Kobiton

AI For All

How Hyperautomation Creates Better Testers

Sections:

For years, the dominant train of thought in the mobile app development industry has centered around a sort of "artisanal" curation throughout the development process.

In-house teams and their project managers work tirelessly to take an idea through the ringer, writing code and testing it for usability while trying to assure all sides of the equation that they're being heard and efficiency is being upheld. In many ways, this specialized method of operating is exemplified in the testing realm.

Manual and primitive automated testing have been the choice du jour for mobile app teams. They represent the typical way that testing is conducted, through testers developing plans and criteria to measure, then ranging through an assortment of test cases themselves as they diligently attempt to bring an app to market.

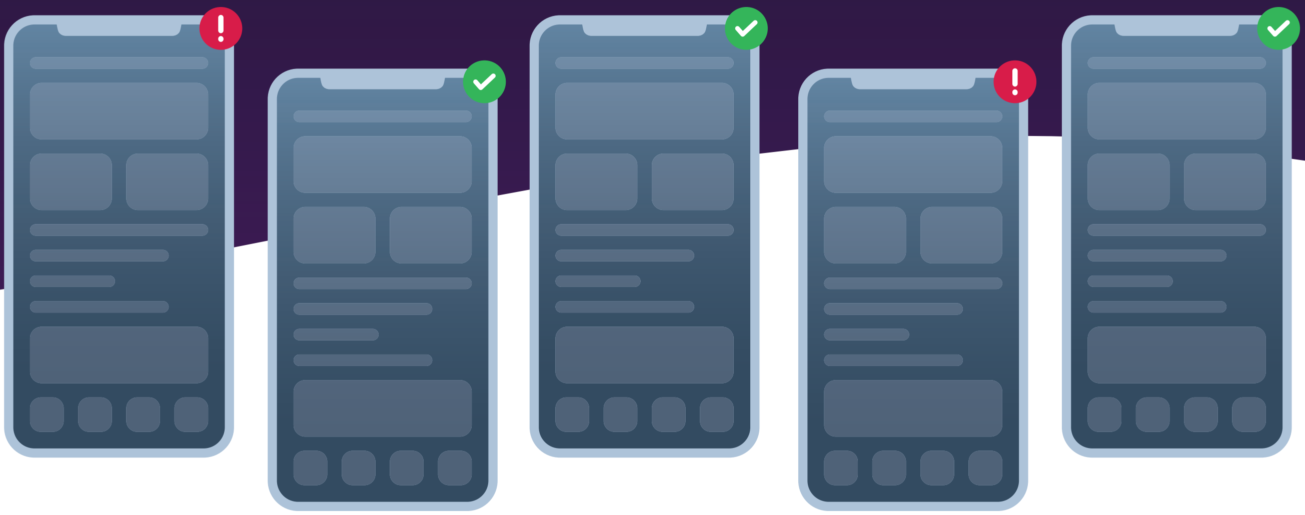
And yet, the entire concept of applying hands-on, pre industrial methodology to a modern industry operating within the forefront of the Fourth Industrial Revolution seems antithetical when you take a step back. It would be not unlike using a sophisticated factory operation to bake tens of thousands of bread loaves, only to ask workers to package them by hand.

So if not manual testing, then what?

The aforementioned Fourth Industrial Revolution holds a focus on integrating artificial intelligence (AI) into a variety of societal realms, and mobile app development should be no different. AI test automation represents the forefront of testing, a means to an end that can ultimately reduce costs, provide for greater efficiency in testing, and expedite the entire process in a way that manual testing simply cannot achieve. Contrary to what some may think, it opens doors for teams to take on greater work roles for employees by reducing the technical prowess needed to operate as a tester. However, as implied, AI can be something of an It-That-Should-Not-Be-Named for manual testers and developers in general. Overcoming these fears as an organization and embracing the notion that AI test automation will simultaneously improve the efficiency of a product's development as well as the outlooks for employees is a major component of what we intend to do. With a proper understanding of how AI can function for testing teams, we believe that testers will see the merit and recognize how their work lives can be positively influenced by the introduction of artificial intelligence.

