



5 Software Quality Challenges Solved Leveraging AI

Table of Contents

CHAPTER 1

Current Challenges in Releasing
High-quality Products

CHAPTER 2

Using AI and ML to
Overcome Software Quality Challenges

CHAPTER 3

Measuring Software Quality
in the AI World

CHAPTER 4

The Future of Software Quality

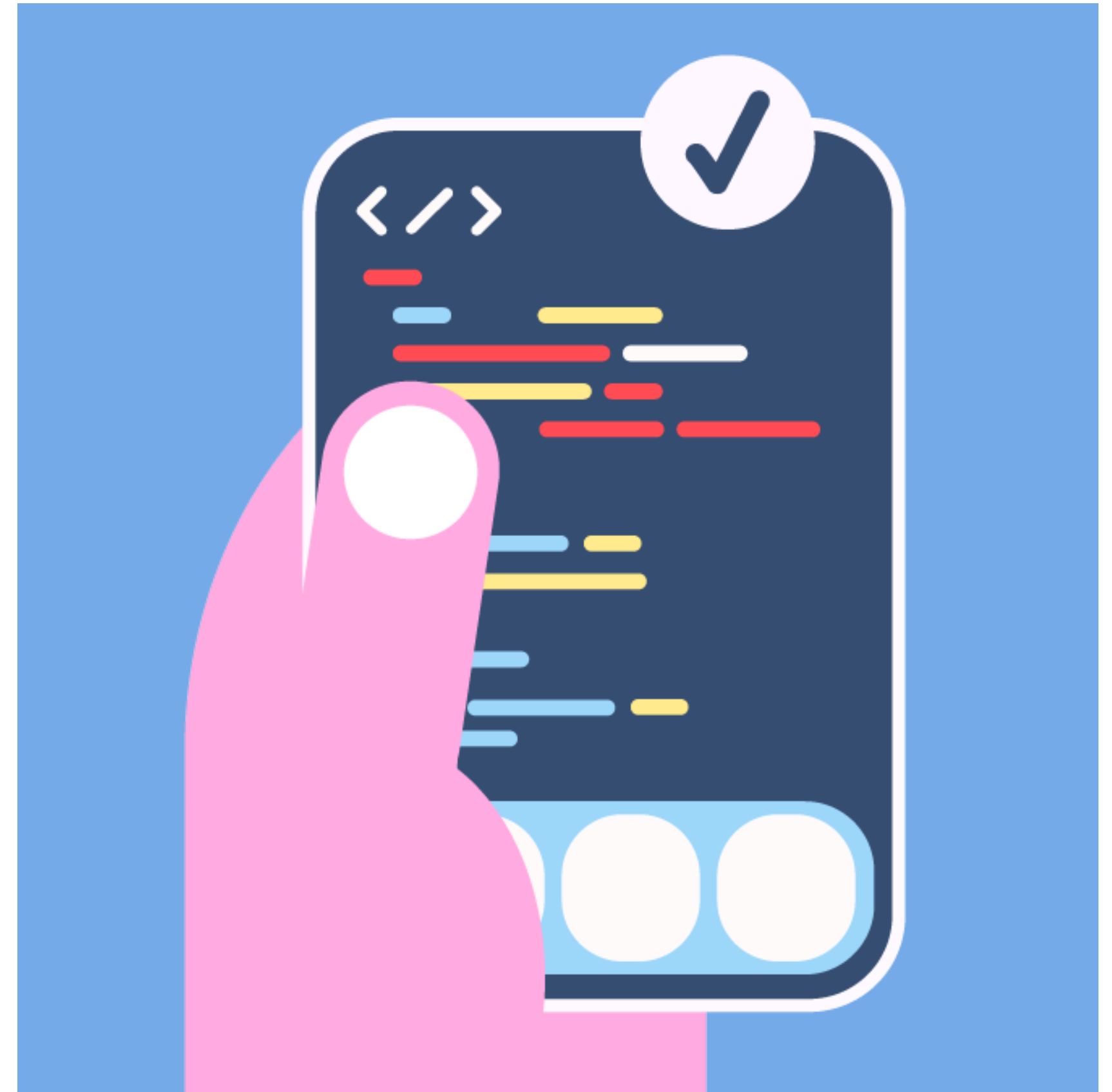
Abstract

Testing practices have changed over the last four decades to keep up with technology and fast-paced software development. Today, companies are investing more in artificial intelligence (AI) and machine learning (ML) to stay ahead.

Software development is riddled with challenges and many are already developing AI-based solutions for those problems. This whitepaper which was written by leading experts in AI and ML to bring you the latest technologies and innovations in the world of software quality.

This white paper will discuss:

- Today's main challenges in releasing a high-quality product
- How to use AI and ML to overcome these challenges
- Where AI and ML can augment major dev and QA initiatives
- What the future of software quality will look like



