real time, teams can effectively communicate complex information and streamline collaboration, leading to better results and improved overall testing outcomes.

Automated model creation

Why do we see automated model creation as a key aspect of future test automation? That's because it enables organizations to create, efficiently maintain, and evolve models that accurately reflect their systems. By automating model creation, teams can save time and increase consistency and accuracy, ensuring that their testing models always stay current. Organizations can quickly generate customized models with automated model creation to match their specific testing requirements. Ultimately, automated model creation empowers organizations to streamline their testing processes and achieve more effective and efficient testing outcomes.

Automated explorative testing

Automated explorative testing is emerging as a game-changer in software testing. By leveraging advanced algorithms and AI, this testing approach automates the creation of test cases and the execution of tests efficiently. This form of testing saves time and resources, as compared to manual test scripts. What's more, since it explores many more testing scenarios, this form of testing also provides a more thorough evaluation of the software. As software development continues to evolve, automated explorative testing will play a vital role in helping organizations keep pace with the ever-increasing demand for high-quality software applications.

DevTestOps for automated continuous deployments and releases

Teams have moved to continuous testing to ensure testing at the pace of frequent product launches. Continuous testing is accomplished by utilizing strategies like Shift-left, Shift-right, anytime deployment, automation, continuous deployment, and release management process. Hyperautomation elevates this automation to the next level by automating as many business and IT processes as possible. **Click here** to know more about how Nagarro is shaping the topic of Quality DevOps.

Organizations that implement hyperautomation, will benefit from Shift-left and Shift-right testing. The Shift-left technique promotes testing in the software development cycle as early as possible. This allows organizations to discover problems early enough and prevent them from progressing to any subsequent phase of the software development life cycle.

On the other hand, the Shift-right strategy involves testing in production to validate that the system behaves as expected in a real-world environment. AI bots can be built which understand production deployment failures & help to build and execute dependencies. With shorter implementation and release cycles, testing, development, and production teams do come closer together with Shift-right testing.

Anytime Deployment is achieved through continuous integration (CI) & continuous deployment (CD) practices and tools. It removes the need to wait for a specific release and allows teams to deploy changes as soon as they are available, while giving them the safety to roll-back modifications quickly and easily. Depending on the requirement, different deployment strategies (including Blue-Green Deployment, Canary Deployment, Rolling Deployment, and A/B Testing Deployment) are used to reduce downtime and ensure that new code changes are extensively tested before they are released to the entire user base. Containerization and Infrastructure-as-Code (IaC) have made easier for teams to manage their infrastructure and any dependencies automatically. This not only makes the delivery of new features fast and reliable but also delivers crucial security updates and patches, ensuring the system remains up-to-date and protected.

Chaos Engineering & Resiliency Testing is gaining popularity because of how it enhances system resilience by purposefully inflicting failures. This leads to reduced downtimes, with improved overall system performance and creates systems that are equipped to handle most failures or errors.

2. Total User Experience

Total customer experience is critical for the success of any business. A great total experience increases customer happiness and satisfaction, resulting in enhanced customer loyalty. These qualities are essential for building a great brand in the long run. Engineering approaches must become more customer-centric to attain their goal of a great total user experience.

Have you ever been in situations when the customer is unhappy with the quality even after you have made extensive efforts to ensure just that? Or the system under test might be working well functionally but the customer might still be away from feeling that "wow" effect, leading to just an "average" total user experience. Why does this happen?

It is also pertinent to note that with entrepreneurship currently at an all-time high, many new businesses are coming up across multiple domains. This means that to thrive amid all the competition, it is imperative for any business to be available online 24x7, and without compromising on quality either. Let's quickly see how testers and customers perceive quality.

According to a study, 88% of users will not return to a website with a bad user experience, regardless of how correct the functionality was. Given almost 80% percent of internet users are willing to pay extra for an awesome experience (as per UX Planet), it is imperative to make user experience a key element of any business strategy. Additionally, in the world of omnichannel platforms, 83% of consumers think that it is necessary to have a seamless total user experience across all channels. This makes it vital for any design thinking and quality engineering strategy to think about the total user experience. And now, let's see how a typical QA engineer would look at quality.

