QATestLab
quality is a rule

TEST IT
software testing system

Comparing Popular Test Management System / Test Automation Tool Combinations

FOREWORD

The primary objective of tech innovations is to make our lives easier. Test automation is no exception. It brings new levels of speed and flexibility to quality assurance.

According to the World Quality Report 2020-2021, automation momentum has risen throughout the QA lifecycle. Almost 70% of respondents say they use automation tools, but only half of them confirmed that existing tools had achieved the desired level of stability. Even fewer companies got ROI from their automation efforts.

How to make sure the investment will pay off?

We at QATestLab and Test IT are passionate about discovering new and efficient ways to help software developers take their QA effort to the next level. With qualified experts and well-written requirements, any team can benefit from test automation — even if it used to be a big rat hole down which to pour your time and budget.

As you learn the pros and cons of some of the most popular test management system/test automation tool combinations we have selected, think about how to apply this knowledge to reimagine the QA approach at your organization — leading the change towards software testing of enhanced accuracy and unprecedented speed.

How to make sure the investment will pay off?

CONTENTS

INTRODUCTION

In this paper, we are going to talk about how to select the right test management system/test automation tool combination for a particular project, which is critical to make your test automation effort a success.

The selected combinations are compared according to the following criteria:

- Intuitiveness

- Functionality

- Resource Intensity

- Documentation Availability

The paper embodies the experience of many people who have suffered through trial and error to help software developers deliver their products faster, with enhanced quality and with less effort. Case studies, reflecting our experiences, are also included.

TEST MANAGEMENT SYSTEM / TEST AUTOMATION TOOL COMBINATIONS

TESTRAIL + RANOREX

RANOREX allows you to create no-code automated tests and get real-time reports in **TESTRAIL**, which is a test case management solution. Combined, they provide an integrated test case management environment in its full functional capacity.

The Ranorex Studio all-in-one test automation solution enables testing any desktop, web or mobile technology, whether it be a legacy desktop application or a native app for the latest mobile devices.

QATestLab
quality is a rule

TEST IT
software testing system

TESTRAIL + RANOREX

Making the most of resources when combining Ranorex Studio's test automation with TestRail test case management will dramatically improve the quality of your software. What's more, one of the advantages of Ranorex is an informative built-in reporter with the ability to save the testing process in video format, which significantly facilitates the analysis of tests.

TestRail is integrated into Ranorex studio by default and its activation in this environment will not be difficult.

Use automated data-driven testing to cover your corner, edge, and boundary cases. You can provide the high-speed feedback you need for DevOps. Ranorex Studio's tech support, object recognition and built-in recommendations help to guarantee the stability and reliability of tests even when the application under test changes.

Desktop testing
Ranorex Studio supports a broad array of desktop technologies, from ones builton Java or .NET to those that rely on 3rd party controls like Telerik, DevExpress or Infragistics. Perfom end-to-end testing of ERP systems such as SAP or Salesforce, or hybrid applications that use Chromium-based frameworks CEF, Electron or QT WebEngine.

Web testing
With Selenium WebDriver built in to the Ranorex core API, it`s easy to build web application tests; then distribute them on a Selenium Grid for cross-browser testing on Windows, macOS and Linux. Ranorex Studio supports non-HTML5 elements such as PDF documents that Selenium WebDriver doesn`t.

Mobile testing
Test native, web and hybrid mobile apps on real Android or iOS devices or emulators. Mimic user interactions such as tap and swipe gestures, and mock services such as the GPS location. Rooting is not necessary: just use the Ranorex instrumentation wizard to instrument and deploy your mobile app using a Wi-Fi or USB connection.

IN TOTAL:

Intuitiveness — 3\5

Functionality — 3\5

Resource Intensity — 3\5

Documentation Availability — video demos and documentation provided
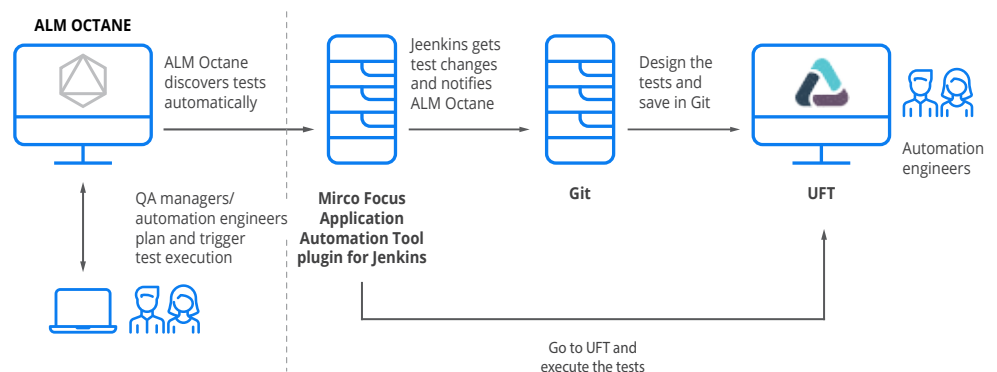
ALM OCTANE + UFT

ALM Octane and UFT are usually integrated via Jenkins or Bamboo. This integration enables ALM Octane to reflect the UFT tests from a Source Code Management (SCM) repository as executable automated tests. You can then include and run these UFT tests in test suites.

