

Buyer's Guide to Mobile Test Automation



Kobiton



Table of Contents



Why you should read this guide

What is Mobile Test Automation?

Introduction

Let's define what we mean by mobile test automation

Web Automation has been around. So what's the big deal about mobile automation?

Common Approaches to Mobile Test Automation

Do you need mobile test automation?

How is Mobile Test Automation Different to Web Automation?

Let's explore the most common ways of achieving mobile test automation

The Business Case for Test Automation

Getting Successful with Test Automation

With so many choices, how do you go about buying a mobile test automation platform?

Trends in Mobile Test Automation

This isn't just about the technology

Buying Mobile Test Automation

This is an exciting field. Let's look at what lies ahead.

The Kobiton Platform

Mobile Test Automation Influencers You Should Know

About Kobiton

Follow these influencers to stay
abreast of trends



Introduction



Why should I read this buyer's guide to mobile test automation?

Organizations the world over have certain commonalities. Namely, a customer base that is becoming increasingly 'mobile centric', a desire to ship apps faster, and to do so with improved quality.

Recent global events have fueled online behavior ... a recent McKinsey study shows e-commerce penetration achieved 10 years' worth of growth in just 3 months. According to the consulting firm App Annie, at the end of the first quarter of 2020, a historical maximum of monthly consumption of mobile applications was reached, reaching more than 200 billion hours in April of this year in the world, representing an increase of 40%.

And in the face of rising usage is increased competition and a need to continually innovate. Which inevitably means releasing faster and with higher quality. In a Kobiton 2020 survey, respondents indicated that they are already releasing, or wish to release, on a *weekly* or *daily* basis. All of which means that organizations have to adopt mobile

testing and test automation if they wish to stay relevant. It's no longer a nice-to-have; it's a must-have.

This guide aims to serve as an introduction to mobile test automation and will serve as an ideal starting guide as you embark on this journey. We'll start by exploring what we mean by mobile test automation, briefly look at some of the technology and then explore the business case.



What is Mobile Test Automation?



What is Mobile Test Automation?



A good place to start our journey is to look at what mobile test automation is.



What?

Mobile Test Automation is a process for executing test cases against a mobile app or website without expending significant human effort in executing those tests.



How?

Software is utilized to perform testing, broadly defined in two categories of scripted and scriptless automation.



Why?

So that companies can release apps faster while maintaining or increasing app quality.

What is Mobile Test Automation?



Who uses mobile test automation?



You might think that test automation is exclusively for the “big guys”, but you’d be wrong.

What is Mobile Test Automation?



What type of features are in a testing platform?

Below is a representative list of what features you can expect in your platform. Don't worry, we'll explore most of these in more detail later on in this guide.