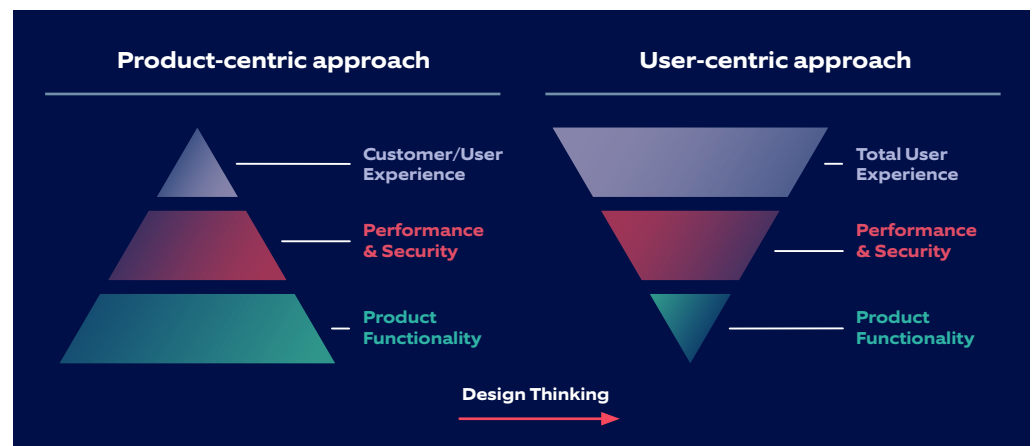## Product-centric approach vs. user-centric approach

Most quality engineers perceive quality as requirement-compliant product functionality and follow a "**product- / functionality-centric approach**". In this approach, the top focus is on ensuring correct product functionality, with medium focus on performance & security. Unfortunately, ensuring a delightful total user experience comes last in the pecking order. This approach generally ensures the correct product functionality but does not result in a delightful user experience for the customers. To find out why, we must first understand the customer's perception of quality.

Product-centric approach of quality

- Customer/User Experience
- Performance & Security
- Product Functionality
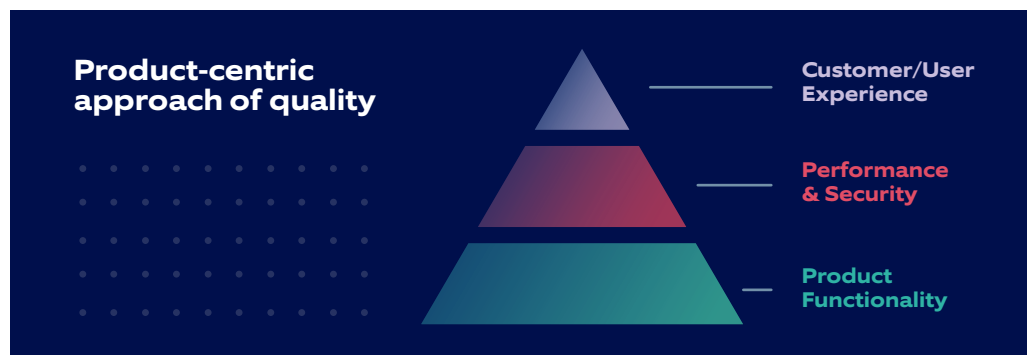
It's about "end-user experience," not about "end user experience"!
Most organizations view test automation as too traditional, without looking into the aspects of intelligent and advanced test automation. Always remember that the customer's perception of quality is driven by their end-user experience. That's right, the "end" in "end-user experience" is not a verb! **The total user experience is a perception, and the best quality engineering approach focuses on the end-user and manages their perception.**

The customer's perception of quality will always be more user-centric than functionality-/ product-centric. Hence, it is very important to understand our customers' total user experience' needs and define a quality engineering approach aligned with this need. If our quality engineering approach is product-centric and ignores end-user experience needs, it will only lead to unhappy end-users and hence, unhappy customers. **So this is why, our quality engineering approach should be user-centric and notinstead of product-centric, as shown below.**

The user-centric approach of quality places its highest focus on end-user experience. A user-centric approach does not mean we deliver a functionally buggy system to the customer. This simply means the system must be tested by keeping the end-user experience in mind. Instead of product functionality, the user experience drives testing and test design and has the highest priority in the overall quality engineering thought process. You can read more about it in this article **here**, written by a Nagarrian. As per a study done by Google, currently 53% of mobile users leave websites after 3 seconds, and 90% of users stopped using an app due to poor performance[4]. This is a strong indication of how important user experience can be for any product.