After setting up the UFT integration, the CI server and the SCM system become transparent, so you can directly work with UFT and ALM Octane:

- Create and edit tests and their data tables in UFT and save them in a Git or SVN repository.

- Run the tests and track their results in ALM Octane.

The image below shows the architecture of this integration (with Jenkins):
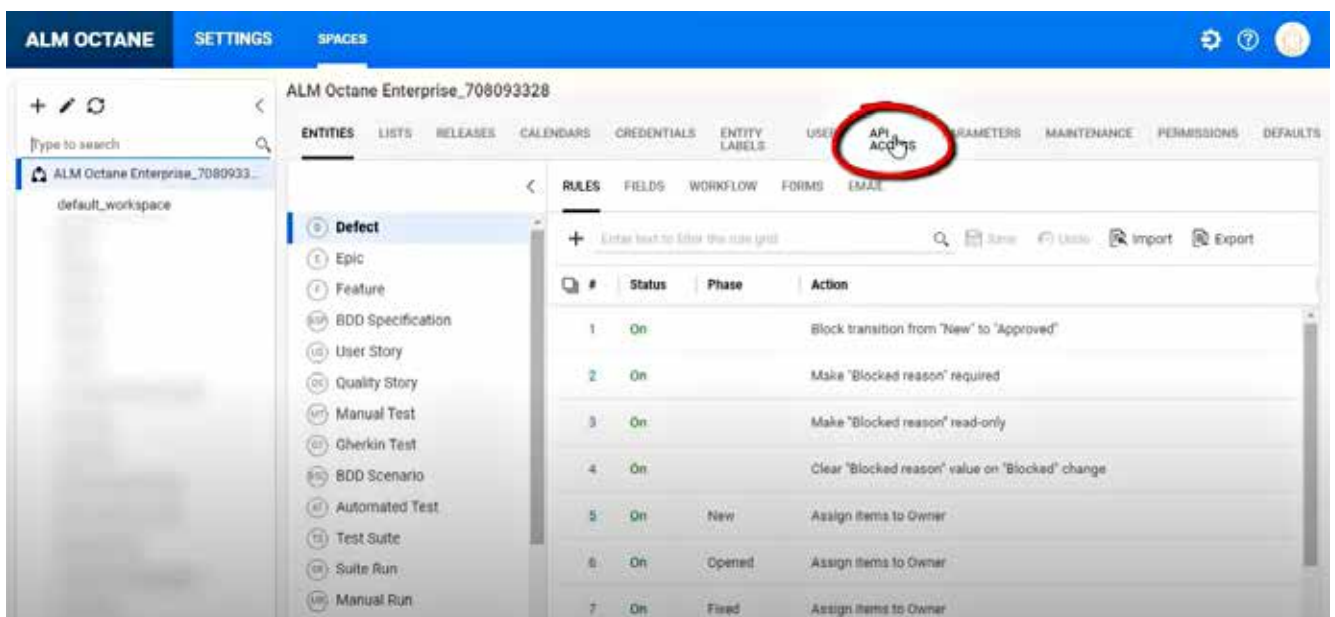
ALM OCTANE + UFT

The integration flow includes the following:

To set up the integration, create a CI server and a testing tool connection in the ALM Octane settings.

ALM Octane discovers UFT One tests and data tables. After that, it creates automated test entities to represent the GUI and API UFT tests stored in your repository – it also periodically checks for changes in the repository.