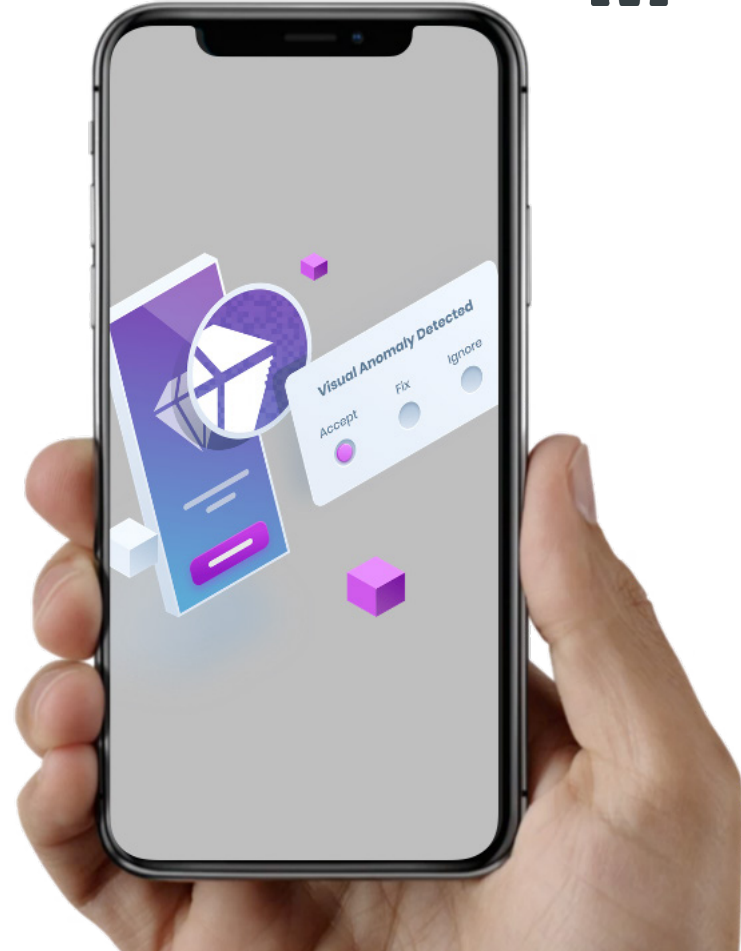
- Support for automation scripts, including Appium
- Support for scriptless automation such as Record & Playback
- Crash assertions
- Performance assertions (CPU, Memory, Battery, etc)
- Visual validation or visual testing
- Ability to test against real-devices, not emulators or simulators
- Integration into your DevOps environment
- Comprehensive management and reporting

**Mobile Test
Automation**

What Mobile Test Automation is NOT...

A solution just for testers -

Mobile test automation benefits the entire organization through faster release cycles and improved app quality. It's also useful for DevOps to implement Continuous Testing, and for Developers to quickly test their code for cross-platform issues prior to checking in their code.



What Mobile Test Automation is NOT...

A solution that gives you all the benefits without any of the effort -

Test automation, and mobile test automation, in particular, is challenging. From requiring strong expertise to having a process to update scripts as part of your change management process, test automation doesn't always come easy or cheap and is most certainly not a set-and-forget type of solution.



How is Mobile Test Automation Different to Web Automation? .



How is Mobile Automation Different to Web Automation?



Web test automation is far more mature than the more nascent mobile test automation. This maturity means there are more automation frameworks and vendor products, and it is generally far easier than its mobile counterpart. Many practitioners coming from the web world are often in for an unpleasant surprise when confronted with the added complexity in the mobile world.

Some Important Technical Bits You Should Know



Why is mobile test automation more complicated? To better understand this, it is helpful to take a small detour and look at how automation technology works conceptually. For the technical purist, we caution you that we take some liberties with our explanation here in order to convey the concept.

Automation is, at its core, attempting to simulate a human interacting with your application. Where a human would have clicked the login button, the software will click that button instead.

And just as you (presumably you're human) have to find the login button on the screen to click it, so too must the automation software. The earliest days of automation (way before mobile and web) actually tried to find objects on your screen by utilizing screen coordinates. This was fraught with issues, and most definitely isn't workable for web and mobile where the location of the button changes on every device and screen.

Some automation technology will use image recognition or optical character recognition (OCR) - this works, but also tends to be very fragile.

On the other end of the spectrum, the most robust mechanism is to give all your elements a unique internal name. For example, our login button may be given an internal name by the developers such as "buttonLogin". This allows your automation framework to scan the elements in a non-visual manner, "looking" for that button name.

Now to move on to the question of why mobile test automation is more complex than web test automation, it is helpful to understand the concept behind the document object model (DOM).

Any web page you see visually rendered in your browser is rendered from an underlying document. That is, the server sends a HTML document to your browser which reads it, creates this document object model, which your browser can read and render to you in a nice visual manner.

Technical Bits Continued...



Mobile apps actually have an underlying DOM too for rendering their UI. The mechanism works differently, but conceptually each element on your screen is represented in the underlying DOM.

Now here is where it gets interesting. The web world is far more mature. In fact, they have specifications (W3C standards) that define exactly how the DOM is to be constructed and how to render that DOM as a visual UI. These standards enable a web page to, for the most part, be viewed in any browser on any device.