So, what approach should be taken for an amazing user experience? Honestly, there is no single answer to this question. **Click here** to learn more about what Nagarro's head of Design Thinking has to say. Careful user design thinking, user personas, the performance of websites and apps, interactive layouts, and the usage of pleasant fonts and colors – all these improvements can enhance the total user experience. Total experience may include how easy and intuitive the app is to navigate and use, but it is not limited to just that. A very intuitive and well-designed app can also fail if it doesn't meet the expected performance. Pages and navigations taking ages to load, repeated timeouts, latency in server responses, and the app taking longer to render on the device – such issues (even on an otherwise well-designed app) make the experience poor and as unpleasant as with a poorly designed app. Not just server-side performance but also how optimized is the usage of the local device resources like memory, CPU, battery, network, etc. - all such factors play a very important role in deciding the overall performance and acceptance of the app.

At Nagarro, since we understand the necessity of performance testing, as you can see **here**, we ensure that device-side performance and profiling are given their due importance.This brings the app profiling tools into the game. These tools help identify and resolve performance issues, improve the overall performance and total user experience, and deliver higher-quality apps to users. Xcode Instruments and Android Studio Profiler come out of the box with the IDEs. Apptim, Apteligent, and Systrace are a few other popular players in the market, which help with client-side performance testing needs.

Three key elements behind a great Total User Experience

To create a great Total User Experience, businesses must focus on three main elements:

- **User-centered design:** User-centred design puts the needs and goals of the user at the forefront of the design process. It involves understanding the target audience and their motivations, challenges, and expectations to create products and services that meet their needs.

- **Consistent and cohesive brand experience:** A consistent and cohesive brand experience across all touchpoints is essential for creating a memorable Total User Experience. This means ensuring that the visual and functional elements of the brand, such as logos, colors, messaging, and overall look and feel, are consistent across all interactions with the brand.

- **Seamless and efficient processes:** A Total User Experience should be seamless and efficient, with no friction or obstacles that prevent the user from accomplishing their goals. This includes the overall process of purchasing a product or service and the experience of using it.

Clearly, a major pillar in shaping the future of QA will be user-centered testing, where we put ourselves in the customer's shoes. This not only helps us deliver better quality software, but also takes us up the value chain for our customers, by bringing us closer to our customer's customers. An enhanced total user experience can lead to better sales, better return on investment, and higher end-user satisfaction, creating a memorable experience for the user.

3. Being Agile and Lean

To meet the drastically increasing expectations put into Quality Engineering, organizations need to adopt a mindset of continuously evolving their tailored way of working. Ultimately achieving a future-proof quality engineering ecosystem that constantly adapts and improves.

Future-proof Quality Engineering requires a tailored and integrated approach that priorities continuous improvement and learning over off-the-shelf solutions. A transition to a service-based perspective with supporting reliable, parallel development requires organizations to adopt a holistic mindset with a focus on automation. Unfortunately, due to the complex nature of these problems, no "one solution" will ever be available for adoption. Organizations must continuously explore different methodologies and adapt to specific challenges to achieve a quality engineering ecosystem that will also be relevant for the future.