CHAPTER 1

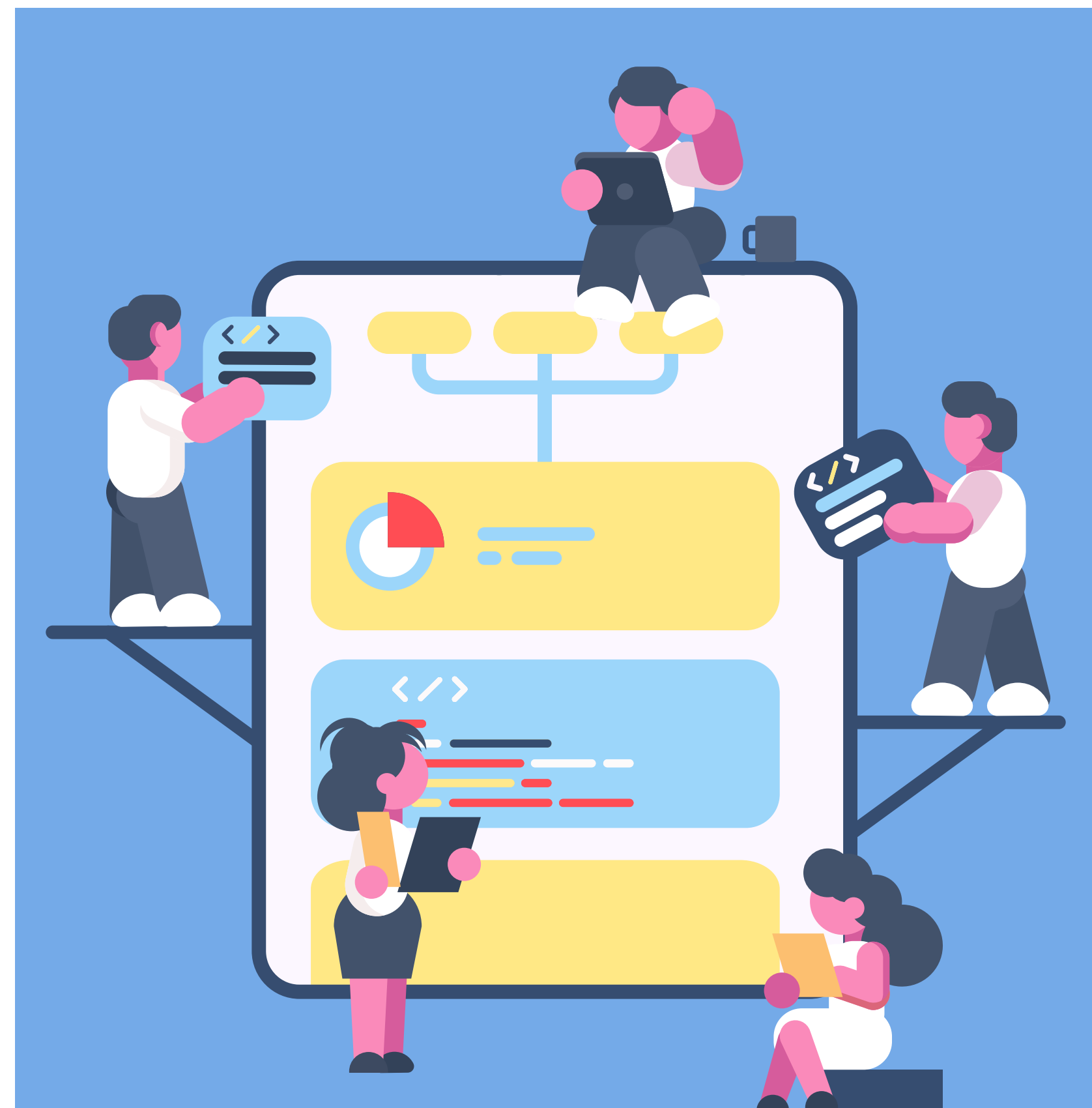
Current Challenges in Releasing High-quality Products

In today's fast-paced development environment, dev teams face a number of challenges in releasing high-quality products. They include

Data Management Challenges

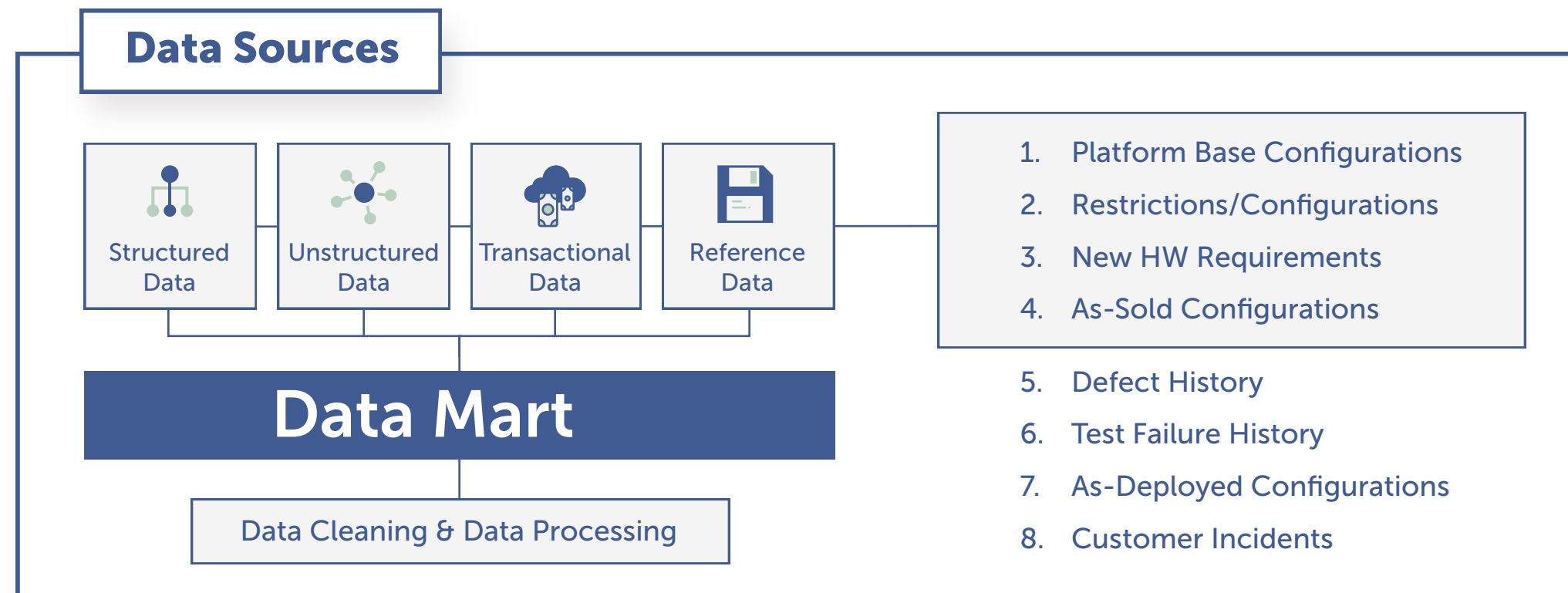
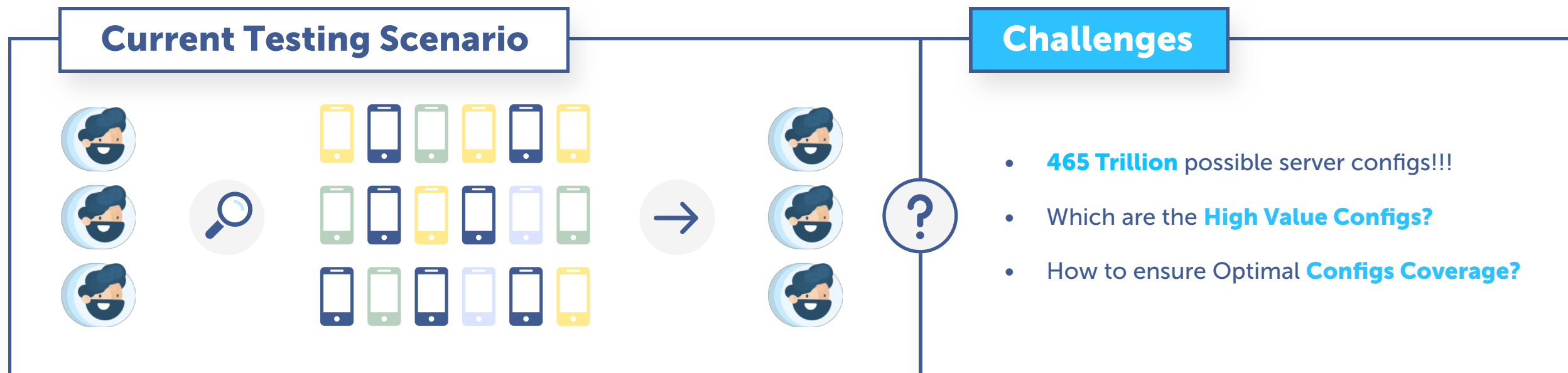
Today's software development pipeline continuously generates massive amounts of data from multiple channels... data which must be taken into consideration during the QA decision making process. This data includes test data (parameters and values to be used), environment data, build data, coverage data and production data.