Associate the tests with your backlog and application modules. This helps you use the test run results in ALM Octane to track your product quality.
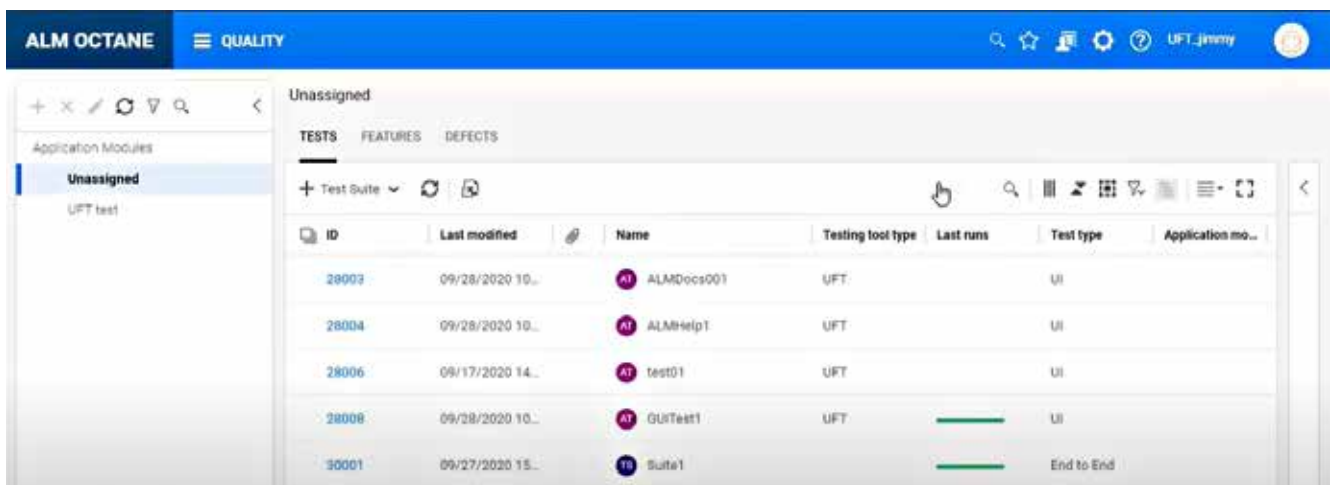
ALM OCTANE + UFT

Include the tests in test suites to plan and run them from ALM Octane.

ALM Octane triggers the test runs via the CI server. The tests run on UFT machines configured as Jenkins execution nodes.

IN TOTAL:

Intuitiveness — 3\5

Functionality — 4\5

Resource Intensity — 5\5

Documentation Availability — manuals, videos and demos provided

ZEPHYR + CUCUMBER

**COMBINE THE POWER OF CUCUMBER BDD
AND ZEPHYR FOR JIRA**

Define your software in user-centric feature
stories, robust scenarios, and integrate with
automation easily.

—

Create shared understanding between testers,
project managers and developers using
Gherkin Syntax.

—

Turn Jira Stories inro BDD Features and Zephyr
Tests into BDD Scenarios.

Zephyr itself is a powerful tool for QA professionals — when combined with Cucumber, it significantly facilitates the test automation process.

Let's analyze the process of enabling/disabling BDD in projects and creating autotests in Zephyr.

ENABLING/DISABLING BDD

Previously, BDD support in Zephyr for Jira required the Cucumber add-on for Jira to be installed. From August 2020, this is no longer necessary.
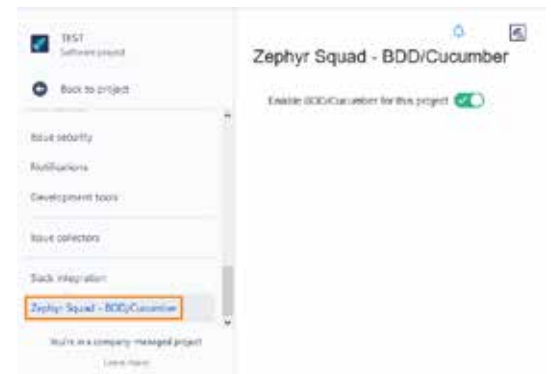
BDD is enabled by default for all projects. You can disable or enable BDD for specific projects as needed.

- Create BDD Function
- Turn on the BDD feature for your specific project.

QATestLab
quality is a rule

TEST IT
software testing system

ZEPHYR + CUCUMBER

HOW DOES THE BDD BACKGROUND FUNCTION WORK?

For all of the features mentioned, background scripts will appear on the Run Test page after all tests have been added to any of the test cycles/folders.

For a script schema, background scripts will be displayed on the Run Test page after all tests have been added to any of the test cycles/folders.

