EBOOK

# LEVERAGING MACHINE LEARNING TO BETTER UNDERSTAND & IMPROVE TEST AUTOMATION

APPLAUSE



### Introduction

In recent years, software development has largely shifted to Agile and DevOps methodologies, aiming to finally make a more mature CI/CD pipeline a reality. As part of this leap forward, organizations automated several processes, including coding, monitoring, and of course, testing.

To provide value to customers, DevOps teams must test quickly and upon each pull request. When testing is done at scale like this, it results in huge quantities of test data. Testing on fragmented digital platforms that are dependent on complex environments makes it difficult to analyze test data and provide fast feedback.

In addition, DevOps processes involve a wide range of practitioners, including product managers, product owners, developers, test automation engineers, business testers and operation engineers. This means the data originates from different tools and personas, and needs to be normalized. To succeed in a complex DevOps digital journey, teams must adopt automated continuous testing that is reliable, self-maintained (as much as possible) and brings value with each test execution cycle.



According to a <u>recent study</u>, issues with test scripts and frameworks "account for 40% of all issues that DevOps teams face." These scripting and framework errors can include:

- » Wrong element locator strategy
- » Unhandled events, such as application popups
- » Incoming and unexpected events
- » Time and sync issues

Without stable automation, DevOps teams lose confidence in the test automation results and, instead of adding value to the team, test automation becomes a liability.

## Common reasons for test automation failures includes scripts, backend issues, lab issues and orchestration

How can these challenges be addressed? The answer is by engaging in a smarter test automation strategy that is backed by machine learning (ML). ML is well-positioned to help DevOps teams in these areas:

- » Make sense of extremely high quantities of test data
- » Make actionable decisions around quality for specific releases
- » Enhance test stability over time through self-healing and other test impact analysis (TIA) abilities

Read on for more information about how machine learning can be helpful for DevOps teams with automated testing.



### How Can ML Help in a DevOps Setting?

Scaling test automation and managing it over time remains a challenge for DevOps teams. Development teams can utilize machine learning both in the platform's test automation authoring and execution phases, as well as in the post-execution test analysis that includes looking at trends, patterns and impact on the business.

Before diving deeper into how ML can help during both of these phases of the test automation process, it is important to understand the root causes to why test automation is so unstable when not utilizing ML technologies:

- The testing stability of both mobile and web apps are often impacted by elements within them that are either dynamic by definition (e.g., <u>react native apps</u>), or that were changed by the developers.
- » Testing stability can also be impacted when changes are made to the data that the test is dependent on, or more commonly, changes are made directly to the app (i.e. new screens, buttons, user flows or user inputs are added).
- » Non-ML test scripts are static, so they cannot automatically adapt and overcome the above changes. This inability to adapt results in test failures, flaky/brittle tests, build failures, inconsistent test data and more.

Let's dig into a few specific ways that machine learning can be valuable for DevOps teams:

### Make sense of extremely high quantities of test data

Organizations that implement continuous testing within Agile and DevOps execute a large variety of testing types multiple times a day. This includes unit, API, functional, accessibility, integration and other testing types.

With each test execution, the amount of test data that's being created grows significantly, making the decision-making process harder. From understanding where the key issues in the product are, through visualizing the most unstable test cases and other areas to focus on, ML in test reporting and analysis makes life easier for executives.

With AI/ML systems, executives should be able to better slice and dice test data, understand trends and patterns, quantify business risks, and make decisions faster and continuously. For example, learning which CI jobs are more valuable or lengthy, or which platforms under test (mobile, web, desktop) are faultier than others.

## With each test execution, the amount of test data that's being created grows significantly.

Without the help or AI or machine learning, the work is error prone, manual and sometimes impossible. With AI/ML, practitioners of test data analysis have the opportunity to add features around:

- » Test impact analysis
- » Security holes
- » Platform-specific defects
- » Test environment instabilities
- » Recurring patterns in test failures
- » Application element locators' brittleness
- » And more

## Make actionable decisions around quality for specific releases

With DevOps, feature teams or squads are delivering new pieces of code and value to customers almost on a daily basis. Understanding the level of quality, usability and other aspects of code quality on each feature is a huge benefit to the developers.

By utilizing AI/ML to automatically scan the new code, analyze security issues and identify test coverage gaps, teams can advance their maturity and deliver better code faster. As an example, <u>code-climate</u> are able to automatically review any code changes upon a pull request and spot quality issues, and optimize the entire pipeline. In addition, many DevOps teams today leverage the feature flags technique to gradually expose new features, and hide them in cases of issues.

## By utilizing AI/ML to automatically scan the new code, analyze security issues and identify test coverage gaps, teams can advance their maturity and deliver better code faster.

With AI/ML algorithms, such decision making could be made easier by automatically validating and comparing between specific releases based on predefined datasets and acceptance criteria.

## Enhance test stability over time through self-healing and other test impact analysis (TIA) abilities

In traditional test automation projects, the test engineers often struggle to continuously maintain the scripts each time a new build is being delivered for testing, or new functionality is added to the app under test.

