

**TEST AUTOMATION TODAY**

# 2016 BUYER'S GUIDE

Evaluation criteria  
to consider when selecting  
a test automation tool

# BEGINNING THE SEARCH FOR FUNCTIONAL AUTOMATION TOOLS

Functional automation is a critical part of an organization's overall delivery strategy. Creating a stable test suite with low maintenance costs will enable delivery teams to dramatically speed up their release cycles while improving overall quality.

Functional automation isn't a cure-all but when it's done properly it enables teams to lock down and quickly execute regression testing.

Automated regression checks ensure repeatability and running those checks in a parallel fashion means your teams can run them frequently to get fast feedback.

Functional automation isn't just about regression testing, however. Functional automation is extremely useful for writing acceptance tests around new feature work. It can also be a channel for close collaboration between testers and developers as the system itself is being built out.

When you employ automation tools, your team will see value beyond just faster regression/acceptance check execution. Your valuable testers will be free to do exploratory testing around your system's most critical high-value features, ensuring the things that keep stakeholders awake at night are properly covered.

## Why This Guide?

When selecting a test automation tool, you'll need to consider many factors. You'll want to measure each solution against your current and future needs and you'll need to carefully weigh these factors against open and hidden costs.

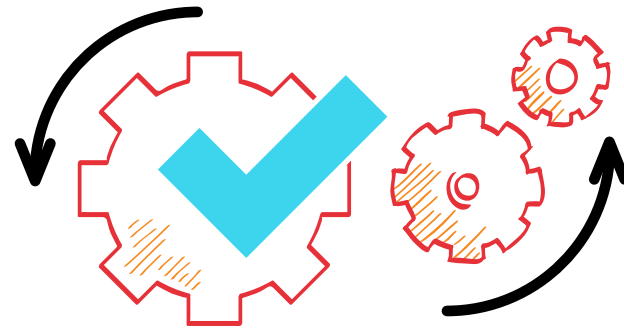
This guide provides a solid starting point for organizations at the beginning of the selection process by outlining some of the most important considerations.

**“Automated tests empower testers to focus on the system's most critical high-value features.”**

# AUTOMATION SOLUTION CATEGORIES

There's a wide spectrum of test automation tools available, from small single-purpose executable applications to complex systems that require their own significant infrastructure. Loosely gathering tools into general categories will help you focus your search and find the most appropriate set of tools for final consideration.

Here, we'll outline the four main categories of automation tools—niche/single solution, service based, mid-range, and high end.



# AUTOMATION SOLUTION CATEGORIES

## Niche/Single-Solution Tools

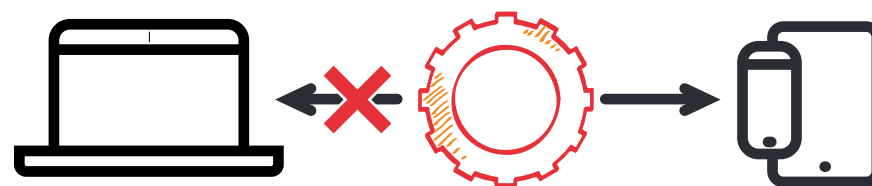
Some automation tools are narrowly focused. These niche or single-solution tools cover one specific technology such as Web, Windows Forms or Java Swing, or one particular mobile platform.

The primary disadvantage of these tools is their narrowness. You won't be able to get full coverage of a complex system or series of applications, which can be a severe hindrance or outright blocker when you're looking to test entire systems/applications.

That said, narrowness also happens to be the greatest advantage of these tools. The tool vendors/creators can focus on providing a great solution for one particular problem area. Which means they don't have to spend time building support for platforms and technologies outside of their focus. Well-done tools in these segments nail their feature set.

An additional advantage is that these tools generally require very little in the way of infrastructure. You won't need large database or application servers—although you may need infrastructure to support remote/parallel execution.

Niche and single-solution tools are generally very affordable since their feature set is smaller and tighter than tools in other categories. Normally these tools are perpetual license-based. Few of these types of tools offer subscription or on-demand licensing. There are a number of open-source tools in this category as well.