The DOM is a highly structured document - the elements in the DOM will always be in the proper order and will not change across browsers.

And it is precisely this standard specification that makes automation on the web easier. The automation framework knows what to 'expect' when reading the DOM and finding elements to interact with.

The mobile world however is the wild-west. There are no specifications or standards.

Which means that if your automation software was looking for a login button in the DOM expecting it to be within a specific area of the document, it could be in a totally different area on another device.

Conceptually, imagine if I send you a Word document and I tell you you can find the author name in the header of every page. Great, now you know where to find it. Just look in the header. But let's say that when you open that same Word document in Google Docs, the information that was in the header now appears in the footer. If you were looking for it in the header, you're not going to find it.

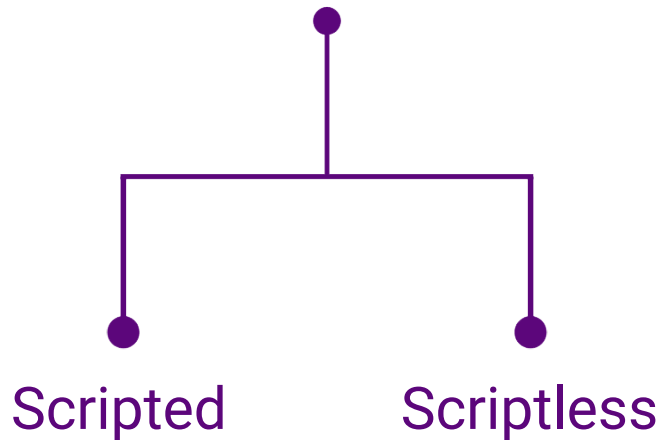
And this is the crux of the problem in the mobile world. Internal element locations in the document object model change across different device types, manufacturers and more.

This results in "brittle" or fragile test scripts. Your automation script may work wonderfully on one device, but fails on another device because it couldn't find, for example, our login button to click it.

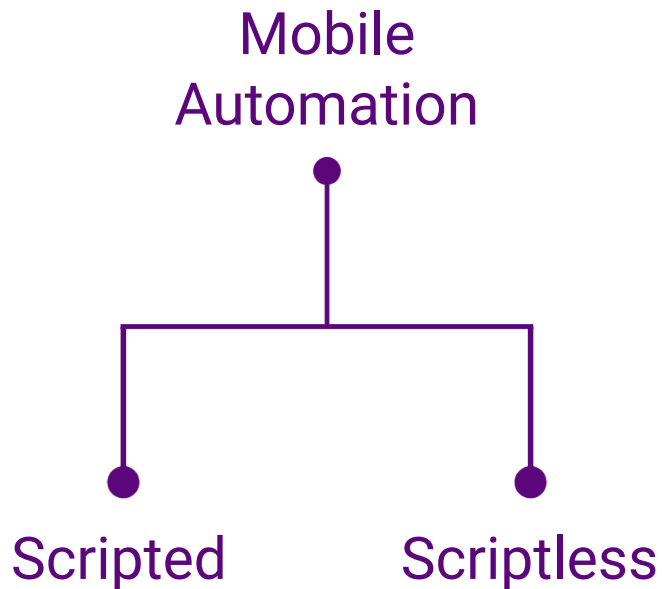
Common Approaches to Mobile Test Automation



Mobile Automation



There are two primary approaches you can take for mobile test automation.



You can code or **script** your automation tests – this is very powerful and flexible but comes at the cost of relatively complexity and requires coding skills – or you can use **scriptless** automation technology which aims to remove the need to code, but may be less flexible.

Scripted Automation



There are a number of automation frameworks on the market today, most of them open-source. They provide the core capabilities to drive automation on a device, and can typically be used from within multiple languages.

Some frameworks work across both iOS and Android, while others are unique to a specific operating system. Much like software development, you will either have one set of test scripts for all devices, or need to develop and maintain test scripts unique to each operating system.