The Skepticism Surrounding AI Testing:

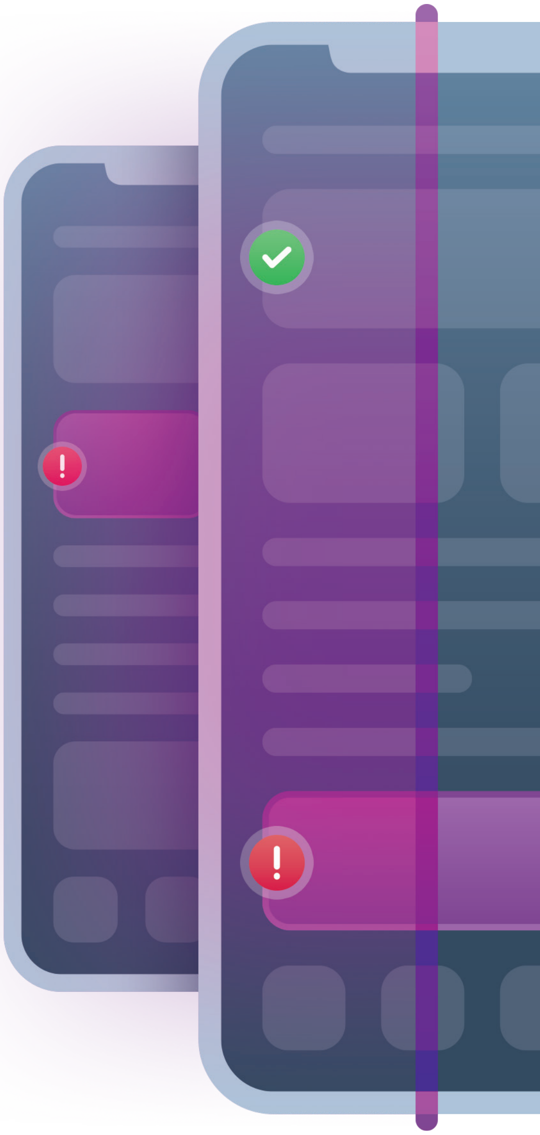
Dispelling the Myths



Much of the fear and trepidation around AI testing derives from the notion that artificial intelligence will replace the workforce with computer-based learning and machinery. Manual testers are often no different in that regard, and one could understand why this hesitancy would develop. With that said, AI testing is not designed to replace testers; it's designed to improve their quality of work and free them from some of the more monotonous tasks that can become ripe for human error from simple task boredom. AI testing is further meant to expedite the time it takes to test, leading to quicker and more succinct product corrections, a faster time to market, and increased revenue which leads to company growth and, eventually, more money for everyone involved.

The efficiency component can further create a situation in which companies that choose to avoid AI testing are ultimately put at a sincere disadvantage when competing with companies that embrace modern testing methods. Timing is everything; every single day matters in the production cycle. Allowing competitors to harness an advantage through AI testing while falling back on traditional manual testing methods could prove to be a self-created hurdle in the industry.

There are additional concerns amongst project managers that AI testing can lead to fallacies in operation that can result in critical errors making it to market. A simple metaphor is considering the bulk of news stories that surround Tesla "self-driving" automobiles. AI learning is a massive component of what Tesla aims to do, but the media is quick to disseminate news stories when a Tesla automobile fails in some capacity. At the source of this is a very human concern that relinquishing control can lead to catastrophic failures, that the machine-makers could be misguided by the machines.



However, when it comes to AI testing, the artificial intelligence piece is largely predicated on factors that human testers identify as critical to operation. The testing software is constantly learning and integrating what the tester desires, measuring for variables that the tester is interested in testing, then anticipating future testing variable measures based off of the AI learning process. The fears of a testing program being initiated and somehow leading to a combustion of the app itself are largely the product of our own human aversion towards fully trusting the technology that we ourselves created. Zooming out to see the bigger picture of how AI testing can improve a multitude of attributes for your business should help to acknowledge AI test automation not as something to be concerned about, but rather something to build off of for a successful app development team.