Although analyzing this constant stream of big data will provide software development teams with invaluable insight and visibility into factors affecting the velocity, quality, and efficiency of their software engineering processes and deliverables, human beings are incapable of processing and making sense of such huge amounts of information. However, disregarding this data will reduce team efficiency, making them more vulnerable to making mistakes.



SUT Configuration Model

"Q" System Under Test



Technology Partner: Dell Performance Analytics Group

Time-based Challenges

With Agile driving the current software development process, dev teams have to code and deploy solutions faster than ever to meet market demands. As such, they have little or no time to perform structural quality analysis or identify critical programming errors, until the testing phase. Increasing release velocity and ensuring the delivery of high-quality software to clients require dev teams to do more with less, avoid waste and shift testing to the left.

The adoption of agile also means that teams have to release applications in frequent incremental iterations. This results in builds that require continuous testing...a situation that puts tremendous pressure on QA teams. To do their job properly, they would have to change on all fronts...starting from the tools and processes they use to their skillsets and organizational structure.

QA: A Bottleneck in Agile Development

In the mid-1900s, it took approximately three years for dev teams to develop and deliver applications that could satisfactorily take care of clients' business needs. This was referred to as the application development crisis. The advent of DevOps (continuous integration, delivery, testing and so on) and Agile reduced time to market by breaking down entire application releases into minimum viable features, enabling dev teams to deliver smaller bits of business value quicker while reducing the risk of shipping something that will not get used because requirements change.

Dev teams collect user information, usage data, and feedback, and use it iteratively during production to improve application over time. However, this means that developers, release engineers, and testers do most of their work post-production... a situation that most testers find uncomfortable.

Software Moving Back to the Front End

The development of JavaScript frameworks such as React, Angular, etc., indicates that software has begun moving back to the front end. This makes code and business logic even harder to test since end-to-end testing on the front end is fragile, flaky, and breaks more often. Due to these factors, dev teams are reluctant to automate testing.

Also, the more teams shift left, the earlier they find bugs. As such, they are forced to write and run lots of integration tests for each software component...a task that is virtually impossible to do manually. This has made QA to become more of a bottleneck in agile development.

Increase in the Number & Diversity of Mobile Devices and Apps

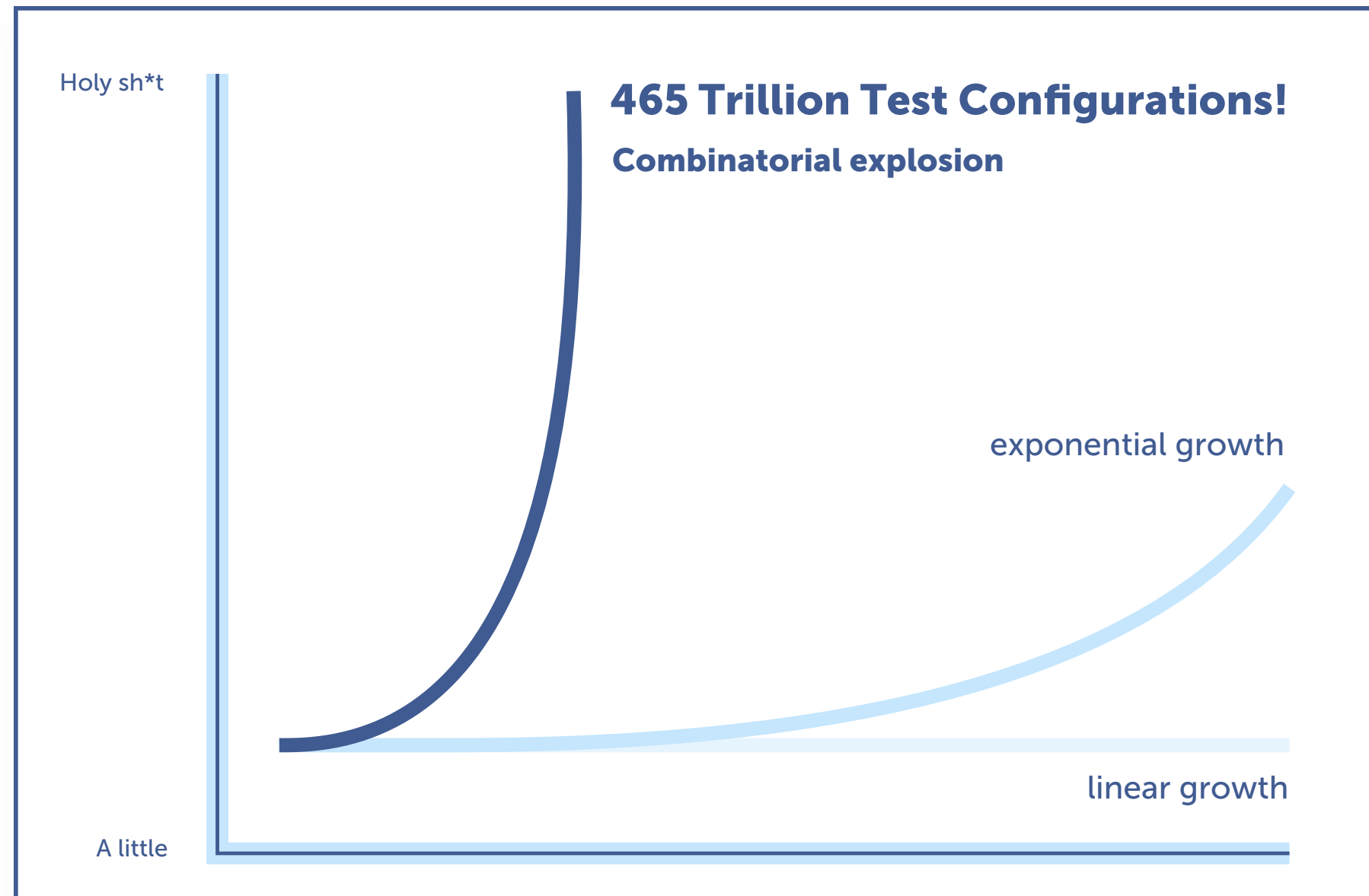
The explosion of the mobile app development space in recent years was caused by the increased use and development of a plethora of smartphone devices and tablets. The popularity of iOS on iPod, iPad and iPhone devices and Android OS on most smartphones also prompted many manufacturers to create their own OS for smartphones and tablets, giving rise to systems like Firefox OS 2, Tizen, Ubuntu, and Windows phones.