Let's look at some of the more common frameworks in use today.

Appium

Appium is arguably the most popular cross-platform mobile scripting automation framework. Appium is the sibling of Selenium, a widely popular framework for web automation.

Its primary benefit is, of course, being cross-platform, you only need to maintain one set of scripts.

Other advantages include the fact that you can automate without recompiling the app source code, supports emulators, simulators and real-devices, and comes with a handy inspector for assisting in element identification.

XCUI

XCUI is specifically designed for iOS apps. Unlike Appium, you will not be able to leverage your XCUI tests on Android devices.

XCUI offers a number of advantages - it is part of a comprehensive offering as part of XCode and is quite feature rich, including having a test recorder. In addition, because it is native to iOS it is already using the same language you developed your app in so for many Apple developers there is instant familiarity with the toolset. Finally, many practitioners will say that it is faster than Appium as it is not an abstraction layer like Appium.

Scripted Automation



Espresso

Just like XCUITest is for iOS apps only, Espresso is for Android apps only. Similarly, it offers tight integration with the Android Studio IDE so Android developers will be able to adopt it quickly and easily.

Espresso developers often express how robust the test scripts are with its built-in validation for ensuring an object is on the screen. There are a number of automation frameworks on the market today, most of them open-source. They provide the core capabilities to drive automation on a device, and can typically be used from within multiple languages.



Scriptless Automation



Scriptless Automation aims to solve the inherent coding complexity that comes with developing test automation scripts. Just like “no-code/low-code” platforms are easing the development complexity, so too are scriptless automation products.

There has been renewed interest in this category thanks to the rise of AI and machine learning. Now that those technologies are entering a new phase of maturity, many of the previous deficiencies inherent to scriptless automation solutions are being addressed.

While there are a number of approaches to scriptless automation, we'll broadly lump them into two categories here:

Record & Playback

The concept of Record & Playback is that one conducts a manual test as you normally would, and the software records the steps taken so that they can be played back again and again. You're effectively creating an automated test from your manual test.

Early incarnations of Record & Playback solutions got a bad name ... they tended to be limited in their capabilities and the playback was often quite error prone.

However, with the advent of AI and machine learning, these technologies are now extremely viable. The playback mechanism is able to find elements under extremely challenging conditions, and rich feature sets including assertions and remediations now make this a very viable test strategy, especially for those organizations who do not have the investment in test engineers needed to develop and maintain scripts.

Scriptless Automation



Autonomous testing

Autonomous testing refers to the idea of letting the software perform testing without being given an explicit test case.

Some approaches are to simply try every option in rapid fashion (“The Monkey Test”) to see if anything ‘breaks’.

Newer solutions on the market are analyzing source code to find high-risk areas, and then attempting to construct test cases based on that analysis.



The Business Case for Test Automation

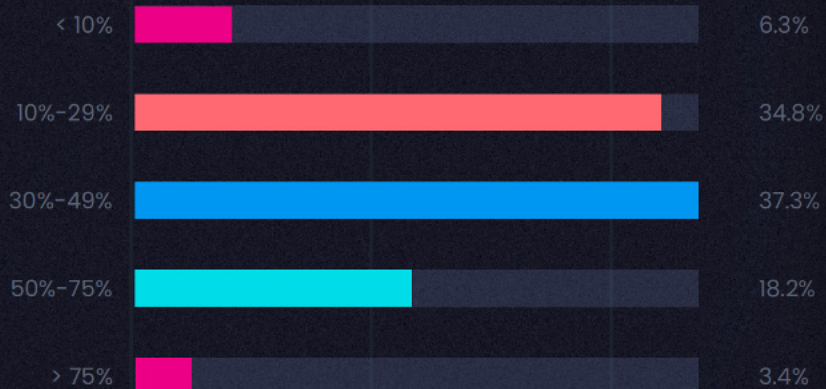


Is It Worth It?