The Basics of How AI Test Automation Functions

AI test automation operates by employing AI software that has the capability of self-monitoring and learning, both using and generating progressive algorithms that can access data points, extracting segments of data to develop patterned learning, and then using what the software has learned to turn around and test new software accordingly.

In a world of testing where continuous testing is the name of the game, AI is built on the concept that continuous learning will augment the testing process with minimal effort required from human testers. One of the critical points of AI test automation centralizes around the concept of scriptless testing. While this may sound like an end to coding as we know it, that's simply not the case. It does, however, provide an opportunity for testers to simplify the process, opening the doors for testers from a multitude of backgrounds beyond computer engineering. Rather than writing the entirety of code for an automated test, testers can skip the more menial steps by indicating to the software the steps and variable points that a team intends to test, allowing the AI to essentially proceed with the coding side itself. These no-code automation test scripts are typically reusable and can continue to be scaled to different facets of the testing process, or different projects entirely, dramatically reducing the operation time that it would take for manual testers to write and edit code for manual testing themselves. This process can also happen in parallel with the development process, ensuring that QA doesn't ultimately become a bottleneck in the software development life cycle (SDLC).

The Benefits of AI Test Automation

The positives of AI testing can be summarized broadly along six measurables:

- ✓ Reducing Errors
- ✓ Decreasing Time-to-Market
- ✓ Resource Allocation
- ✓ Broader Test Coverage Net
- ✓ Visual Validation
- ✓ More Accurate Results

Each of these represent a component of your SDLC that may have given rise to frustration in the past, and each can represent a major hurdle in the race to the market for an app developer.

In terms of reducing errors, AI test automation can dramatically reduce the defects that your app may otherwise hit the market with. Of course, we all know how troublesome it can be to have a freshly-launched app crash on users immediately after downloading, so any steps that could remedy this potential problem are worth their weight in gold. AI scripts can quickly detect bugs early in the testing process, saving days and weeks of time that may otherwise transpire under the auspices of a flawed app. As mentioned previously, AI also eliminates human errors that can result from routine coding that don't directly stimulate and inspire the coder, who's essentially just going through the ropes trying to get the product to the finish line and testing the more top-down performance analytics that are integral to the app's function, but don't necessarily represent the most exciting components of the app.

Decreasing the expected time-to-market speaks for itself. Speed is everything in the world of mobile app development, but risking quality for expedited testing is always a tightrope that can snap on teams that choose to overlook the essentials of the testing process. Having an AI tool that can be deployed in conjunction with the product development, then learn patterns through deep learning as the testing process is conducted, will free up days, weeks and potentially months for a development team, allowing your company to outpace competitors, begin work on patches and updates sooner, and transition to additional money making projects that grow your business and your brand.

There are additional resource incentives to AI testing. Besides saving time on current projects, time can be saved on future projects as well. This is through AI testing automation learning as it goes, allowing for tests to be reapplied in the future to subsequent projects, saving budgets that would be devoted to developing manual tests from the ground up time and time again. The efficiencies of the software allow for personnel to devote time to other projects, which can further reduce budgets per project and strain on your team.

The type of testing your team receives under AI automation has expanded coverage in comparison to typical manual testing. This increased test coverage ranges everything from file contents, data tables, internal program states and memory, all with seamless measurements. AI software will further test performance with the same methods of machine learning, examining expected outcomes and ensuring that apps are hitting their marks when it comes to testing experiences within an app.