Success in Quality Engineering requires a fully integrated DevOps mindset that drives organizations to work on culture improvements, with automation providing actual added-value, transparency, and measurable empirical process controls. It should also embrace learning and failure as input for continuous improvement. Essentially, this is a mindset that is based on DevOps thinking and principles. If you are interested in the current state of DevOps adoption, please **click here for the 2022 Nagarro State of DevOps Report**.
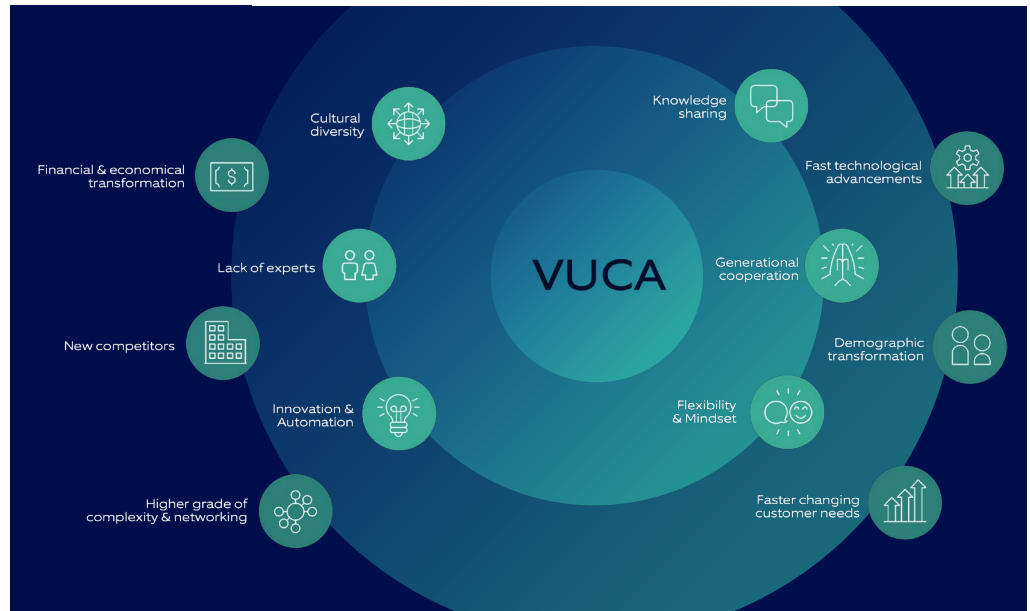
The volatility, uncertainty, complexity, and ambiguity of the modern world makes searching for the one true answer or solution a trap. Instead, one needs to embrace complexity and evolve a tailored way of working.

Evolve a tailored way of working

To meet the ever-growing user demands for reliable products, organizations need to embrace a culture of instability and learn from failures, right from any feature's ideation up to its delivery and beyond.

While the ways of working may vary across different organizations and products, here are four guidelines that are universally applicable:

1. **Maximized Shift-Left Engineering** focuses on incorporating quality, right from the early stages of software development to foster collaboration and improved quality. Organizations can implement models like Behaviour Driven Development (BDD) and Domain Driven Design (DDD), which focus on end-user behavior and the business domain. Achieving Shift-Left requires a behavior change and may require overhauling the development process and employee upskilling.

2. **Introduce a Continuous Quality Engineering System** to maintain high quality throughout software development. This includes tools, processes, Continuous Deployment pipelines, static code analysis, live data monitoring, automation, and a user-centric, test-first approach through BDD. Integration requires investment in a modern tooling platform and upskilling the team.

3. **Make instability a habit** by embracing chaos engineering and a culture of failure. Focus on identifying weaknesses by randomly disabling services and developing self-healing approaches. This leads to a resilient team of engineers who can quickly handle outages, without impacting production or end-users, resulting in increased system reliability.

4. **Continuously experiment and learn** to stay ahead in quality engineering. Incorporate systems thinking to constantly adapt to changes and improve your way of working. Align your team with shared goals and user-centered design using hypothesis-driven development and testable evolutionary architecture. Embrace new ways of working, team structures, and technology.

Quality Engineering is evolving towards Lean Continuous Quality, which incorporates quality into all stages of software delivery and prioritizes user experience and reliability. To meet the demands of a rapidly changing business dynamic, organizations must adopt a holistic and integrated mindset, based on the principles of DevOps, and should be ever-willing to continuously experiment, learn, and evolve a tailored way of working. **Click here** to see how Nagarro supported an organization to adopt this way of working, along with the factors that ensured this successful transformation.

Ultimately, the goal is to achieve a future-proof quality engineering ecosystem that constantly adapts and improves. This must not be considered as just a one-time effort but as a continuous process. You can achieve success only when culture, processes, and technology converge.

4. AI & Big Data

While adopting AI-driven testing can work as a catalyst for day-to-day QA activities, there is also another facet to it. It is also worth noting that disruption of AI-based applications also presents a challenge for QA teams on how effectively they can test AI-based applications.