“If your project targets a single platform or browser, a niche tool can be a good fit.”

### WHEN TO CONSIDER TOOLS IN THIS CATEGORY

**Look for tools in this category when you have smaller projects that focus on a specific target audience—one browser type, one mobile platform, etc. If you have a singular or very narrow set of platform delivery targets, tools in this category may meet your needs.**

# AUTOMATION SOLUTION CATEGORIES

## Service-Based Tools

A number of successful companies have created service-based test automation tools. These tools offer device or web testing from the cloud. Most service-based tools are specific to mobile or web testing only.

Hosted services offer a number of advantages.

First, infrastructure is completely out of your hands. You don't have to worry about scheduling, execution or agents. The service (generally) takes care of all that for you.

Secondly, as with niche tools, service-based automation providers are very focused on what they provide. Moreover, upgrades and rollouts of new features are easy—the service provider handles all that for you.

Unfortunately, some organizations are hesitant about—or outright forbid—using off-premises services. Those restrictions often come into play when working in financial, healthcare or other regulated industries where various regulations control how sensitive information must be stored. Other organizations chose to avoid off-premises services based on their own policies.

Licensing for service-based tools is, of course, most frequently done on some form of subscription or on-demand basis.



“A mid-sized project that is focused on a limited number of platforms can benefit from a service-based tool, provided there are no off-premises restrictions for the organization.”

### WHEN TO CONSIDER TOOLS IN THIS CATEGORY

**Tools in this category are appropriate when you have a narrow set of target platforms to support, generally web-based. You'll also find these solutions appropriate for smaller to mid-sized projects.**

# AUTOMATION SOLUTION CATEGORIES

## Mid-Range

Organizations often require more features than the niche/single-solution sets are capable of providing. Sometimes the service-based route isn't appropriate due to the restrictions mentioned earlier.

Mid-range automation tools offer organizations a richer feature set than the previously mentioned categories without the high costs and infrastructure requirements of the high-end category.

Many tools in this category will cover several different types of testing including mobile, web, desktop, load and performance. Tools here also support multiple devices on each platform, so teams are able to get the most out of their coverage matrixes.

Tools in this category provide organizations with many critical features missing in the niche/single-solution and service-based categories. Additional features such as scheduling, reporting, remote execution, scalability and critical-test maintenance features are generally included right out of the box.

Just as importantly, many tools in this category give organizations flexibility by easily integrating with existing organizational infrastructure and services. Teams are able to choose to utilize their existing build, scheduling and database servers rather than standing up new pieces. This can be a significant advantage for organizations looking to standardize on particular infrastructure services wherever possible.

Some tools in this range can also facilitate (not force, create, or solve) collaboration between roles in the team. Employees can work out requirements, specifications or user stories between roles. For example, developers can assist

“When looking to cover a wide range of testing types—platforms and browsers—yet save on high upfront costs and infrastructure investments, a mid-range testing tool is worth considering.”

testers in creating test cases and testers can consume support APIs built by developers.

These tools are generally appropriate for small, medium and even large teams. Teams of hundreds may find themselves constrained by scalability or usability restrictions—high-end tools are more appropriate for teams of this size.

The mid-range category spans both open-source and commercial tools. Commercial tools in this category have both subscription and perpetual license strategies.

### WHEN TO CONSIDER TOOLS IN THIS CATEGORY

**This category is appropriate for many small, medium, and large teams working on a varied set of projects or projects with a wide range of delivery platforms. Tools in this category will help teams successfully deliver small, medium and complex projects across mobile, desktop and web-based platforms.**