When conducting manual testing, the potential for human error is always present in coding and variable assessment. Even the strongest coders have momentary oversights and omit small parcels of code that can lead to hours or days of corrective remedying to get a project back on track. AI testing automation and scriptless testing work to reduce and mitigate that risk by creating a more accurate test cycle. Repetitive tasks are handled more effectively by AI scriptless testing operating under a premise of no-code automation, so the odds of a tester writing code that could cause confusion in the testing cycle is largely curbed. This, in turn, leads to a more refined and accurate testing cycle with results that can be more efficiently delivered to dev teams with a higher degree of confidence.

The final benefit to AI automation testing addresses a frequent concern that teams have when it comes to taking their hands off the wheel in the testing cycle. This issue centers around what happens when manual testers are not present in the testing process to assess visual parameters and functions within an app. Sure, automated testing can run through lines of code and check for deficiencies, but what about the visual cues that the consumer directly interfaces with?

Some may be surprised to learn that this is a parameter where AI test automation excels. Artificial intelligence software in the testing realm is designed to account for (and specialize in) pattern recognition and image recognition, visually checking for bugs and performing a visual test of applications, much in a similar way that manual testers would. AI testing can manage the visual elements of an app, testing for engaging properties and proper functioning. Furthermore, AI can account for the gestures and UI dynamics that a user would manipulate when operating an app, unbounded by size, shape and pixel level. In this way, AI testing can operate on an enhanced visual level, seeing things that a manual tester may be physically unable to witness.

With the benefits of AI automated testing outlined, let's consider the key approaches to AI testing that would shape how a testing team conducts their process. It's important to remember, again, that the testing approach allows for testers to be active and involved, while simultaneously opening the doors to testers of more varied professional backgrounds to approach the process.

Key Methods of Testing Using Artificial Intelligence

AI testing automation can typically be divided into four methods and four key types of tools that guide testing teams. These methods describe different approaches that teams can take to testing their products using AI-driven software and machine learning. These methods can happen within a DevOps team framework, whether you're using agile or another system to outline your SDLC procedures.

- **Declarative testing** describes a type of test in which the system decides how to perform the test. The aims of the test are spelled out most typically in domain-specific languages. At the end, the system figures out how to best carry out the test, harnessing the specs of AI automated testing to their fullest potential.

- **Visual testing** implies a software testing technique in which gestures and user interfacing are tested and measured. The software will leverage a form of image-based learning and screen comparisons to assess how the aesthetics and general feel of the app complies with dev team expectations.