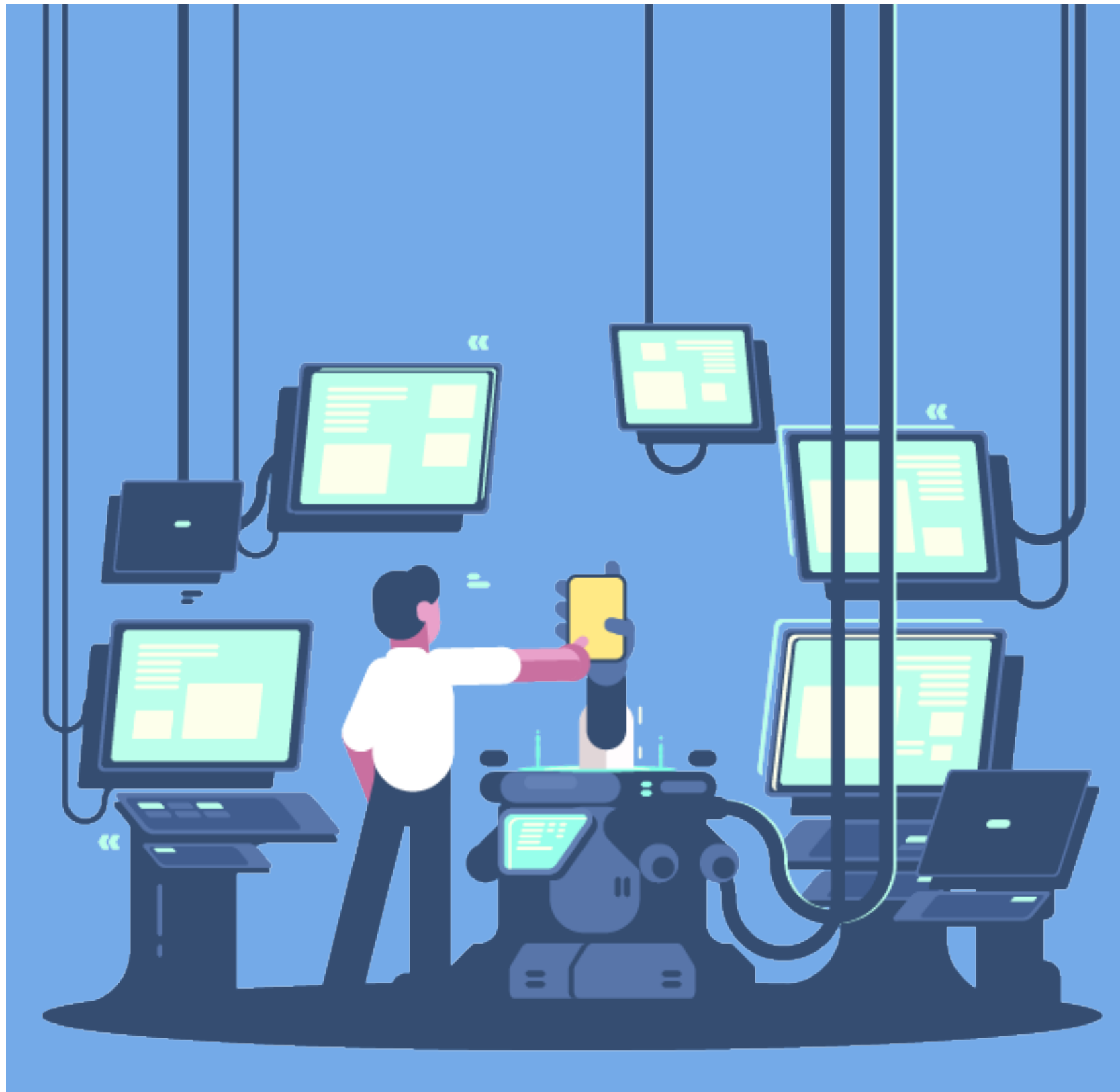
Also, each of these OS's came with their own SDKs (source development tools)...used to develop native applications. With such a diverse number of devices and different OS versions to support and SDK peculiarities to keep track of, as well as the continuously evolving nature of the smartphone and mobile apps market, testing has become more complex and difficult on the mobile front. Similar problem occurs in organizations that are building servers and this can explode into this test configuration metrics. This is a real good opportunity to do this with machines.

Server Software Example

Server Configuration Elements

- Chassis
- Processor
- Memory DIMM
- Memory Configuration
- Hard Disk Drive (HDD)
- Non-Volatile Memory (NVM)
- Embedded Systems Management
- Power Management BIOS
- Power Supply
- Bezel
- Network Daughter Card
- RAID Controller
- Network Interface Card (NIC)
- Host Bus Adapter (HBA)
- Additional PCIe Cards
- Cooling





CHAPTER 2

Using AI and ML to Overcome Software Quality Challenges

When starting up a new software project, dev teams should understand that quality isn't the sole responsibility of QA engineers or the QA team. It's everyone's responsibility. With this mindset governing how teams go about sprint planning and the rest of the software development process, organizations are better able to achieve successful high-quality product release.

Teams should also do all they can to stabilize the entire continuous testing workflow and ensure a mature DevOps pipeline. This would require continuous collection and analysis of data from relevant sources to better inform sprint planning and testing activities.

Leveraging Big Data and Smart Analytics Solutions

The analysis of the massive amounts of data generated by the software development pipeline can help teams overcome most software quality challenges, especially those related to testing. Though it's easy to feel overwhelmed and stop collecting the data, software organizations should instead look for ways to make sense of the data and turn it to their advantage.

A Deeper Look at Test Configurations

One of the methods that fast-paced development organizations can use to solve some of their software quality challenges is by using the data generated by end users to inform their

testing. This data is essential to testing activities since it can help dev teams determine what should be tested through the data-driven selection of test configurations and combinations.

The test configuration matrix in most dev environments can be massive, amounting to billions or trillions of unique configurations that testers must run through.

By leveraging big data and smart analytics solutions to take a deep look at the data sources from which these configurations are pulled, testing engineers can narrow down the number of configurations to those that really matter (i.e. those that end users/customers are actually deploying) as well as test case history and operational data.