# AUTOMATION SOLUTION CATEGORIES

## High End

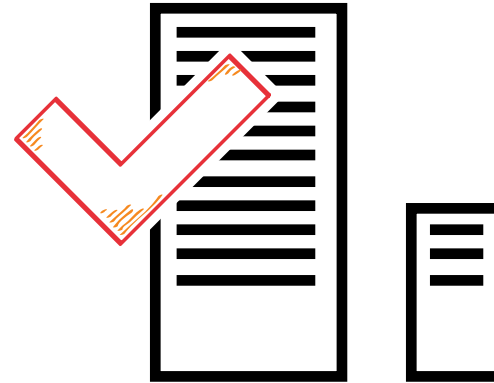
Teams that number in the hundreds or those working on projects integrating large-scale commercial systems often have significantly higher automation requirements than other teams. These teams regularly have many people on distributed teams working on the same automation projects. Projects in these environments often have testing artifact repositories in the hundreds of gigabytes, with data histories running back ten or more years.

Moreover, these projects often center around integrating with complex enterprise systems like Oracle Applications or SAP.

Organizations operating at these levels need to look to high-end automation tools to meet their scalability and system integration needs.

High-end tools generally have significant infrastructure requirements including dedicated servers for building, scheduling and databases. These tools often can't be easily integrated into existing services/servers and they often require dedicated administrators to handle tool setup, configuration and maintenance.

These high-end tools usually offer support for enterprise systems such as Oracle Applications, SAP, etc., however, that support is often offered in the form of fee-based add-ons. Additionally, these high-end tools are normally quite extensible. Large organizations often have dedicated teams extending and customizing these automation tools to fit the needs of the organization.

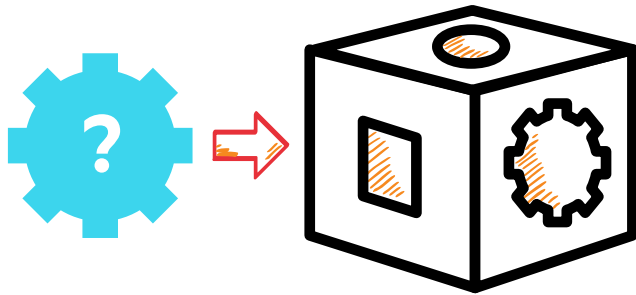


“Organizations that require integration with complex enterprise systems, have huge testing repositories and do not have budget constraints should look into high-end tools.”

### WHEN TO CONSIDER TOOLS IN THIS CATEGORY

**Teams building top-end enterprise applications using vendor-specific platforms (Oracle Applications, SAP, etc.) will need to look to this category for test-automation solutions. Additionally, tools in this category are suitable for teams with hundreds of members or projects that have been running for years and have hundreds of gigabytes of test artifacts.**

# EVALUATION CRITERIA



Once you've narrowed down the category, you'll be able to start evaluating individual tools. Evaluating automation tools based on the following will help ensure the toolset you select meets your organization's needs.

- Suitability
- Information accessibility
- Flexibility and extensibility
- Maintainability
- Ease of adoption
- Pricing

## Suitability

Not every tool will meet the needs of your organization. Make sure that the tool you select will be suitable for your needs. Ensure the tool is compatible with your system's existing technology stack. Note that automation tools don't need to be on the same exact stack—they just need to be able to work together. There's no reason that a Ruby-based tool can't be used to test .NET or Java applications.

As you evaluate the suitability of a tool, ask the following questions:

- Is the tool future-proof? Will it meet your needs as your organization grows and projects adapt?
- Does the tool vendor—commercial or open source—have a history of staying up to date with the latest technologies such as browsers and devices, etc.?

“Is this a future-proof solution? Is the vendor reliable?”



# EVALUATION CRITERIA

## Information Accessibility

Testing is meant to provide information to your projects' stakeholders so they can determine whether to ship or not. Any test automation tool you choose needs to provide the right sort of information to help your stakeholders make smart decisions.

Ask the following questions to examine how information-accessible the toolset will be:

- Do the reporting facets of your tool give you the right information?
- Is it easy to understand how defect and failure rates are trending?
- How is coverage and quality looking on the features you're currently working on?
- Are your old features stable and solid or are you seeing regressions pop up?