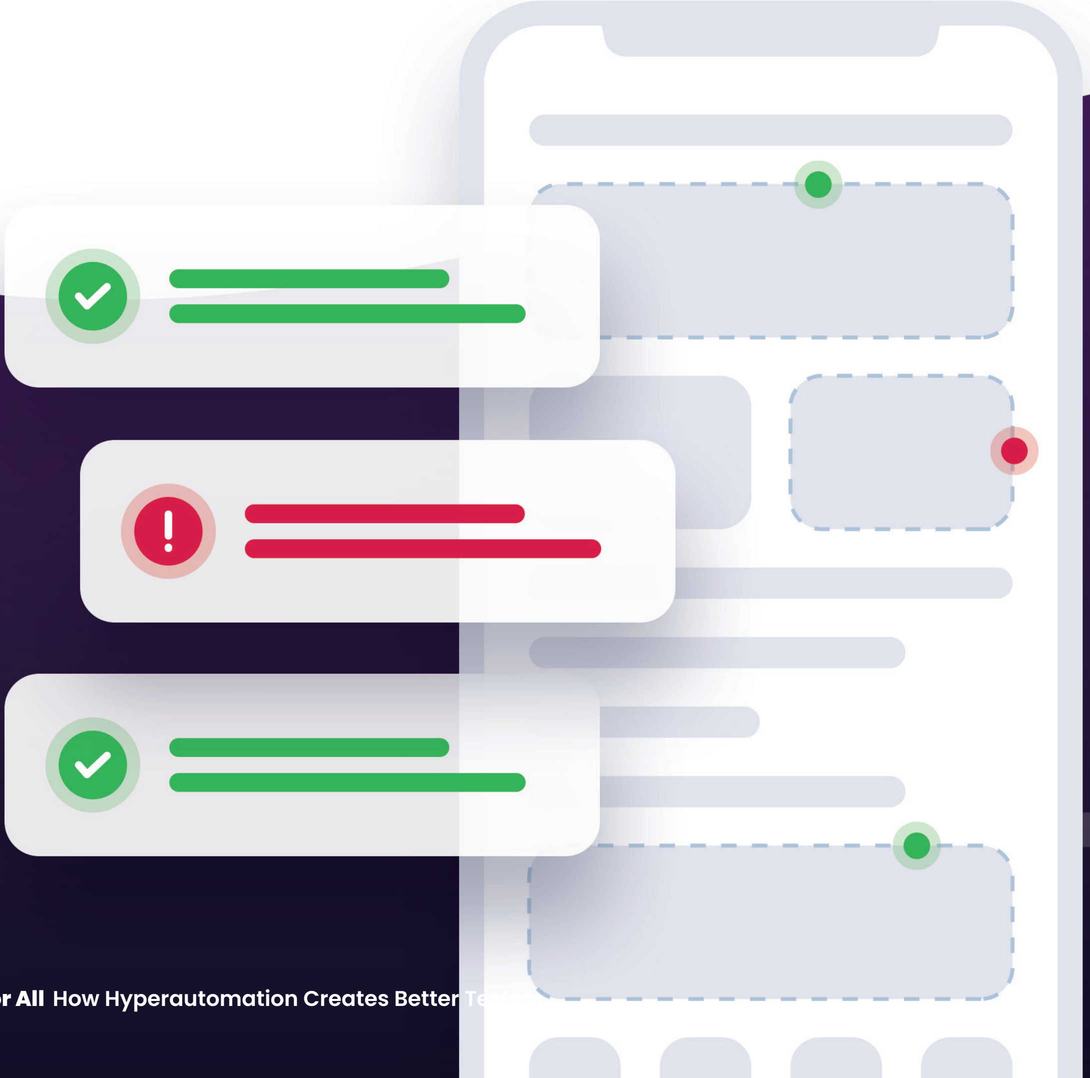
- **Self-healing automation** functions as a testing method by focusing on the automation of correcting code in select elements when the UI changes over a period of time, either within the parameters of the app or through adjustments in code as the app continues to be developed.

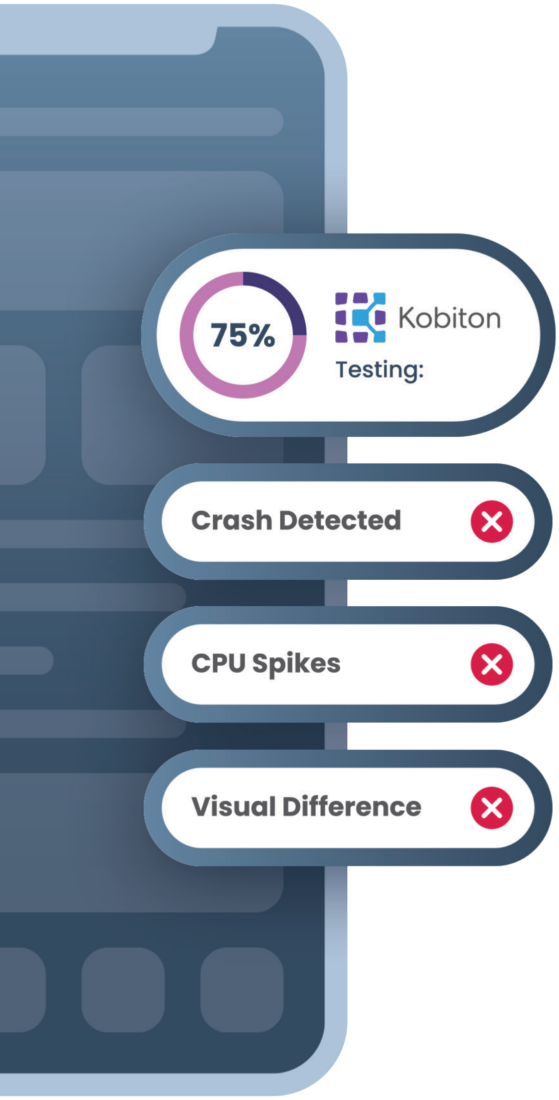
- Finally, **differential testing** serves as a key testing method by classifying differences and assessing the overall changes in an app as builds continue. From version to version, the AI absorbs what changes and learns from the tester feedback to incorporate these changes in a predictive manner.