The good news is that the business case for test automation is compelling – both from a hard dollars and cents perspective, and from an operational perspective – which we will explore first.



Based on the overall quality assurance budget in your organization, what percentage is spent on test automation when factoring in employees, tools, and other resources?



The business case is so compelling that companies are not afraid of spending on automation.

“Operational ROI”



Companies the world over are under tremendous pressure to continue to innovate, deliver faster, with better quality and at a lower cost. And while the adage of “you can pick two of those” became software development lore, it simply doesn’t hold true anymore. Organizations are at risk of being out-innovated and out-maneuvered at all times.



STATE OF TEST AUTOMATION
2020-2021 EDITION
kobiton

SURVEY

In their 2020-2021 State of Test Automation survey by Kobiton, over 51% of respondents said that they are releasing on a weekly or daily basis! Organizations doing monthly or quarterly app releases are in grave danger of falling behind their peers.



It's no longer nice-to-have

Given the need for rapid release cycles, companies really have no alternative but to embrace test automation. Over 69% of respondents indicated that they run up to 1000 manual test cases... a number that quickly becomes unsustainable when trying to release on a daily basis.

It should be clear that moving to automated testing is therefore an imperative ... its moved solidly from nice-to-have to must-have territory. It is so important to organizations that despite automation having a massive ability to lower costs, this isn't why organizations are embarking on the journey. In the same survey, only 9% of respondents cited cost-cutting as a reason for their automation journey.

Hard ROI



In this section we'll look at the savings and ROI behind using an automation solution. While you may realize it's a "must-have", it's still great to justify the spend. The ROI behind automation (scriptless in particular), is enough to make your CFO smile.

Our ROI Model Assumptions

We'll make a few assumptions for our model. The interested reader can tweak these values for personalized results with an interactive [calculator found here](#).

Assumption	Value
Number of app releases per year	12
Average number of test cases executed per release	30
Average duration of executing a manual test case:	10 minutes
Number of devices to test on (Recommend 158 to cover 90% of the market)	158
Automation time savings over Manual testing	90%
Time to code an Appium test	12 hours
Yearly fully loaded cost of a test engineer	\$110,000

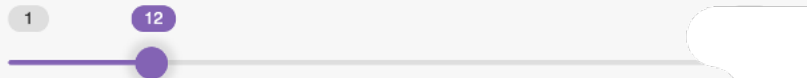


The cost of executing test cases manually in the previous scenario is **\$554,000 per year. Taking into account the licensing cost for a scriptless automation solution, you could be looking at **\$400k to \$500k yearly savings**. That's a compelling ROI.**

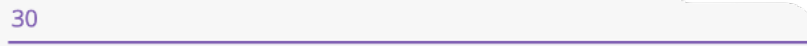
Step 1: Your environment

You can accept our default values or enter values you believe more accurately reflect your environment.

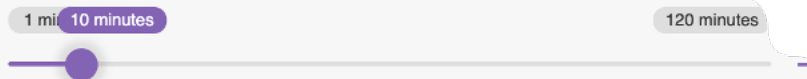
1. Number of app releases per year ?



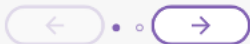
2. Average number of test cases executed per release of your app



3. Average duration of executing Manual test per test case



4. Number of devices to test on (158 recommended to cover 90% of the devices)



Calculate your ROI

Use this [online ROI calculator](#) to input your own values and gain an understanding of the ROI behind making the move to automation.

**Getting Successful
with Test
Automation -
more than
technology.**



In This Section...



Automation is a journey. It isn't a matter of just buying the latest automation technology and sitting back thinking your automation initiatives are now complete. Automation investment can be costly, implementation can be complex, and the road is fraught with pitfalls if not approached carefully.

And while there can be significant differences in various automation platforms, the success of your project is primarily going to be driven by your strategy and process, and your people. Let's explore these.