- Create BDD scenarios

ZEPHYR + CUCUMBER

- Create BDD text scenarios

- Add BDD Test to the test cycle
- Create a BDD function
- Go to the Test Cycle Planning page
- Create a test cycle and then add the function/test as you would in any other test case.
- Load function file from test cycle or folder

ZEPHYR + CUCUMBER

- Add Scenario Outline
- Create a BDD function.
- Click on the content of the function and enter the script name, then click Create.
- Click on the custom script you just created.
- Click on Test Information and go to the Test Information section.
- Insert the following script:
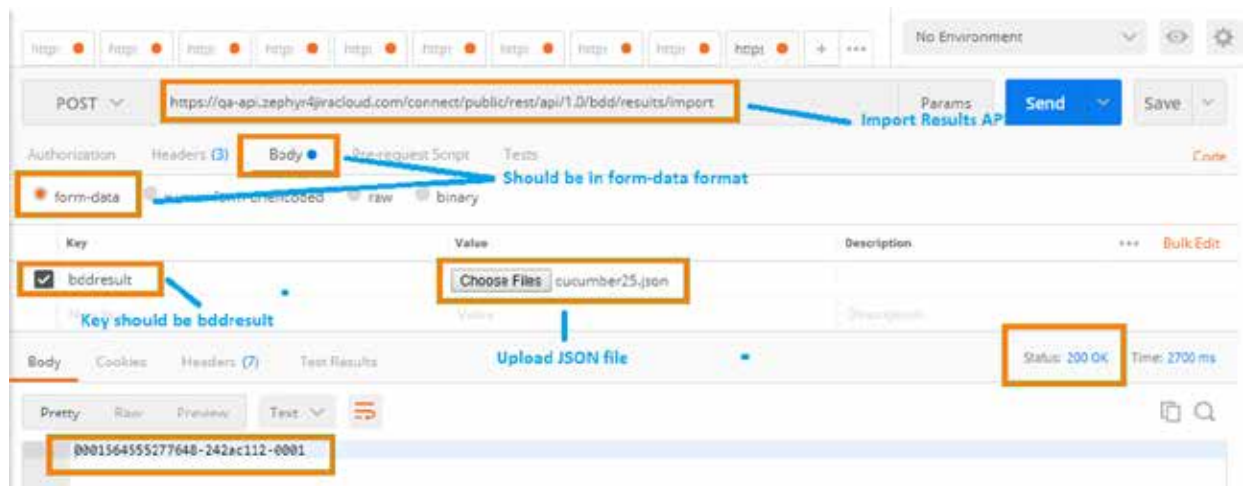
- Download the file with JSON results via Postman.
- Add the BDD function/test to the test cycle and/or folder.
- Load the function file from the test cycle and/or from the folder.
- Load the result file generated from the function file as shown in the image
- After downloading the results file, you may need to refresh your browser to see the results.

IN TOTAL:

Intuitiveness — 3\5

Functionality — 3\5

Resource Intensity — depends on Jira

Documentation Availability — a huge number of documents/videos that describe the process.

QATestLab
quality is a rule

TEST IT
software testing system

SPIRATEST + RAPISE

When using SpiraTest with Rapise, you get the ability to store your automated Rapise tests in a central SpiraTest repository, including full version control as well as scheduling and testing capabilities.
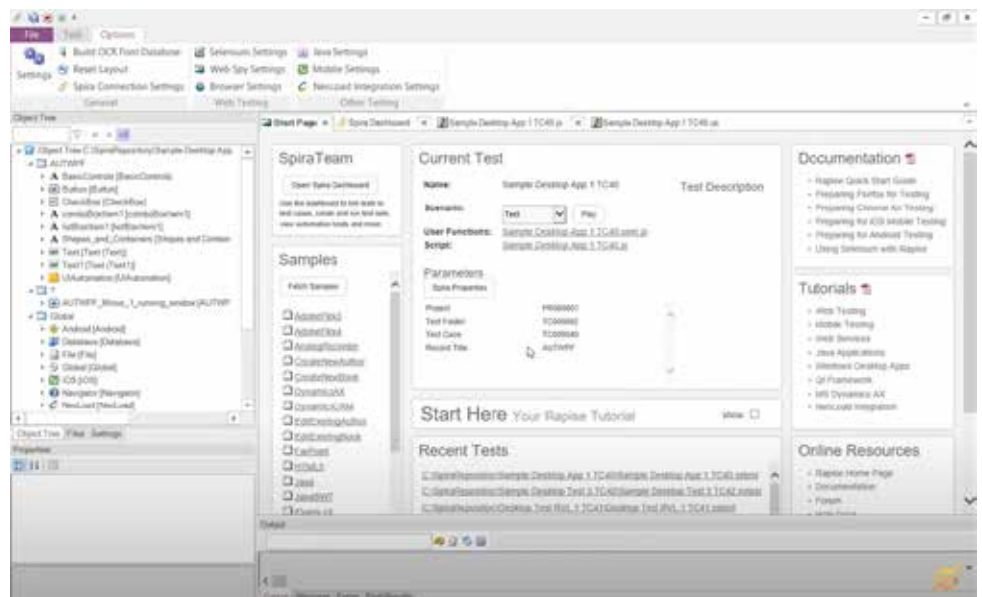
You can record and create test cases with Rapise, upload them to SpiraTest and compose test cases that will be executed on multiple remote computers. Tests can be executed instantly or according to a predefined schedule. The results are then transferred to SpiraTest, where they are archived within the project.
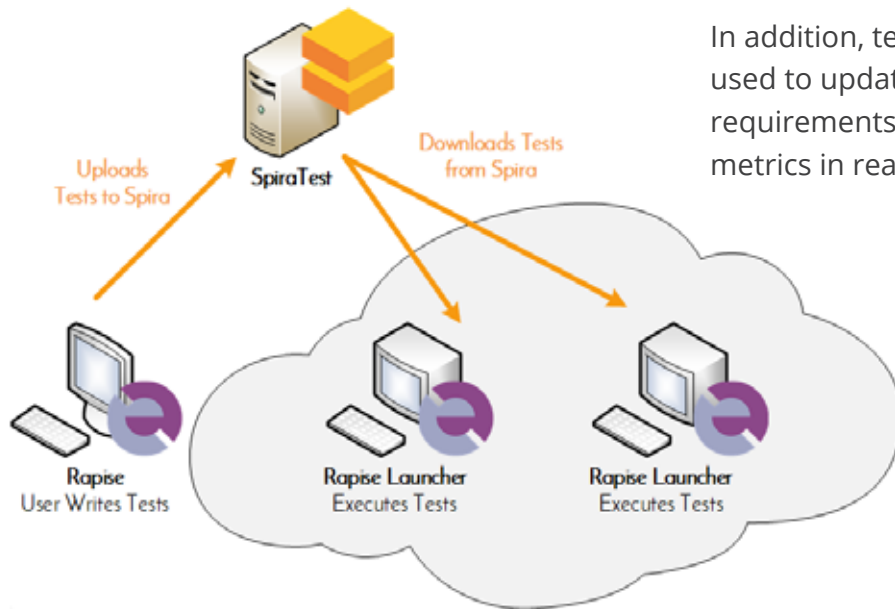
SPIRATEST + RAPISE