Having an automation tool that's great for testers and the delivery team is one thing, but you have to keep the business side of the house in mind when selecting your tool. (They're likely the ones writing the checks, anyway!)



“All stakeholders should have visibility into the quality status of the project.”

# EVALUATION CRITERIA

## Flexibility and Extensibility

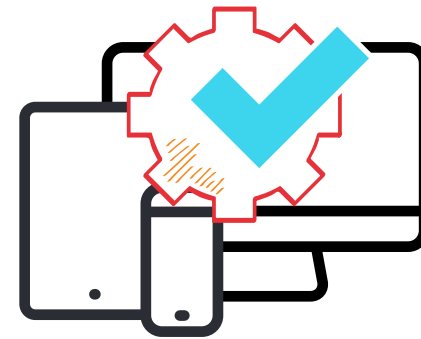
No tool can fit an organization's every need. The key to getting the most value out of your automation tool is to understand how you can tweak it to fit your needs.

Flexibility and extensibility are critical for the long-term success of your automation efforts. You don't want to go overboard with a one-size-fits-all tool but you do want to make sure the tool you choose is flexible enough to handle needs across your organization.

The following questions are key to understanding how flexible the toolset will be:

- Will the tool work for more than just one project?
- Will the tool work with different workflows used by different teams?
- How much time will it take to integrate the tool into other project infrastructures?
- Will the tool work with your existing infrastructure, or will you need to buy or create new servers and systems?

Can you extend the tool with custom libraries to fit your system APIs and

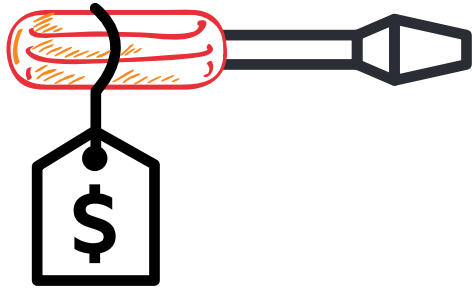


controls? Can you tweak how the tool handles building or running test scripts?

Does the tool have APIs you can access to build your own reports or mine the results database?

**“Find out whether or not you can tweak the tool to fit your needs, your existing processes and your infrastructure—and how.”**

# EVALUATION CRITERIA



## Maintainability

As with the software systems, you have to build your automation suites with an eye on long-term maintenance costs. Choosing a tool that helps you keep maintenance costs under control is critical. You don't want your teams spending more time adjusting and fixing tests than actually testing.

Avoiding duplication is critical for long-term success. For example, the ability to store element locators in a single spot is crucial because it will prevent wasted work when your UI changes. The ability to modularize tests and leverage utility libraries will enable you to quickly reuse functionality without having to rewrite it. If the tool facilitates collaboration between testers and developers, your teams will fend off brittle automation and handle changes and enhancements to your test suites more quickly.

Avoid solutions that force an all-coded approach. Productivity will suffer if your automation efforts become another code base to maintain. Just like any software project, they will require the attention of people with more experienced skillsets to troubleshoot, update and ultimately maintain.

“The top three benefits of selecting a non-brittle automation tool are the ability to avoid duplication, the ability to modularize tests and the ability to facilitate tester-developer collaboration.”

# EVALUATION CRITERIA

## Ease of Adoption

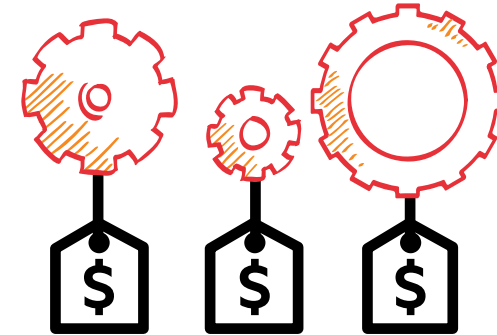
Every new tool will cause some amount of churn during adoption. It's critical to understand what that churn will look like.

Test automation is an incredibly challenging domain, particularly in the functional user-interface domain.