Strategy and Process

This should not be a technology led project - You need to ensure that your organization has a solid plan in place to embrace automation and to be successful. Here are just a few of the things you should be thinking through:

Measurements and Metrics	QA, Dev and DevOps alignment	Approach
<ul style="list-style-type: none">• Why are you making the move to automation?• What percentage of test cases do you want automated and by when?• How will you measure automation success?• Do you need to measure ROI?	<ul style="list-style-type: none">• How will your automation strategy work with DevOps?• Will you be implementing continuous testing?• How will test results be fed back to Developers?	<ul style="list-style-type: none">• What is your budget?• How will you make the business case to management?• Do you need a single platform or use best-of-breed solutions?• Is scripted or scriptless a better approach for you?

The People



Ultimately the people, more than the technology, are going to determine the success or failure of your automation initiatives.

In addition to aligning all the groups (see previous section), you will need to ensure that you have the right talent on-board. It is beneficial to decide your automation approach first – if you're opting for scripted automation, you need to ensure you have test engineers onboard that are well versed in the framework of your choice, such as Appium.

Finding talent in a strong economy isn't easy. Respondents in a survey listed **Training or Hiring** as their second biggest pain point in using or starting test automation.

Organizations who don't have the ability to have a team of test engineers may be well suited to look at the scriptless automation technologies which can deliver the benefits without the resulting cost overhead.



Buying Mobile Test Automation



Buying Mobile Test Automation



So you've looked at the benefits, you've looked at the ROI and you have a strategy in place and want to buy an automation solution! Great! In this section we provide some guidance on moving forward, from tool selection to implementation guidelines.

In a recent survey, QA professionals were asked "*What is your biggest struggle/pain point with using or starting test automation today?*" While one may have suspected that a lack of skilled resources would be the primary concern, in fact it was a rather surprising twist: By far the number one concern cited was "**Finding the right tools**".

There are so many options that many organizations face analysis-paralysis - a situation not helped by various claims put forth by vendors.

The first key takeaway is that technology should not drive the outcome. To paraphrase Steve Jobs, start with the outcome in mind and work backward to the technology. That said, there is a pivotal decision point that will be reached - selecting your approach: scripted or scriptless.

Deciding on your approach will help you not just "narrow the field" in your vendor selection, but provide your direction on the resourcing you will need to support your initiative.

Begin with the Business Case

One of the leading success indicators for any project is whether the project is driven by a business case or by a technology decision. Those projects built on a solid business case are overwhelmingly more likely to succeed. A good business case will also demonstrate a clear ROI and, more importantly, address the success metrics. While this business case is of course important for internal funding, it is a good practice to share your business case and success metrics with your vendor. A good vendor/customer partnership is one with a mutual understanding of the business case. Other advantages include the fact that you can automate without recompiling the app source code, supports emulators, simulators and real-devices, and comes with a handy inspector for assisting in element identification.

Buying Mobile Test Automation



Capability Prioritization

Armed with your use cases (see above), you can now map out the capabilities you need. This is important for two reasons: First, you can prioritize and roll out the most needed capabilities first. Second, knowing the full set of capabilities you will need will help with vendor selection. Ideally, most of your capabilities will be provided by one or two vendors, instead of relying on a patchwork of various technologies, and trying to integrate them all.

Don't Buy Capabilities. Buy Outcomes. Ask for Usage Based Pricing Models.

Every technology vendor will gladly sell you functionality. However, your primary concern should be about delivering outcomes and ultimately this is what you want to buy. For example, don't just buy an automation solution. Buy an automation solution that specifically enables continuous testing to cut release cycles by n%.

A vendor that is closely aligned with your business case will reflect this outcome-based approach in their proposal.

This is another reason why choosing a patchwork of various vendors is problematic - each will "pass the blame" and no single party is responsible for achieving the overall outcomes. Limit the number of technology vendors in your mobile testing stack.

Assess Skills Required to Own & Enhance

Implementing mobile automation isn't a "set and forget" implementation. In fact, a comprehensive roll-out may involve many components, including access to real-devices for increased testing, automation frameworks like Appium, and scriptless automation solutions and visual testing. Each of these systems requires a specific role or set of skills in your organization.