In addition, test results can be used to update test coverage requirements and other key metrics in real-time.

GENERATING AUTOMATION SCRIPTS FROM TEST CASES

Rapise can create new automated tests directly from your existing manual SpiraTest test suites. With this powerful feature, you can be confident that your automation script would cover all the required functionality and fully validate the requirements that are being met.
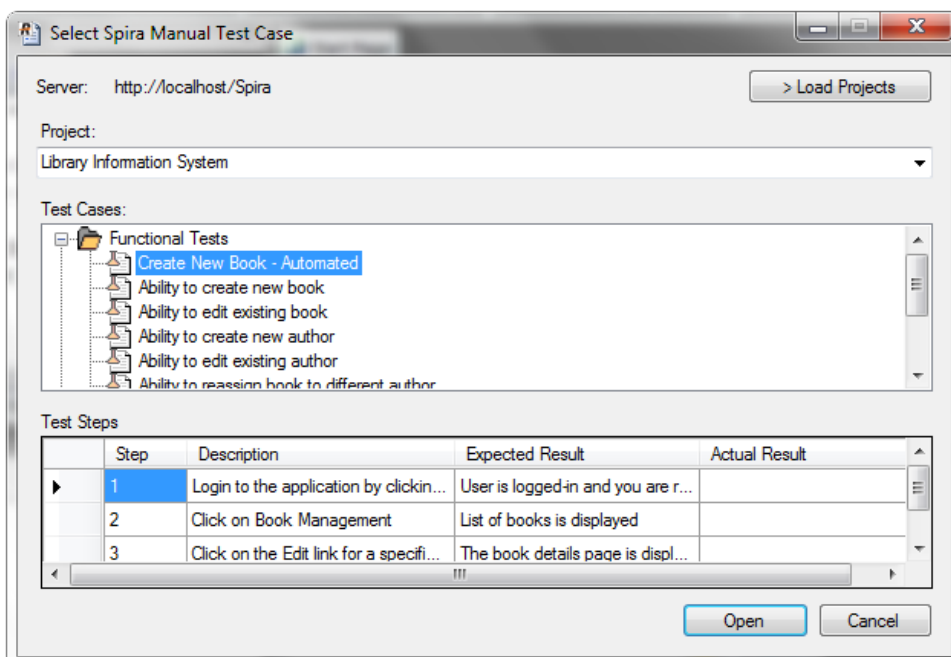
When you run these automated tests, Rapise can report the results at the test stage level, thus providing sufficient feedback to your developers and testers.

SPIRATEST + RAPISE

STORING EXPRESS TESTS IN SPIRATEST

Once you connect your Rapise to SpiraTest, you can use it to store versioned copies of your automated cases associated with the corresponding test cases in SpiraTest.