Smart analytics solutions help dev teams and QA managers make more informed decisions on testing activities, resulting in improved product quality and increased velocity and efficiency of the entire software development process.

SeaLights' Quality Intelligence Platform

The best solution is one that can apply real-time analytics to thousands of data items, test executions, insights, builds, code changes, and historical data from production to determine software release readiness. Such a solution should be able to analyze quality trends and the risks presented by code changes over time and present QA managers with accurate information (in real-time) to enable them plan more efficient sprints.



This is where SeaLights' Quality Intelligence Platform comes in. Test Gap Analytics (one of SeaLights' functionalities) leverages big data analytics to help dev teams improve software quality by identifying what should be tested and areas where new test development should be executed. Test Gap Analytics works like a funnel in which each layer of information it provides is an additional prism that allows teams to focus attention on the areas of code that matter the most. In this instance, the first prism that is provided is test coverage... which is based on either a specific test type or all the different test types that are run on the application during testing. SeaLights is able to reveal the code coverage no matter the test framework being used.

The second prism provides data on code modification. It identifies code that was modified either during a specific sprint or in any time period of choice. When the data provided by the two prisms are overlapped, the areas of code that are common to both are called test gaps...code areas that were modified or added recently and were not tested by any test type. These are the areas of code that matter the most and represent a much higher risk to software quality.

Since QA teams tend to test everything (leading to inefficiencies and waste of engineering and other resources), identifying test gaps helps them focus on areas that are more likely to contain bugs while ignoring other areas with confidence. In essence, this facilitates data-supported risk-based testing.

The third prism represents important areas of code. SeaLights' production agent enables organizations to expose end user usage data...data that comes from production and represents which methods are used and which aren't.

Combining all three prisms enables organizations to determine the files and methods (in vital areas) that were modified but not tested.

Testim

Testim.io is an AI-assisted test automation solution/platform that enables dev teams to overcome software quality challenges by addressing the following aspects of test automation;

- Authoring and Execution
- Stability
- Reusability
- Extensibility
- Maintenance
- Troubleshooting
- Reporting
- CI/CD Integration

Testim.io uses a hybrid approach to the authoring and execution of tests, simplifying the test automation process such that technical and non-technical members of dev teams can collaborate and start writing automated tests quickly. It uses AI models to speed up the writing, execution, and maintenance of automated tests, enabling developers to rapidly write and execute tests on multiple web and mobile platforms.

Testim.io learns from every test execution, enabling it to self-improve the stability of test cases, resulting in a test suite that doesn't break on every code change. Hundreds of attributes are analyzed in real-time to identify each element vs. static locators and minimal effort is required to maintain these test cases. It also gives dev teams the ability to extend the functionalities of the platform using complex programming logic with JavaScript and HTML and simplifies the process of capturing and reporting of bugs by integrating with popular bug tracking tools.

The platform possesses robust features which not only help to automate tests but also takes care of the other important aspects of testin. These aspects include the execution of tests locally and in the cloud (either in a private or third-party grid), integration with CI/CD systems, capturing logs and screenshots of test runs, and the delivery of detailed reporting on test runs with graphs, statistics, and other related information.

Perfecto Mobile

Perfecto is a leading provider of continuous testing solutions that enable mobile and desktop web application developers and testers to create reliable and stable test scripts, to execute and leverage machine-learning algorithms to analyze the test results.

Perfecto offers an intelligent cloud and SaaS-based platform that hosts globally real smartphone, tablets, and desktop machines for efficient and high-scale test execution. The solution is highly secured, enterprise-grade, and fully integrated into the DevOps tool chain.

CHAPTER 3

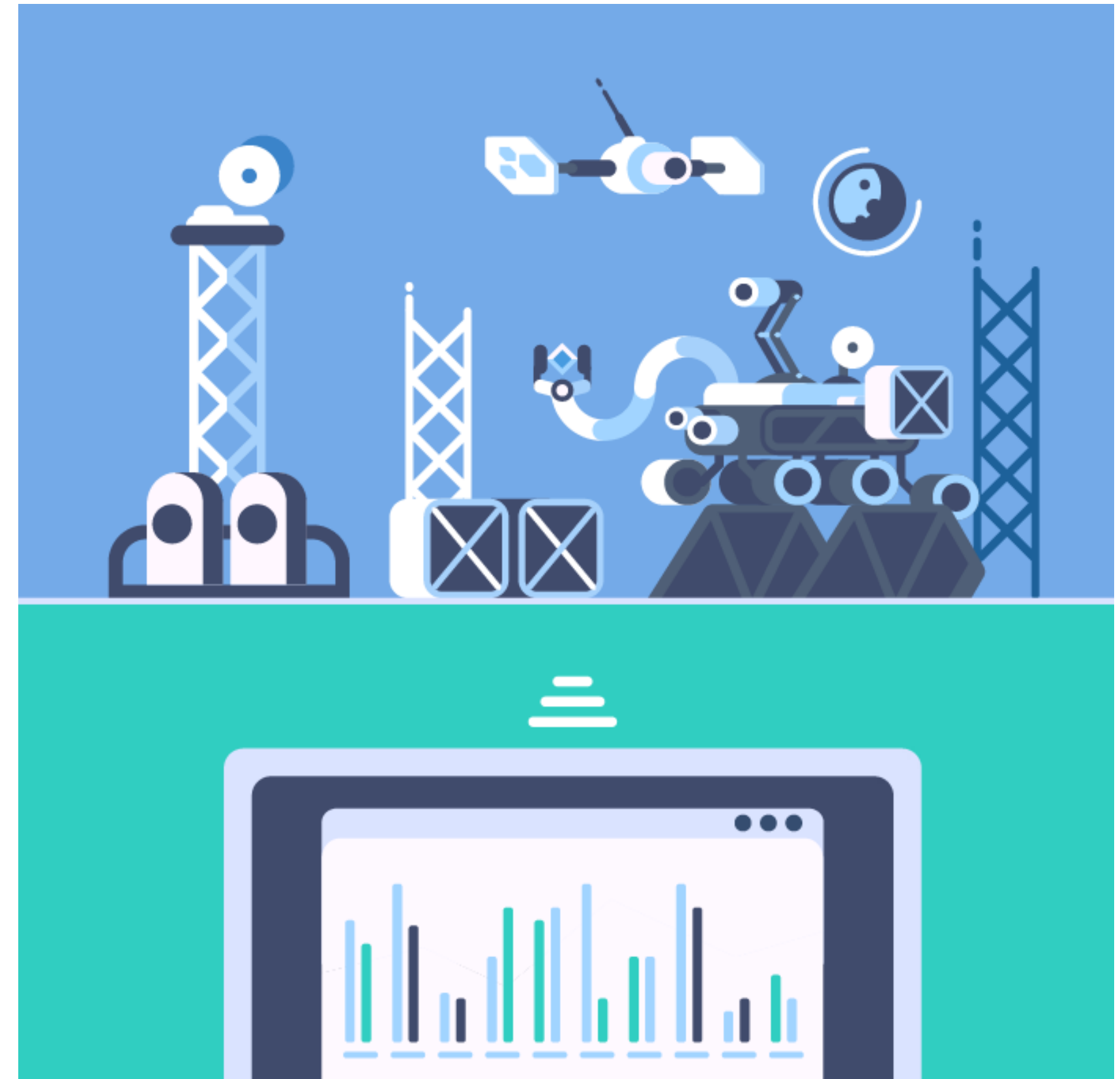
Measuring Software Quality in the AI World