With AI changing every aspect of digital work, QA needs to focus on not only faster ways of testing but also more innovative and AI-driven testing approaches. If you want to know how things might evolve in the next decade, do check out our white paper called **AI Testing AI** A new class of workers has emerged in the form of "augmented workers," - this is your AI code, embedded in any tool or framework. Ideas like automatic script generation, auto bug logging, intelligent log analyzers, predicting bugs in the application, etc. are rolling out thick and fast.

While AI disruption provides many such new avenues and opportunities in testing, it also presents a challenge for QA on how effectively they can test AI-based applications.

Artificial Intelligence imparts cognitive ability. Cognitive processes are the natural intelligence of human beings. For example, we see a picture and make sense of it. AI is the computational model of these cognitive processes. By implementing these AI models, we try to solve real-world problems. But a computational model is not the same as a cognitive process. It is the simulation of the cognitive process and an approximation of these cognitive processes.

In a traditional non-AI-based application, based on the required behavior, the developer defines some rules in the code. Since these defined rules are Boolean, they are deterministic. The pass criterion for testing is very simple: If the given rule passes the testing, it is passed, and if the rule fails, the testing is failed.

Our test strategies revolve around these two risks:

1. Gaps in the common understanding in the form of missing, incomplete, and misinterpreted requirements.

2. The incorrect implementation of the rules in the form of (errors of omission, errors of implementation, conflicting rules etc.)

On the contrary, an AI-based application is based on the required behavior, where the developer selects a learning algorithm that fits with the client's requirement to solve the problem. This algorithm is nothing but a complex mathematical equation. The learning algorithm defines the system of rules. These rules are based on probabilistic logic.

Here, the pass criteria are not simply black or white. In most cases, it is the accuracy level or percentage. For some applications, an accuracy level of 80% is acceptable; in other cases, a higher level is required. It is made to learn via a feedback loop to increase the model's accuracy level/percentage. This learning can either be done manually for the model or via self-learning by the model through Robot Sophia, Deep Blue, AlphaGo, etc.

Testing AI-based applications

A typical AI-based application comprises five layers:
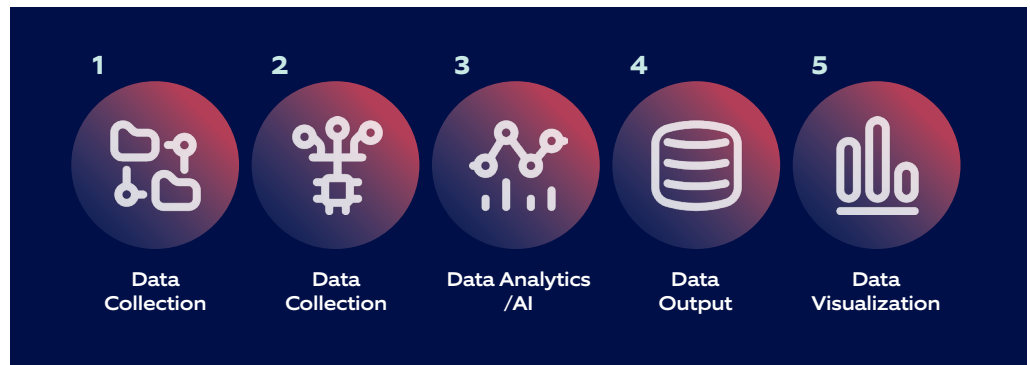
Five layers of a typical AI-based application

While testing such applications, a QA:

- Ensures Data Quality
- Test the data analytics logic and/or the AI model
- Conduct end-to-end testing by following general testability principles

AI needs data, and that too, on a massive scale. Big Data technologies and tools like Apache Spark, Apache Hadoop, MongoDB etc. store, analyze, and manage this voluminous data. AI and Data analytics can further fuel this data to extract insightful information.

QA can help ensure data accuracy, validity, consistency, completeness, timeliness, and uniformity. This can be achieved through a four-step data cleaning process:

1. Removing unwanted observations
2. Fixing structural errors
3. Filtering unwanted outliers
4. Handling missing data

However, the complexity of Big Data ecosystems poses challenges for QA testers, particularly in generating adequate test data. Besides this, there are other challenges like understanding the pulse of the data, dealing with sentiment data, requiring specialized skills, and dealing with frequent requirement changes.

