

net solutions

Test Automation Checklist





POINT 1

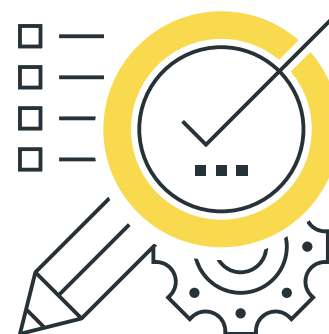
Introduction

Automated testing will shorten development cycles, avoid cumbersome repetitive tasks and improve software quality.

POINT 2

Pre-Assessment

- Is your testing mix more than 90% manual?
- Has the testing activity become deskilled?
- Are people dissatisfied with testing?
- Is there frustration by the length of time it takes to finish projects?
- Is there a need to hire additional testing resources, but no budget?
- Has the manual testing activity become tedious and repetitive?



POINT 3

Automation Feasibility Self Assessment

- Is the test executed more than once?
- Does the test cover an often used feature path?
- Does the test cover a high risk area?



- Is the test impossible or prohibitively expensive to perform manually, such as concurrency or performance-related testing?
- Are there timing-critical components and dependencies that are a must to automate?
- Does the test cover a complex or error prone feature or part of the code?
- Does the test require many data combinations using the same steps for thorough feature testing?
- Are the expected results constant?
- Is the test result analysis time-consuming, such as hundreds of outputs?
- Does the test need to be verified on multiple software and hardware configurations?
- Does the product being tested is considered "mature" or has certain level of stability?





- Is executing manual testing considered difficult?
- Does the testing includes smoke tests and repetitive verification of the same requirement in multiple places?
- Does the testing includes "Monkey Tests" that utilize large amounts or long sequences of data, transactions, or other inputs at a system in a random search for errors?
- Does the testing include Date and Time handling?
- Is the functionality testing currently manual (and stable)?
- Does the testing include load testing and traversing large amounts of paths through an application?
- Does the testing involve addressing risks related to concurrency problems, error handling, and feature interactions?
- Does the testing include functional regression and confirmation? (Rerunning a test against a new release to ensure that behavior remains unbroken—or to confirm a fix did indeed fix the underlying problem)

POINT 4

Select the Right Tool

- Consider the application and the technology involved. How was the application built? What is the user experience like?
- Think about testing requirements. Does the application have a very complex workflow?
- Determine license cost of the tool. What costs are associated with each framework?
- Evaluate the skill sets available within your organization. What skills does your team already have? Is there a team that could plug into one of the frameworks?



- Which level of software testing is within our scope? (Unit testing, functional testing, performance testing, load testing – what else?)
- Where will the software testing be used? (Front-end/back-end/API)
- Which devices are within the product scope? (laptops, tablets, smartphones, what else?)
- Which browsers need to be supported?
- How many concurrent and real users do we have, or expect to have?
- What technologies and third-party dependencies need to be integrated?
- What is the ideal reporting output?
- What is the level of community support? Does the community actively contribute to the tool and documentation?





POINT 5

Test Tool Selection

Based on the requirement, choose the best automation testing tool. Here's a comparison of top automation testing tools:

Features	Katalon Studio	Selenium	UFT	TestComplete
Test Development Platform	Cross-platform	Cross-platform	Windows	Windows
Application Under Test	Windows Desktop, Web, Mobile Apps, API/Web Services	Web Apps	Windows Desktop, Web, Mobile Apps, API/Web Services	Windows Desktop, Web, Mobile Apps, API/Web Services
Scripting Languages	Java/Groovy	Java, C#, Perl, Python, JavaScript, Ruby, PHP	VBScript	JavaScript, Python, VBScript, JScript, Delphi, C++ And C#
Programming Skills	Not required. Recommended for advanced test scripts	Advanced Skills Needed To Integrate Various Tools	Not Required. Recommended For Advanced Test Scripts	Not required. Recommended for advanced test scripts
Learning Curves	Medium	High	Medium	Medium
Ease of Installation and use	Easy to set up and run	Require installing and integrating various tools	Easy to set up and run	Easy to set up and run
Script Creation Time	Quick	Slow	Quick	Quick
Object Storage and Maintenance	Built-in object repository, XPath, object re-identification	XPath, UI Maps	Built-in object repository, smart object detection and correction	Built-in object repository, detecting common objects
Image-based Testing	Built-in Support	Require installing additional libraries	Built-in support, image-based object recognition	Built-in support
DevOps/ALM Integrations	Many	No (require additional libraries)	Many	Many
Continuous Integrations	Popular CI tools (e.g. Jenkins, Teamcity)	Various CI tools (e.g. Jenkins, Cruise Control)	Various CI tools (e.g. Jenkins, HP Quality Center)	Various CI tools (e.g. Jenkins, HP Quality Center)
Test Analytics	Katalon TestOps	No	No	No
Product Support	Community, Business support service, Dedicated staff	Open-source Community	Dedicated Staff, Community	Dedicated Staff, Community
License Type	Proprietary	Open-source (Apache 2.0)	Proprietary	Proprietary