There are a lot of metrics that are relevant to software quality. These metrics include test execution time, failure rates, and coverage areas as well as test coverage per product area, number of open defects in each area, execution status, and so on. Others include how many hotfixes were created, the number of version regressions discovered in the field, customer satisfaction levels, net promoter score, etc.

Although all these metrics are also relevant to the AI world, measuring them would generate a vast amount of information...most of which won't necessarily help dev teams properly track the most important element of all...release quality status.

To solve this challenge, SeaLights provides organizations with a Quality Trend Intelligence Report, which wraps up (in a very easy to consume way) several key quality development KPIs. The report also provides a key metric...known as the quality score...a number between 0 to 10 which indicates the maturity level or quality status of an application.

An application's quality score is calculated using a complex equation that combines several metrics or parameters together. SeaLight's quality score is a new KPI that aggregates a lot of metrics together to form a single kind of absolute/universal metric that organizations can monitor to determine what's most important to them.



CHAPTER 4:

The Future of Software Quality

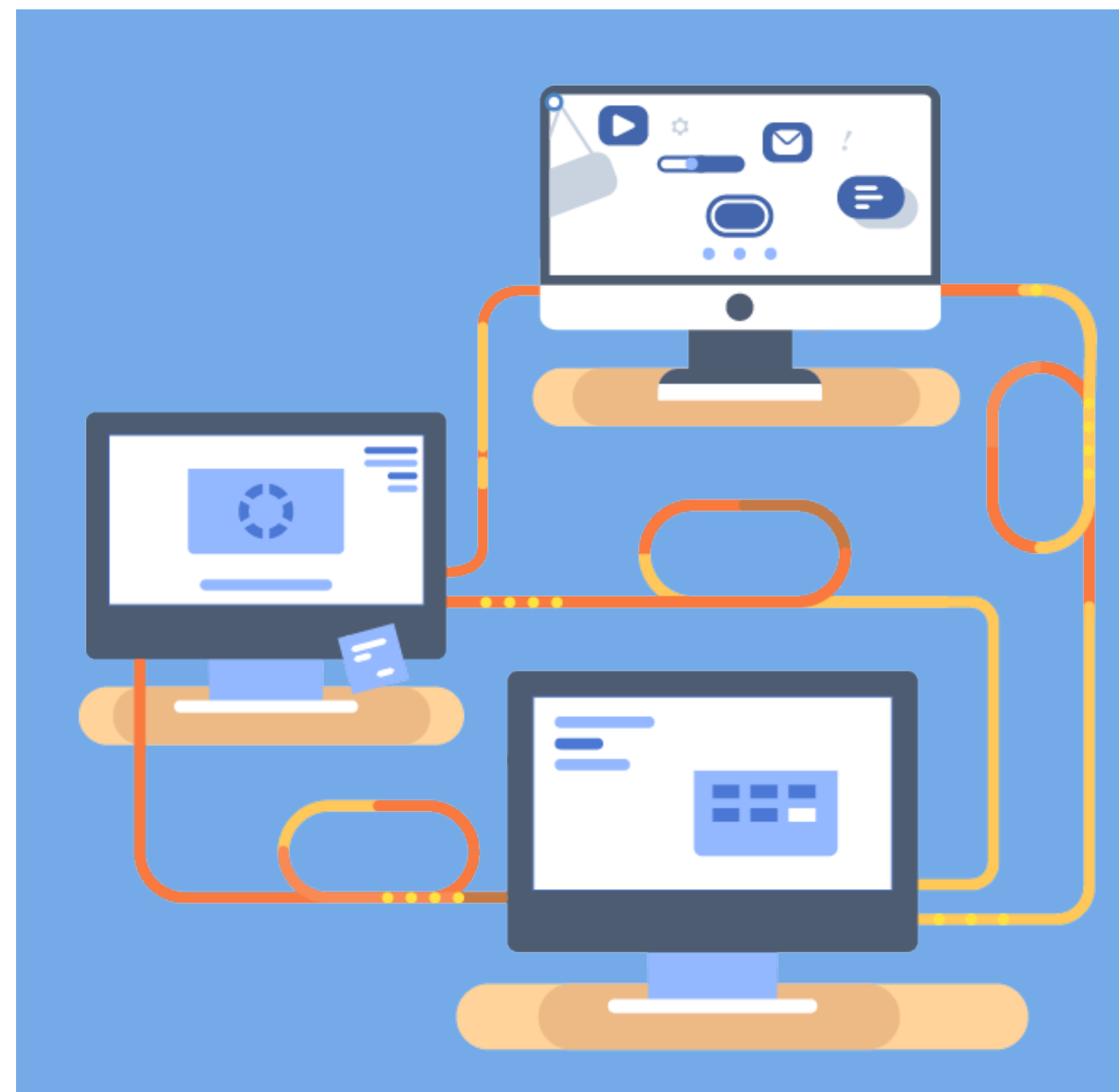
The common theme surrounding the emergence of AI and ML in automated testing is one of continuous improvement. However, these initiatives are driven by people and supported by processes...a development that is preceded by a change in mindset and culture. As such, the advent of AI-assisted test automation has had a significant impact on software development culture, processes, and people.

With big data analytics facilitating the real-time analysis of the huge amounts of data being generated from the current software development pipeline (as well as data flowing in from customer environments and production sites), software organizations are changing their culture to make testing a data-first and a data-informed process.

The analysis of said data has helped dev teams identify what not to test. This has caused a huge shift in the cultural mindset of dev organizations, especially with regards to testing and development leads.

The Effects of AI on Testing Activities

Over the last decade, various developments such as the emergence of agile, DevOps, shifting left, shifting right, and the more recent integration of AI, ML, and analytics into software development activities has radically changed testing activities. Testers must learn/develop new skills and evolve their roles to fit in with emergent technologies/processes.