This allows you to manage your Rapise automation scripts and interact with multiple testers and developers working on those cases.

Each time you save tests, Rapise checks the versions stored in the central repository to determine which files to upload or download, to keep your workstation in sync with the test documentation.
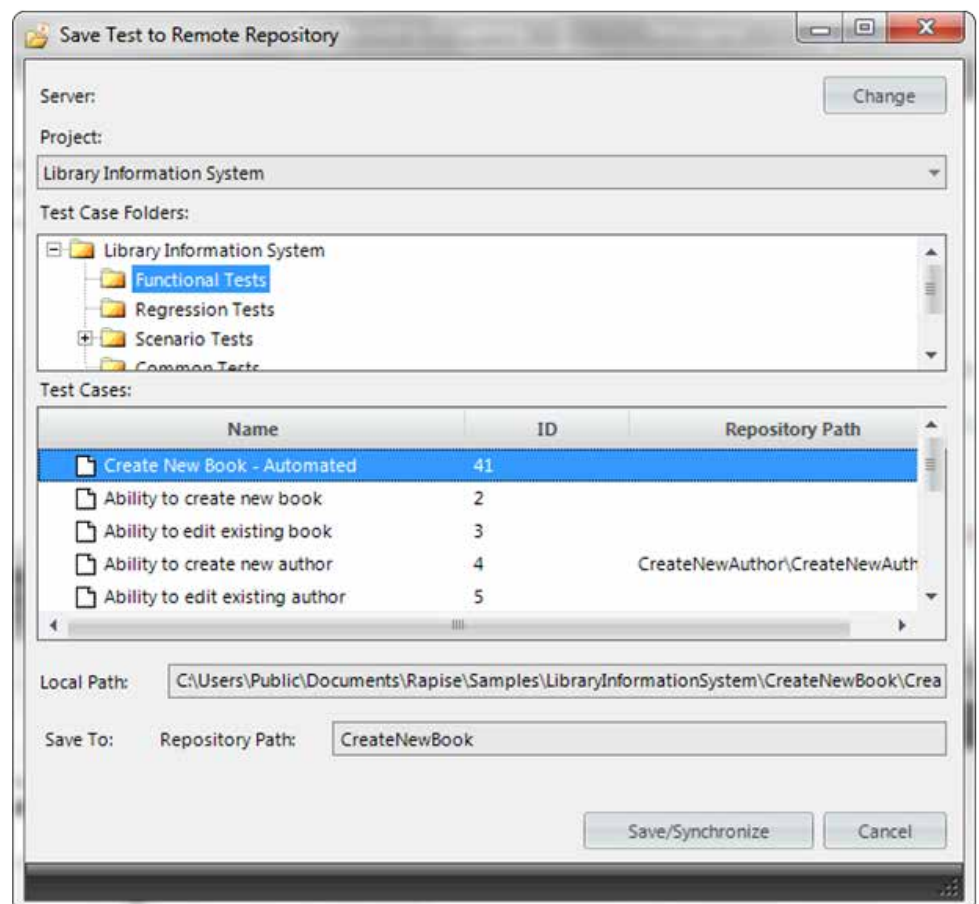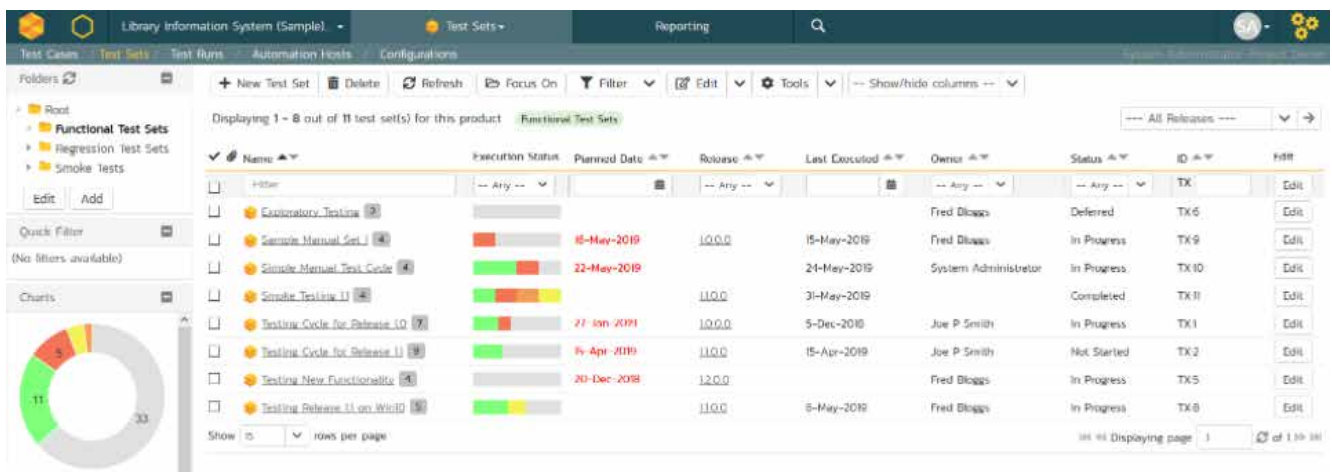
SPIRATEST + RAPISE

PERFORMING EXPRESS TESTS FROM SPIRATEST

After your Rapise cases are saved in SpiraTest, you can use SpiraTest's capabilities to schedule test case runs.