Last but not the least, while testing Big Data applications, we must also ensure that we consider validation points such as data ingestion validation, data processing validation, and data storage validation.

5. IoT

IoT-based applications require a more holistic approach for testing along with strong domain knowledge. QA teams in IoT should be comfortable testing not only software but also hardware. IoT testing is not confined to only a particular environment but involves dynamic environments.

IoT testing involves various components and stacks, with each layer requiring unique testing requirements due to the different ways in which data is handled. Understanding IoT's role in testing requires an understanding of its solution architecture, including sensors, gateways, cloud servers, and user interfaces.

Testing IoT-based applications

The testing of Internet of Things (IoT)-based applications presents various challenges, such as diversity in devices, networks, and scenarios, protocols and interfaces, and frequent environmental changes that require end-to-end testing.

These challenges can be addressed through four focus areas:

1. Device validation
2. Gateway validation
3. Platform validation
4. DB validation
5. Application validation
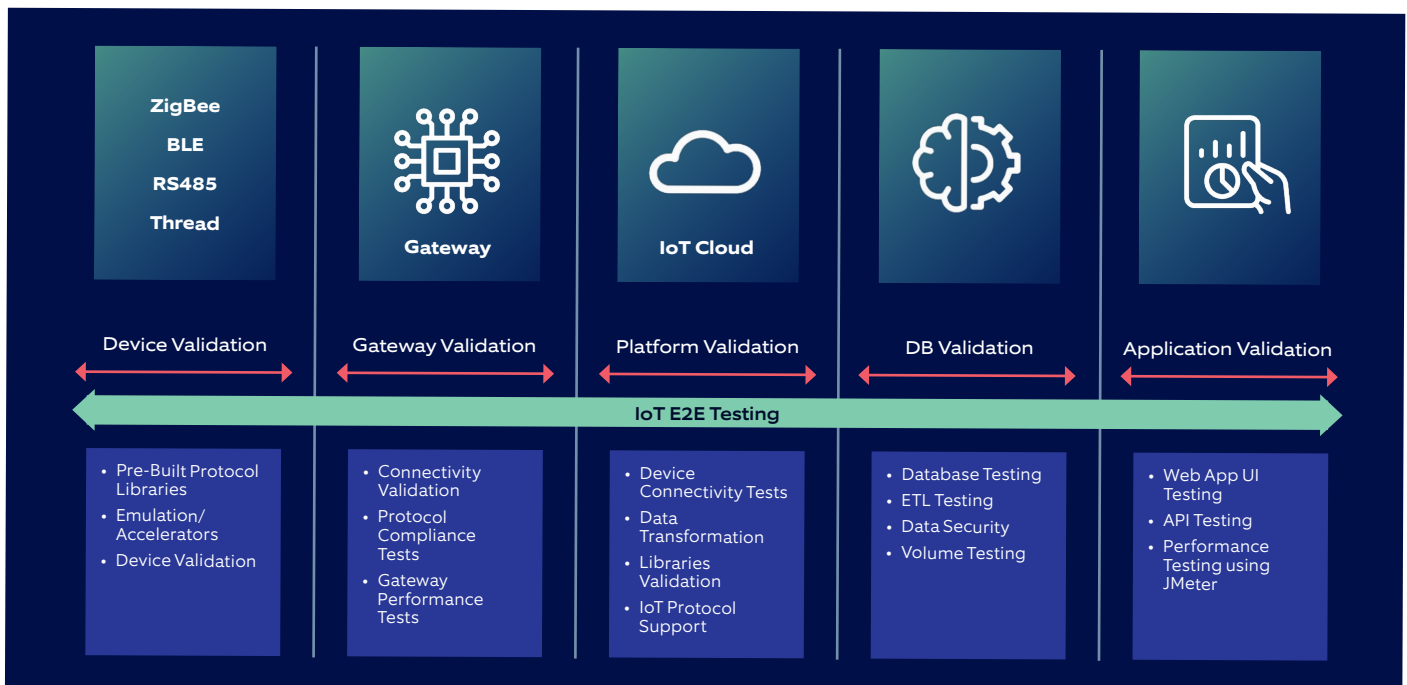
IoT E2E Testing Paradigm

The key trends in IoT include the use of a digital twin for simulating real-world scenarios, digital thread for creating a closed loop between the digital and physical world, the metaverse and IoT, and blockchain technology. The integration of these technological trends with testing can improve the testing process, which enables validation of the vast amount of data generated by IoT devices and the synchronization of real-world data collected by IoT sensors with the metaverse.