For each methodology in a testing cycle, a number of different tools exist to assist testers in their work. These AI automated tools are designed to focus on the methodology and act according to the specifications of the facilitating tester.

Looking Ahead to the Future of AI Testing Tools

AI testing represents a tremendous jump from the previous era of manual testing in terms of computational power and quality of life improvements for testers. With that said, what innovations are on the horizon for AI testers looking ahead?





The first major evolution of artificial testing that we'll look at is automated and intelligent gap analysis. This process applies to untested new code within an application. Just as testing software is advancing, mobile apps themselves are constantly becoming increasingly complex. It's only natural that, in this process, certain application flows may go untested. This isn't the end of the world when certain flows are rarely used, but what happens when we're discussing a new application flow that's integral to the function of the app? It's highly likely that this code, if it remains untested, will contain bugs and errors that could range from the mundane to crashing an app.

What is needed here is a gap analysis. As things currently stand, to perform a gap analysis a team would need to conduct both a static analysis of every code change that has occurred up to date, as well as running a dynamic analysis of all current testing. As any tester can tell you, this can be a time-intensive course of action, and not the ideal way that a team member wants to spend their time on a project.

By harnessing artificial intelligence in testing apps, mobile testing tools in the future will learn from how consumers are using your app. If the AI notices that users are frequently interacting with a segment of the app that features a string of untested code, it will be able to alert for gaps in testing so that your team can quickly review the code and patch if necessary. They'll also be able to utilize predictive features to optimize test coverage by pointing out the areas that testing is most needed based on user usage frequency.

Another realm that future AI testing can tackle is that of automated test generation. As it stands, automated testing only handles the execution of tests. Test execution is great, but when you consider the days and weeks that it takes to write a new test for a mobile app, this is really just a drop in the bucket. Future AI testing will be able to predictively generate unit tests and API tests in a sustained fashion, so that reliable tests can be created and executed, allowing testers to free up their time to focus on more nuanced aspects of quality assurance during the testing cycle.

Conclusion

Concerns surrounding AI are understandable, and for teams to operate with a degree of healthy skepticism is only natural. However, test automation through AI has allowed development teams to save countless dollars and hours on testing their products, creating a cyclical flow of financial windfalls that boost the success of their teams, all while not skimping on the quality of testing that they continue to conduct. Automated AI testing has seen continued growth; according to a survey conducted by QASymphony and aggregated by Infoworld, among QA decision-makers at mid- to large-size firms, seventy-two percent of teams are currently automating their testing cycle. Furthermore, among organizations that utilize AI automation for fifty percent or more of their tests, an astounding eighty-eight percent of teams report faster testing cycles, with over seven out of ten teams declaring improved overall test coverage and the ability to catch errors earlier in the testing cycle. These teams further reported higher morale amongst their testing teams, and a greater sense that their testing cycle was maximizing efficiency and their time.

While automated AI and scriptless testing continue to be an emerging practice, allowing your team to sit on the sidelines of this testing transformation that is soon to reach the status of industry standard would be doing a disservice to your company's needs. This should provide a foundation for your inevitable deep dive into AI testing automation, with a variety of tools on the market to best satisfy your testing team's objectives. With an array of tools available from Kobiton and beyond, it's beyond time to consider how artificial intelligence can augment your testing routine and improve the testing cycle by leaps and bounds.

Kobiton

www.kobiton.com · info@kobiton.com