Using the Automation Hosts feature in SpiraTest, you can create different test suites to run different combinations of Rapise simultaneously on different computers.

When you use the ability to parameterize tests and pass values from SpiraTest to Rapise, you can run the same tests on multiple platforms and/or browsers using different combinations of test data. Run Rapise tests remotely using the free Rapid Launcher add-on for Rapise, which is installed with the main application.

SPIRATEST + RAPISE

When launched, the program will run minimized to the system tray and start polling the server. A poll will occur every X minutes (default 60) for any automated test cases that are scheduled to run.

When it's time to run the test, the software will launch Rapise. Rapise will then perform test actions and report the results to SpiraTest.

Basic Navigation Test

[TR-471096]   Execution Status: Failed

Releases

Release #:	5.2.0.0 – SpiraTeam v5.2
Build:	-- None --

People

Tester Name:*   ○ Rapise User

Properties

Test Set:	📁 User Interface Tests Change
Test Case #:	◆ Basic Navigation Test [TC1325]
Test Run Type:	Automated

Dates and Times

Execution Date:	4/27/2017 11:34:17 AM
Estimated Duration (h)	
Actual Duration (h)*	0.00

Test Steps

Step	Test Step Description	Expected Result	Sample Data	Test # / Step #	Actual Result	Execution Status	
1	Spira Simple Web Navigation Test	Test		/	Test Started: D:\SpiraRepository\Spira Simple Web Navigation Test\Spira Simple Web Navigation Test.js	Failed	🔗
2	Starting scenario: Test	Message		/		N/A	🔗
3	Chrome Process Created	Assert		/		Passed	🔗
4	Connected to Chrome window	Assert		/		Failed	🔗
5	Error calling SeS('User_Name').DoSetText("administrator")	Assert		/	Executed at: Test()-> login('administrator', 'PleaseChange')	Failed	🔗
6	Execution terminated by user	Assert		/		Failed	🔗

IN TOTAL:

Intuitiveness — 4\5

Functionality — 5\5

Resource Intensity — 2\5

Documentation Availability — documentation is available, but there are no video tutorials, which makes integration a little difficult.

PRACTITEST + EGGPLANT

The EggPlant and PractiTest integration enables testers and test managers to control their testing projects and provide complete visibility to their stakeholders.

By integrating these two systems, you can:

- Trigger the EggPlant's automated tests from within PractiTest
- Have one centralized area for the results of both your manual and automated testing efforts
- Provide everyone in the project with the information they need, in the format they expect it, and at the time they require it
- Get full confidence of having a full QA coverage

PractiTest supports both manual and automated tests, allowing you to run tests from any commercial or open-source tool.
Integration requires the Eggplant application to run in Eggdrive mode.

01 Create equivalents of your Eggplant Functional tests in the Test Library, choose Automation type test — Eggplant. Under the Automation Info tab, type the Suite Path and Script Name.
Suite Path: the path of the suite where the test resides.
Script Path: the name of the Eggplant Functional test.

PRACTITEST + EGGPLANT

02 Group these tests into a Test Set. Put values into the two new custom fields in the Test Sets that were added automatically in your project after enabling the integration.

Users can also choose to display the fields in the Test (Test Library) and Test Instances if they prefer.

03 Enter the Test Sets you configured earlier and press the «Save & Run Now» button within the test instance you would like to run via Eggplant.