- Who will be responsible for ensuring your test scripts are ready for DevOps?
- Who will be responsible for building automation scripts?
- Who will monitor the performance of the solution?

Buying Mobile Test Automation



We caution customers to identify the roles and skills required before embarking on a project like this. Anything else is disingenuous - it would be a terrible and costly realization if this is only addressed after the project goes live. Further, you would ideally have these roles work side-by-side with the vendor during the implementation process to ensure adequate knowledge transfer takes place.

Prove the Value. Then Scale.

Earlier in this guide we discussed the importance of continuous improvement. This goes for the business case too: Prove out the value incrementally. Often, you can choose high-risk areas in your app to automate regression tests against those or use automation to perform a quick smoke test that you can run nightly.

By starting small and proving out the value, you also substantially mitigate the risk. Risk may manifest itself in various forms.

There is an inherent risk that a vendor's claims are, to put it bluntly, fabricated or "wishful thinking".

Starting with incremental rollouts lets you quickly put the vendor's claims to the test.

There is also a technology risk, and these should be discovered as soon as possible, so deploying early and often is a good way of catching any issues.

Finally, broad adoption is an inherent risk so being able to incrementally provide new features, collect feedback and implement suggestions is a great way of improving buy-in to your automation initiatives.



Vendor Selection

Many organizations polled state that vendor selection is often the biggest impediment to getting started with test automation. Confronted with an overwhelming array of choices, a barrage of potentially misleading messaging and dubious claims, companies often face analysis-paralysis. As we've outlined in this guide, the technology choice should be the last item to consider and vendor evaluation should only happen after you've identified your use cases and business cases.

When the time comes to look for a vendor, it is helpful to stack-rank the various vendors so that you can obtain a quantifiable score for each to create a shortlist. Use this guide below as a starting point for selecting a mobile test automation vendor:

Vendor Selection Checklist



Item	Comments	Score (10=best, 1=worst)
Comprehensive access to real devices for enhanced coverage testing	A critical component of mobile test automation is ensuring you test on real devices instead of simulators or emulators.	
Manage Device Complexity	Many organizations already have local on-premises devices that they wish to continue to use, or their use case may require that they are utilized in the future. Look for a platform that will allow management and remote access of these devices.	
Support for the major automation frameworks	Even if you have decided to do scriptless automation, your solution should support the use of the major automation frameworks, such as Appium, Espresso and XCUI. This way, you are not restricted from adding these capabilities when the situation warrants.	
Scriptless Automation	Scriptless automation has a place in every organization. Whether you use it for quick smoke tests, or it's your strategic path forward for automation, scriptless automation is a powerful capability. Select a vendor that offers both scripted and scriptless automation	
Single Platform	Although you should plan for an extensive set of capabilities and features, ideally these should come from as few vendors as possible. Preference should be given for solutions that allow you to do all of the above on a single platform.	
Outcome-based pricing	Ultimately you don't want to buy technology or features, you want to buy outcomes for your organization. Your vendor should be able to work with you to develop an outcome-based strategy and pricing model.	

Trends in Mobile Test Automation



Trends in Mobile Test Automation



At the risk of using a term that has become a buzzword, our discussion wouldn't be complete without talking about Artificial Intelligence and Machine Learning. But we don't mean AI in the sense that AI replaces the tester. Instead, AI is being used in many exciting areas to augment the tester and provide exciting new solutions.

Source Code Analysis

Here, AI analyzes your source code for areas that are high risk for errors or breakage so that your testers can concentrate on that area of the application. This is an exciting trend that should help drive quality and time-to-market by focusing on the high-risk areas.

Natural Language

In this guide we broke down test automation into two categories: scripted and scriptless. Natural Language is developing as an interesting hybrid ... giving you the relative ease of use of scriptless but more of the flexibility of scripted. Natural language lets a tester describe the test case in plain language (eg. "Click on the login button") making script development far easier.