To avoid becoming obsolete, testers must become lifelong technology learners. They must learn to leverage emerging technologies/processes to deliver more value to their organizations.

Also, some organizations are working on implementing a “feature team structure.” Such a structure entails developers sitting in with QA engineers (now called dev testers) to not only plan activities that should be part of the next sprint but also to look at must-have testing/automation technical.

By involving QA engineers in sprint planning/execution activities right from day one and making them part of the value creation team/process, software development organizations are better able to rapidly release higher quality products to clients and consumers.

Achieving such objectives would require embedding QA into the organization’s DNA through the implementation of practices like behavior-driven development, test-driven development and so on.

The Role of QA Engineers in the Face of Recent Advancements in AI and ML

Recent advances in AI and machine learning technologies have facilitated the creation of test automation tools, solutions, and platforms that are slowly but surely transforming the traditional role of the QA engineer by handing more and more of the manual tasks off to the machines. However, the impact of these technological advancements will be hardest on those who don’t learn to leverage emergent technologies or strive to improve their collaborative skills. Such individuals are usually adamant about using manual processes to execute tasks related to testing/regression testing.

What are our roles?

Strategy

Test Strategy
Predictive Modeling

Planning

Requirements Elaboration
Test Case Analysis
Test Case Development
Test Suite Planning
AI Trainer

Provisioning

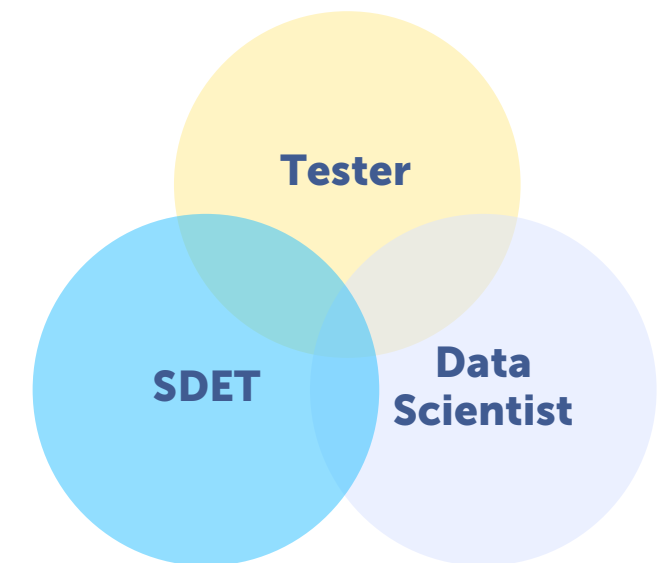
Test Environment Mgmt
Test Configuration Planning
SW/FW Provisioning(DevOps)
Test Data Mgmt

Dev/Testing

Test Case Execution
Test Automation Development
Exploratory Testing
AI Maker
Data Connectors

Tracking

Test Failure Triage
Defect Tracking



The Dawn of Cognitive Automation

Although most dev teams believe that the current age of automation and smart assistants will remove the need for human testers, there are certain areas where the skills of human QA engineers are indispensable.

The widespread adoption of agile has created a growing pile of regression testing...whether dev teams are automating the legacy part of the app or automating all of the new features in the current release cycle.

Continuous testing is concerned with confirming expected behavior as well as looking out for any deviations or anomalies. Since these are tedious, repetitive and robotic tasks, they are best left to automation tools/solutions.

With the advent of AI and ML, software development organizations focused on automating these areas. Using AI/ML frees up the testers time to collaborate and create value. time that testers can use to concentrate on the most important part of testing (i.e. human thinking).

As such, development organizations are beginning to turn their attention on the cognitive aspect of testing or cognitive automation.

In agile teams, value creation occurs within the boundaries of the sprint. In such a collaborative environment, testers work to establish expected behavior during sprints. Their task is exploratory by nature and requires a lot of thinking/ collaboration. These are the sorts of tasks that humans excel at and coincidentally, where testing engineers are really needed.

The future role of QA engineers lies in the realm of cognitive testing as well as test case design and planning. Although QA teams can leverage AI and analytics to help them execute such tasks, the core functions of exploratory and cognitive testing are best performed by humans.

In essence, traditional QA roles must evolve in the face of recent advancements in AI and ML. QA engineers must learn to take advantage of the various test automation tools in the market and harness the power of smart analytics and artificial intelligence to deliver more value to the entire software development pipeline

Testing vs. Checking

Iterative development



The New Challenges in Software Development

As software development continues to evolve, organizations will encounter new challenges. What is currently happening is that organizations are trying to release applications more quickly, knowing that they are first approximations rather than the final product. The software team (including software developers, test engineers, documentation people, release engineers, and so on) continually work on, improve and stay with the application throughout the application lifecycle.

Although this may be slow in coming, the attitudes of software teams (whether DevOps or Agile) are evolving to the point where the team takes responsibility for the application not only for what it does on developer machines build servers, and test environments but also in production.

The Need for More Correlation Between Testing And Production

QA has now become a big part of software development and the business goals of software development organizations. As such, it affects everything...from specs and design all the way to the enterprise.

Ultimately, the success of software development would be measured by end users and clients, and how they experience the application. This will create a need for more correlation between testing and production...a situation that would make tests even more stable.

By leveraging AI and machine learning to analyze production data, dev teams can better understand what is happening, enabling tests to become more stable and smarter (with smart locators).

The Need For a Continuous Monitoring Environment

Previous years witnessed the transition of software development into agile, continuous testing, DevOps, and test automation. Presently, we're beginning to move into the age of cognitive automation. Although automating most checking activities have enabled software organizations to focus more on the value creation process, it has created new challenges.

There will come a time when there are so many automated processes running in the background that organizations will be forced put in place very effective continuous monitoring solutions on both the pre-release and customer environment. These solutions must also monitor the data sources that collate the information that is fed to the AI systems as well as the systems that generate testing results. In essence, there would be so much automation going on that software organizations would need to put in place a continuous monitoring environment.