Ask the following questions to get a clear idea of what the adoption process will be like:

- How long it will take your team to become initially effective with the tool?
- How long it will take to achieve mastery?
- Does your team have the skills necessary to start using the tool?
- How much development skill is required?
- How much knowledge of the underlying system will your team need in order to use the tool?
- What sort of training is available for the tool?
- How can you get support once you're underway with your project?

“What's the learning curve associated with the tool?  
Are there training and support resources available?”



## Pricing

Cost is an obvious consideration. Even open-source tools have adoption costs.

- Does the tool offer varied pricing options to give you the best set of alternatives?
- What aspects of the tool cost extra?
- Will you need to allocate part of your budget to cover add-ons and additional feature sets?
- Is there a subscription model that fits your needs?

“Upfront cost should not be  
your sole evaluation criteria.”

# SAMPLE EVALUATION CHECKLIST

This checklist will give you a starting point for evaluating automation tools. It also demonstrates how Telerik Test Studio responds to the key evaluation questions we posed in the previous section of this guide.

Category	Telerik Test Studio
Suitability	
Mobile test automation	iOS, Android
Web testing	HTML, HTML5, MVC, AJAX
Multi-browser support	Chrome, Safari, Firefox, Internet Explorer
Cross-browser recording and playback	Yes. Out-of-the-box.
Desktop application support	Windows Presentation Foundation
XAML application support	Yes.
iFrame support	Yes.
Password encryption support	Yes.
System language platform support	Irrelevant. Test Studio operates at UI level.
Performance testing	Yes. Out-of-the-box.
Load testing	Yes. Out-of-the-box.
Exploratory testing	Yes. Via built-in plugin.
Manual testing	Yes.

# SAMPLE EVALUATION CHECKLIST

Category	Telerik Test Studio
Data-driven testing	Yes. Completely codeless.
Dialogue handling	Yes.
Image comparison	Yes.
JavaScript support	Yes. Easily invoke JavaScript commands.
Dynamic Test Lists	Yes.
Drag-n-drop actions	Yes. Easily build complex Drag and Drop actions.
Set breakpoints	Yes.
Desktop commands	Yes. Easily Invoke Desktop commands such as right-click and double-click.
<b>Information Accessibility</b>	Yes.
Reports show critical quality trends	Yes. Different reports available.
Integration with Test Case Management systems	Yes. Team Foundation Server and TeamPulse out of the box. Others via plugin.
Export to other formats	Excel, HTML, image
Ease of report distribution and sharing	Yes. Visual storyboard test view with ability to export as Test Case Documentation
Integrate with other reporting tools	Export to HP Quality Center and Team Foundation Server.
Bug tracking systems integration	Yes. Out-of-the-box with TFS, TeamPulse and more via APIs.
Project management system integration	Binding of Acceptance Criteria with TeamPulse integration.
Email notification service	Yes.

# SAMPLE EVALUATION CHECKLIST

## Category

## Telerik Test Studio

### Flexibility and Extensibility

Scheduling server	Out of the box.
Execution agents	Out of the box. Test Studio Run-time Edition is required.
Build/CI server integration	Integrate Test Studio with any Windows-based server. Test Studio Mobile integrates with Windows, Linux and Mac CI systems.
Source control systems integration	Yes. Out-of-the-box.
Extensible reports	Yes.
Extensible for custom UI controls	Yes.
APIs for interacting with reporting and execution servers	Yes.
Tests reuse	Codeless text extraction and reuse.
Extend recorded tests with code	Yes. C# and VB.NET are supported.
Nested elements	Rich element find logic including nested logic for problematic dynamic IDs.

### Maintainability

No duplication of critical assets	Yes. Element repository centrally stores locators.
Facilitates collaboration between team members	Yes. Testers focus on building tests, developers focus on extending tests with code where needed.
Built-in DOM explorer	Yes.