Features	Katalon Studio	Selenium	UFT	TestComplete
Cost	Freemium Enterprise editions is \$839 /license/year	Free	License and maintenance fees	License and maintenance fees
Strengths	<ol style="list-style-type: none"> 1. No licensing and maintenance fees required (paid dedicated support services are available if needed). 2. Integrating necessary frameworks and features for quick test cases creation and execution. 3. Built on top of the Selenium framework but eliminating the need for advanced programming skills required for Selenium. 	<ol style="list-style-type: none"> 1. Open source, no licensing and maintenance fees. 2. Large and active development and user community to keep pace with software technologies. 3. Open for integration with other tools and frameworks to enhance its capability. 	<ol style="list-style-type: none"> 1. Mature, comprehensive automated testing features integrated into a single system. 2. Dedicated user support plus an established large user community. 3. Requiring only basic programming skills to get started with test creation and execution. 	<ol style="list-style-type: none"> 1. Mature, comprehensive automated testing features integrated into a single system. 2. Many scripting languages to choose from. 3. Only basic programming skills needed.
Limitations	<ol style="list-style-type: none"> 1. Emerging solution with a quickly growing community. 2. Feature set is still evolving. 3. Lack of choices for scripting languages: only Java/Groovy is supported. 	<ol style="list-style-type: none"> 1. Testing teams need to have good programming skills and experience to set up and integrate Selenium with other tools and frameworks. 2. New teams need to invest time upfront for setup and integration. 3. Slow support from the community. 	<ol style="list-style-type: none"> 1. Costly solution: license and maintenance fees are considerably high. 2. Possible high costs for upgrades and additional modules. 3. Supporting only VBScript. 	<ol style="list-style-type: none"> 1. Like UFT, considerable licensing and maintenance fees are needed for TestComplete. 2. Additional fees for extra modules and add-ons.



POINT 6

Questions To Ask To Optimize Test Automation Backlog (Without Sacrificing Coverage)

- Are you using your test management tool for transparency, collaboration, benchmarking and traceability?
- Does your operations and processes explicitly support automated test suites as the final stage upon building, deploying and distributing new versions of your software?
- Does your test design support automation efforts? (For example, does your team make sure there are test cases that verify the flow and dialogs of the UI?)
- Do you list and sort your test cases in priority order? (Are you automating sanity tests or fragile features first? Are you following a standard convention for prioritizing test automation?)
- Do you classify each test by risk, how expensive it is to do manually versus to automate?
- Do you have one dedicated team member who manages and prioritizes the test automation backlog?
- Do you have a clear plan for automation execution?
- Is your test automation framework modular and maintainable? (Do you use shared object repositories, common libraries, reusable actions, etc.?) Get the complete overview of modular and maintainable automation frameworks.





- Is your team improving test automation coverage a little bit each sprint, starting from the highest priority?
- Are you able to keep up with test script creation and framework enhancements to meet your automation goals?
- Are both developers and testers trained to write scripts during down time?
- Is there an open feedback loop between team members? (Do your automation engineers give feedback to manual testers and vice versa?)





Contact Us

Net Solutions is a strategic design & build consultancy that unites creative design thinking with agile software development under one expert roof. Founded in 2000, we create award-winning transformative digital products & platforms for startups and enterprises worldwide.

To discuss our services, call us on any of these numbers, or email us on info@nestolutions.com



LOS ANGELES

11601 Wilshire Blvd, West
Los Angeles, CA 90025,
USA



NEW YORK

101 6th Ave, 8th floor,
New York, NY 10013,
USA



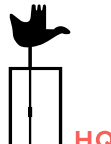
TORONTO

111 Queen St E #450,
Toronto, ON M5C 1S2,
Canada



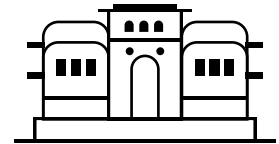
LONDON

Aldgate Tower,
2 Leman Street,
London E1 8FA,
UK



CHANDIGARH

Site No. 15, Phase 1,
Chandigarh Technology Park,
Chandigarh, U.T. 160101,
India



PUNE

Pride Purple Square,
B 315-316, Wakad, Pune,
Maharashtra 411057,
India