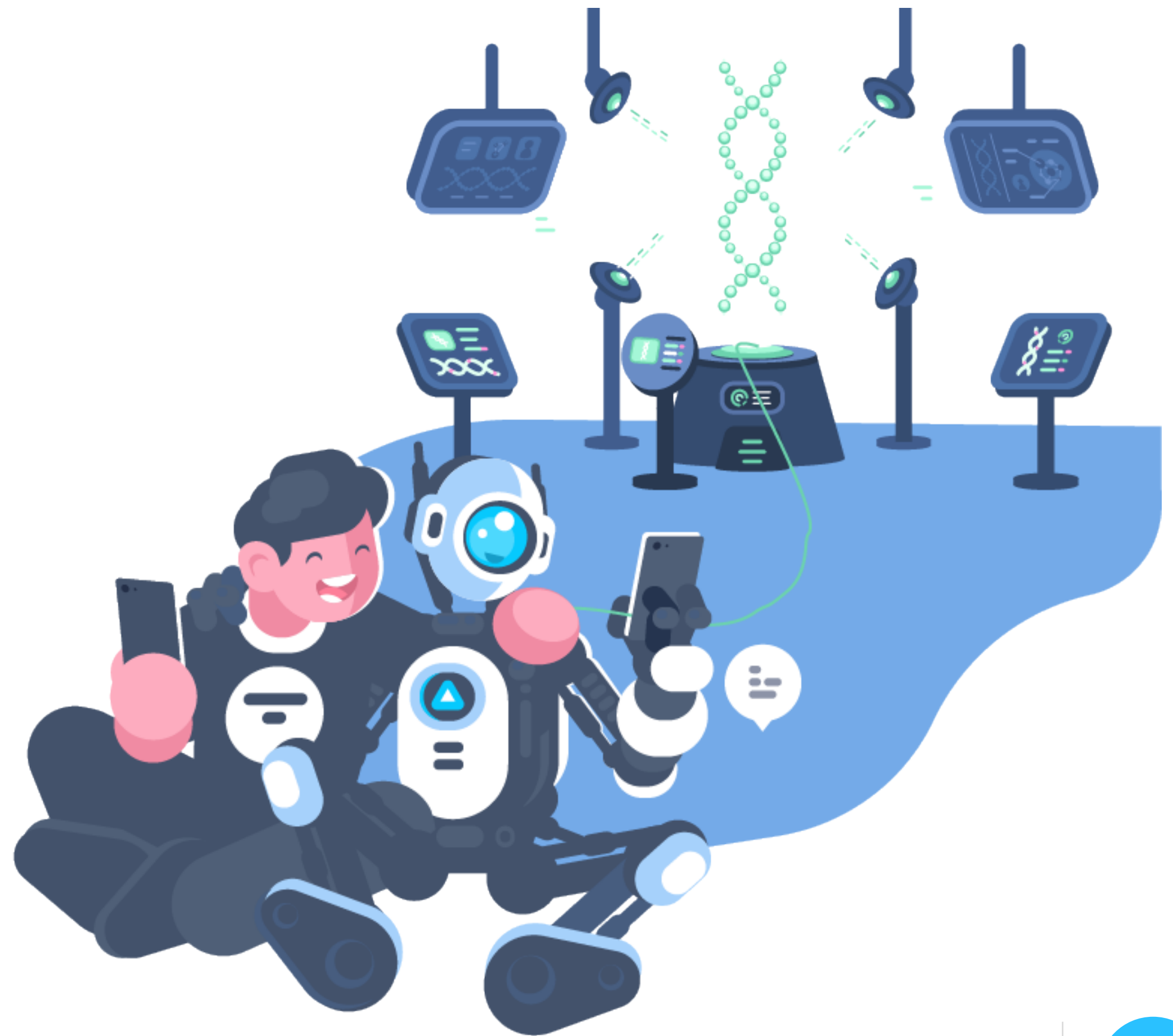
Lastly, there is the challenge of actually testing the new software that is being built with the assistance of AI and machine learning. The World Quality Report shows that the industry is massively adopting AI and machine learning capabilities in order to provide better software and create new offerings.

Since such capabilities differ from the tools being used to provide more value to QA, testers must look for a way to test the quality of applications developed with the assistance of AI and machine learning. Resolving this challenge would require testers and QA teams to review the tools they are using, update the skillsets they bring to the table and look for a viable solution to this new reality.

Conclusion

Improving software quality in the face of the challenges brought about by recent technological advancements and the fast-paced nature of current software development practices requires dev teams to leverage AI-based solutions. AI and ML can help augment major dev and QA initiatives through test automation and the real-time analysis of thousands of data items, test executions, insights, builds, code changes and historical data from production to determine software release readiness.

By automating testing and providing QA teams with relevant insights based on actual data from the software development pipeline, AI and big data analytics can help organizations develop and deliver higher quality products to clients.



Bio



Peter Varhol

is a well-known writer and speaker on software and technology topics, having authored dozens of articles and spoken at a number of industry conferences and webcasts. He has advanced degrees in computer science, applied mathematics, and psychology, and is Managing Director at Technology Strategy Research, consulting with companies on software development, testing, and machine learning. His past roles include technology journalist, software product manager, software developer, and university professor.



Eran Kinsbruner

is the Chief Evangelist and Author at Perfecto, a Perforce company. He authored two books, "The Digital Quality Handbook" and "Continuous Testing for DevOps Professionals." Eran is also a monthly columnist at InfoWorld.com and The Enterprisers Project. Eran is a software engineering professional with nearly twenty years of experience at companies such as Matrix, Sun Microsystems, General Electric, Texas Instruments, and NeuStar. He holds various industry certifications from ISTQB, CMMI, and others. Eran is a recognized mobile testing influencer and thought leader, as well as an experienced speaker in the major software engineering conferences.

He is also a public speaker, researcher, blogger, and a patent-holding inventor for a test exclusion automated mechanism for mobile J2ME testing.



Geoff Meyer

is a Test Architect in the Dell EMC Infrastructure Solutions Group. He drives Test initiatives for Agile Testing, Test Automation, Continuous Testing, Infrastructure as a Service(IaaS), Predictive Analytics and Machine Learning.

Geoff is a member of the Agile Austin community and frequent speaker at international Agile and Testing conferences. He is on the board of the Veterans4Quality.org 501c3 and is a mentor/instructor to Veterans participating in the program, which provides them an on-ramp to a career in software quality assurance.





Oren Rubin

has over 20 years of experience in the software industry, building mostly test-related products for developers at IBM, Wix, Cadence, AppliTools, and Testim.io. In addition to being a busy entrepreneur, Oren is a community activist and the co-organizer of the Selenium-Israel meetup and the Israeli Google Developer Group meetup. He has taught at Technion University, and mentored at the Google Launchpad Accelerator.



Amir Schwartz

is the VP Product & Engineering at SeaLights. Amir has over 16 years of experience in R&D; prior experience includes management positions at Mercury Interactive, HP Software, and Jive Software. Responsible for product requirements and definition, including utilizing feedback from the field and translating it into product changes and improvements, as well as leading all development and innovation efforts.



Thank You!

For more information contact us at

info@testim.io

or

contact@sealights.io



www.testim.io