# SAMPLE EVALUATION CHECKLIST

Category	Telerik Test Studio
<b>Ease of Adoption</b>	
Requires in depth coding skills	No. Leverage code where needed.
Supports testers of all skills	Yes.
Easy to use UI	Yes.
Extensive documentation and guidance	Professional online documentation, Getting Started resources, numerous white papers
Adoption training	Online training, onsite training, personalized consultations.
Implementation assistance	Yes, via Telerik Services branch and Telerik Partners.
Ease of test creation	Create load test scenarios directly from functional tests.
Ease of test creation	Create multiple verifications at once.
Ease of test creation	Built-in scroll actions.
Ease of test creation	Simple keyword driven step creation via 'Step Builder'.
Conditional logic	Yes. Completely codeless.
Familiar user environment	Yes. Testers use standalone QA app while Developers work with the Visual Studio plugin.
<b>Pricing</b>	
Subscription licensing	Yes.
Flexible per-node licensing	Yes.
Support package	Yes. Industry leading support. 24-hour response time.
Public feedback portal	Yes.



# SAMPLE EVALUATION CHECKLIST

## Category

## Evaluation Checklist: Tool 2

Suitability	
Mobile test automation	
Web testing	
Multi-browser support	
Cross-browser recording and playback	
Desktop application support	
XAML application support	
iFrame support	
Password encryption support	
System language platform support	
Performance testing	
Load testing	
Exploratory testing	
Manual testing	

# SAMPLE EVALUATION CHECKLIST

## Category

## Evaluation Checklist: Tool 2

Data-driven testing	
Dialogue handling	
Image comparison	
JavaScript support	
Dynamic Test Lists	
Drag-n-drop actions	
Set breakpoints	
Desktop commands	
<b>Information Accessibility</b>	
Reports show critical quality trends	
Integration with Test Case Management systems	
Export to other formats	
Ease of report distribution and sharing	
Integrate with other reporting tools	
Bug tracking systems integration	
Project management system integration	
Email notification service	

# SAMPLE EVALUATION CHECKLIST

## Category

## Evaluation Checklist: Tool 2

### Flexibility and Extensibility

Scheduling server	
Execution agents	
Build/CI server integration	
Source control systems integration	
Extensible reports	
Extensible for custom UI controls	
APIs for interacting with reporting and execution servers	
Tests reuse	
Extend recorded tests with code	
Nested elements	

### Maintainability

No duplication of critical assets	
Facilitates collaboration between team members	
Built-in DOM explorer	

# SAMPLE EVALUATION CHECKLIST

## Category

## Evaluation Checklist: Tool 2

### Ease of Adoption

Requires in depth coding skills

Supports testers of all skills

Easy to use UI

Extensive documentation and guidance

Adoption training

Implementation assistance

Ease of test creation

Ease of test creation

Ease of test creation

Ease of test creation

Conditional logic

Familiar user environment

### Pricing

Subscription licensing

Flexible per-node licensing

Support package

Public feedback portal

# ABOUT TELERIK TEST STUDIO

Telerik Test Studio is a powerful, reliable test automation solution that helps you create maintainable test suites for web, desktop and mobile applications. It inspires testers and developers to collaborate on building high-value test automation and increase team velocity.



I had to make a tough decision to dedicate time when I had virtually no time and projects / priorities being tossed at me seemingly non-stop. I decided to carve out a small amount of time to discuss our needs with Telerik's ALM consultants who took me on a brief tour. In a matter of hours, I could see the light at the end of the tunnel and today our product quality had improved tremendously, our team is more organized and I embrace projects as they come my way, knowing we have a system that works."

Jeff Freeman  
Senior Consultant, NAV Canada



## Test Studio Wins Gold at the ATI Automation Honors

ATI Winner! It's an honor for us to have Test Studio recognized as the best commercial functional automated testing tool in the .NET category in the 5th Annual ATI Automation Awards.



## Telerik named a Visionary in 2014 Gartner Magic Quadrant

Gartner recently released its Magic Quadrant for Integrated Software Quality Suites report. We couldn't be more excited about being named a "visionary" in the quadrant.



## Telerik Test Studio

Functional, performance and load testing of web, desktop and mobile apps.