In most cases, these events break the test automation scripts — either due to a new element ID that was introduced or changed since the previous app, or a new platform-specific capability or popup was added that interferes with the test execution flow. In the mobile landscape specifically, new OS versions typically change the UI and add new alerts or security popups on top of the app. These kinds of unexpected events would break a standard test automation script.

With AI/ML and self-healing abilities, a test automation framework can automatically identify the change made to an element locator (ID), or a screen/flow that were added between predefined test automation steps, and either quickly fix them on the fly, or alert and suggest the quick fix to the developers. Obviously, with such capabilities, test scripts that are embedded into CI/CD schedulers will run much smoother and require less intervention by developers.

#### How can AI/ML and self-healing ability help a test automation framework?



» Automatically identify change made to element locator (ID) or screen/flow added between predefined test automation steps



» Quickly fix the changes on the fly, or alert and suggest quick fix to developers



» Test scripts embedded into CI/CD schedulers will run smoother, require less intervention by developers An additional benefit would also be the reduction of "noise" within the pipeline. Most of the above mentioned brittleness in testing are not real defects, but interruptions to automation scripts. By eliminating them proactively through AI, teams will get more time back to focus on real issues.

## Machine Learning for Web and Mobile App Testing

In the test creation and execution phases ML solutions can help record and self-heal test automation scripts, regardless of the element locator ID. Modern ML solutions are capable of modeling a mobile native app or a web application that are by nature complex and unexpected, and help create test automation flows or scenarios that can be plugged into CI builds for acceptance and functional testing.

During test execution, ML algorithms can automatically adjust the flows to handle changes that were either made to the UI components in the app, or that are different across platforms. Some of the ML tools utilize visual testing and other advanced app crawlers through a reinforcement learning method/bot that simply maps the application screens and expedites test creation.

Other ML tools utilize the Appium framework at their core and use a self-healing element locator algorithm during test execution runtime via a fully codeless creation technique. These solutions generate an independent application map end to end that identifies how to get from one screen or button to the next, even if a new screen is added in the middle.

This map does not rely on the element locators, but rather looks at the app from a visual perspective, and searches throughout all of the app's pages for the best and most reliable "pattern" to get from point A to point B.

Lastly, ML can ensure that the test lab is always on and up to date by automatically identifying – and fixing – platform disconnection or availability issues. Another benefit of ML within the mobile and web test lab is the ability to identify OS versions or platform configurations that are not aligned with the market adoption, and so pose a business risk by not being covered in testing.



ML Training Model for Application Mapping and Decomposition (Source: PNSQC.org)

The end results of utilizing such ML methods are faster test automation creation, robust testing environments and faster feedback upon application code changes.

## **Machine Learning for Trends and Patterns Identification**

The whole purpose of investing time and resources into test automation creation is to be able to answer on-demand quality and business risk-related questions. As digital test data continuously grows along with the cadence of test cycles, ML systems can help identify patterns in the quality of web and mobile applications, and advise DevOps leaders where to focus their attention in order to optimize ongoing processes. Only at the test reporting and analysis phases can DevOps teams gain value from the overall testing activities.

ML algorithms can filter noise based on issues, platforms, functional application areas (e.g., login screen, fund transfer, etc.), history, coverage and other factors.

The algorithms enable decision makers to dive deeper into the application quality and scope the next software release better. In other words, an efficient ML model can employ a continuous testing impact analysis process within DevOps that will highlight the value of test automation while minimizing the noise and waste throughout the pipeline.

When thinking about ML within the DevOps pipeline, it is also critical to consider how ML is able to analyze and monitor ongoing Cl builds, and point out trends within build-acceptance testing, unit or API testing, and other testing areas. An ML algorithm can look into the entire Cl pipeline and highlight builds that are consistently broken, lengthy or inefficient. In today's reality, Cl builds are often flaky, repeatedly failing without proper attention. With ML entering this process, the immediate value is a shorter cycle and more stable builds, which translates into faster feedback to developers and cost savings to the business.

There is no doubt that ML will shape the next generation of software defects with new categories and classification of issues. But most importantly, it will increase the quality and efficiency of releases.



## Automation Essentials for the Age of Agile

The white paper has a six-step process to success with test automation in an Agile methodology.

DOWNLOAD

### **About Applause**

Applause is the worldwide leader in crowdtesting and digital quality. Software is at the heart of how all brands engage users, and digital experiences must work flawlessly everywhere. With highly vetted testers available on-demand around the globe, Applause provides brands with a full suite of testing and feedback capabilities. This approach drastically improves testing coverage, eliminates the limitations of offshoring and traditional QA labs, and speeds time-to-market for websites, mobile apps, IoT, and in-store experiences.

Thousands of leading companies — including Ford, Fox, Google, and Dow Jones — rely on Applause as a best practice to deliver high-quality digital experiences that customers love.

Learn more at www.applause.com

#### NORTH AMERICA 100 Pennsylvania Avenue Framingham, MA 01701 1.844.300.2777

#### EUROPE

Obentrautstr. 72 10963 Berlin, Germany +49.30.57700400

#### ISRAEL

10 HaMenofim Street Herzliya, Israel 4672561 +972.74.757.1300