UI/UX Best Practices

Another exciting trend is for the testing platform to analyze your Apps UI and make recommendations based on industry best practices. By way of a simple example, the platform may alert you to a font and background color combination that is hard to read.

Next Best Action

Executing 100s of test cases and managing 1000's of failures is all part of a tester's job. However, little is done by way of helping the tester to remediate issues when they do occur. Newer trends here have the testing platform recommend the next best action for the tester, ensuring they can get through the backlog in the most efficient way possible.

Trends in Mobile Test Automation

Cognitive testing

As AI engines consume larger datasets they invariably get 'smarter'.

And the premise behind this form of testing is to allow the platform to recognize the function under test and run the tests it has seen before.

For example, it isn't hard to recognize a login screen. And from there, one could deduce that you should automatically test various forms of user ID and password combinations, test the max length and min length of a username field, test the forgot password link etc.

While this form of testing will not be an immediate replacement for testing your business rules, it can serve as a great counterpart for increased coverage testing.



About Kobiton



We let organizations deliver perfect mobile experiences with every release.

Why Kobiton?



01

Lower App Abandonment & Improve CX

One-in-four apps are abandoned after first use. Ensure your app works as expected on all devices.



02

Accelerate App Delivery

Integrate real-device testing into your CI/CD processes and release at the speed of digital.



03

Tame Device Chaos

Manage public, private and local devices in on centralized administration console.



Our Heritage



About Us

Core Values

01
Performance

02
Flexibility

03
Customer
Experience

04
Affordability



60,000
Users

620+
Companies

85,654,123
Test Minutes

We love the company we keep



Rakuten

FRONTIER

Deloitte.
Digital

Office
DEPOT

Abercrombie
& Fitch



Q2



Capgemini



evertec

GreenSky™

AmerisourceBergen

RIVER ISLAND



Wolters Kluwer



What We Do

Full Lifecycle App Quality

- We drive app quality from development, to QA to DevOps
- Our platform has a feature set unmatched by anyone else
- All with the most flexible deployment options in the industry

Management

Development

Quality

DevOps

Prevent

Improve code quality before releasing to QA

- Find cross-platform issues before checking in code
- Development time testing

Perfect

Comprehensive Test Suite

- Manual Testing
- Appium Automation
- Scriptless Automation
- Assertions

Resolve

Debugging with pinpoint accuracy and efficiency

- Session Explorer with Forensics
- Remediation
- Self-Healing Tests
- Active Debugging

Release

Mobile DevOps

- Continuous Testing
- Monitoring, Analytics and Insights
- Performance
- Integration

Device Infrastructure and Management: Cloud (Public or Private) | On-Premises | Hybrid

Mobile Test Automation Influencers You Should Know



Daniel Knott



Mobile Test Automation done right, with the right setup and vendors is key to success for mobile apps. With mobile test automation, teams can deliver high-quality apps faster to their customers.

Daniel Knott, Senior Product Manager at XING

Chris Kenst



It's tempting to try the latest or most sophisticated tool. Or to ask someone else what you should do. The challenge though is no one knows your situation and problems better than you do.

Fortunately the more we learn, the more options we have available when considering how to cope with a new situation. Therefore a good approach is the one you think solves your problems now and is flexible enough to change as you and your projects do.

Chris Kenst, Automation Engineer at BloomNation and President of the Association for Software Testing

Frank Moyer



Delivering high-quality mobile apps with rapid cadence is paramount in today's mobile-driven economy. Missing a user's expectations on performance, stability, or user experience means an uninstall at best, and maybe a 1-star review. The best way to hit that rapid cadence is through test automation, across many devices and by evaluating different facets of your mobile app. Test automation done right is the only way to achieve that goal at scale.

Frank Moyer, Chief Technology Officer of Kobiton

Jeff Payne



Trends come and go. As a leader, you need to find out what is really keeping you up at night. Start building your roadmap as soon as possible.

Jeff Payne, CEO and Founder of Coveros

Thank You!



Kobiton