Artificial intelligence & IoT

As IoT devices generate voluminous data, it often becomes tough to collect, process, and analyze all of it. Artificial Intelligence (AI) has enabled IoT systems to create intelligent machines that simulate smart behavior and support decision-making without human interference. It helps systems to perform tasks that need human intelligence. Combining these two streams benefits not just the common, lay person but also the specialists. While IoT deals with devices interacting through the internet, AI makes the devices learn from their data and experience.

Some of the advanced use cases of large scale AI and IoT adoption include self-driving cars, robots in manufacturing companies, and retail analytics. Artificial Intelligence and IoT systems testing have the potential to provide reliable results compared to manual testing, which requires extensive human hours and is also prone to inaccuracies and discrepancies.

Machine learning and deep learning technology are set to flourish at a 5.7% CAGR in the global AI in the IoT market. According to a report by Future Market Insights, the global market for AI in IoT was worth US$6.2 billion in 2022 and is projected to reach US$142.4 billion by 2032. During the forecast period from 2022 to 2032, the market is expected to reach a CAGR of 6.2%.

Conclusion

The age of automated testing is here, and the onus is on key decision-makers and stakeholders to take the bull by its horns. With the software industry developing innovative and AI-based solutions at a rapid pace, quality engineering must adapt by developing intelligent and efficient solutions. This is why organizations must have a robust plan to ensure future-ready testing that can hold them in good stead.

At Nagarro, we have defined a roadmap for advanced testing techniques:

- Intelligent Test Automation with self-healing capabilities, AI-based data generators, automated analysis of log files, etc.

- Focus on design thinking and total user experience by shifting from being product-centric to a user-centric.

- Testing of probabilistic logic of AI-based algorithms is a complex area and needs many different testing techniques. With Big Data becoming more and more mainstream, AI testing is going to be a key focus area.

- It is imperative to be lean and agile. The product engineering teams must aim to eliminate anything that does not add value to the product development life cycle.

- Today, enterprises are digitally connected, with many smart devices playing a critical role in their overall operations (IoT). Quality engineering of this digitally connected mesh is going to become more and more mainstream.

Click **here** to learn more about how Nagarro is shaping the future of quality.

References

[1] Surviving Disruption, Leading Change: Winning in the Application Economy - SPONSORED CONTENT FROM CA TECHNOLOGIES (hbr.org)

[2] Five Keys To Understanding Algorithmic Business (gartner.com)

[3] The need for mobile speed (google.com)

[4] Gartner Identifies the Top Strategic Technology Trends for 2022

About the Authors

The team of authors are internationally-renowned top experts on the topics covered in this white paper. They have been speakers in international conferences, thought leaders in key projects, and authors on those topics:

Daniel Eder
Global Practice Lead for Agile Quality

Khimanand Upreti
Co-Lead of Global Business Unit for Accelerated Quality & Test Engineering

Stefan Gwihs
Global Practice Lead for Quality DevOps

Thomas Schweiger
Topic Lead for Hyperautomation & AI in Testing

Vimmi Walia
Global Practice Lead for Emerging Technologies Testing & Performance

About Nagarro

Nagarro is a global digital engineering leader with a full-service offering, including digital product engineering, digital commerce, customer experience, AI and ML-based solutions, cloud, immersive technologies, IoT solutions, and consulting on next-generation ERP. We help our clients become innovative, digital-first companies through our entrepreneurial and agile mindset, and we deliver on our promise of thinking breakthroughs.

Our guiding principles are defined by one word – CARING, denoting a humanistic, people-first way of thinking with a strong emphasis on ethics. Caring guides us as a global company.

We have a broad and long-standing international customer base, primarily in Europe and North America. This includes many global blue-chip companies, leading independent software vendors (ISVs), other market and industry leaders, and public sector clients.

Today, we are over 18,000 experts across 33 countries, forming a Nation of Nagarrians, ready to help our customers succeed.

www.nagarro.com