04 Analyze the results of your automated test run.
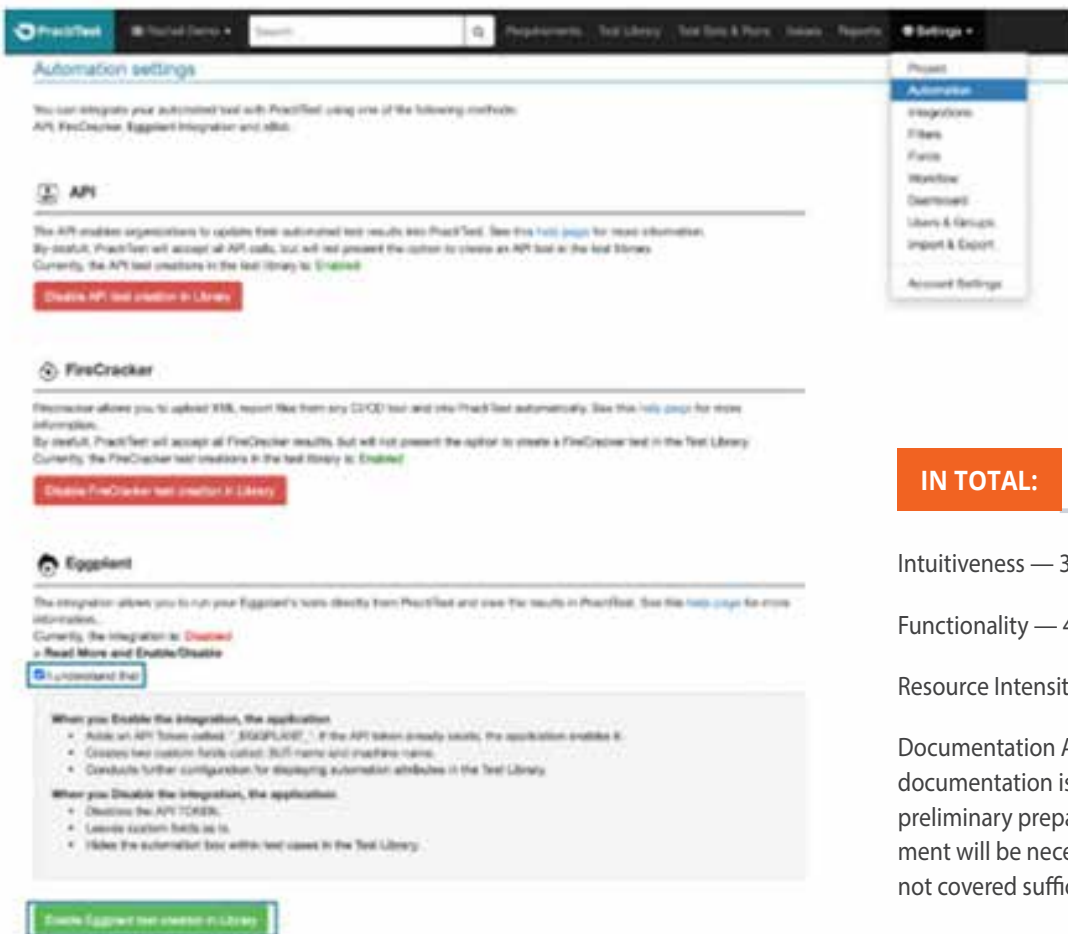
IN TOTAL:

Intuitiveness — 3/5

Functionality — 4/5

Resource Intensity — 3/5

Documentation Availability — the documentation is available, however, preliminary preparation of the environment will be necessary — this process is not covered sufficiently.

QATestLab
quality is a rule

TEST IT
software testing system

CASE STUDIES

QATestLab
quality is a rule

TEST IT
software testing system

BUDGET-FRIENDLY ALTERNATIVES TO TESTRAIL+RANOREX

The project we would like to start with is a platform for fitness coaches and nutritionists where they can advertise their services, as well as contact customers to schedule sessions, etc. It involves two interconnected websites — for customers and service providers.

PROJECT OVERVIEW

The project budget was considerably limited — that is why we had to cross out TestRail (prices start at $34/month per user) from our list of available options from the very beginning. Ranorex ($2890 per user) was excluded due to the same reasons. Besides, Ranorex implies C#, which can make it difficult to execute automated tests on Linux machines.

Therefore, here is our advice: if you have enough resources to invest in test automation — or there are no complex tests (or time to study programming) — then go for TestRail and Ranorex.

On the other hand, it is your only choice when it comes to the test automation for desktop applications for Windows, as there are not many relevant tools on the market at the moment.

BUDGET-FRIENDLY ALTERNATIVES

Selenium WebDriver TestNG Allure	One of the most powerful frameworks in terms of test automation and integrations. Provides convenient and customized reporting.
Selenium WebDriver TestNG TestRail	A great option for combining manual and automated testing. Enables simple integration thanks to the TestRail libraries.
Zephyr Cucumber	Only reasonable to implement if the project follows the BDD methodology
Selenium Junit/PyTest Test IT	A great alternative to 1st and 2nd options if you want to cut costs and combine manual and automated testing

BUDGET-FRIENDLY ALTERNATIVES TO TESTRAIL+RANOREX

**Test Management System /
Test Automation Tool Selection Process**

During the selection process,
the following options were considered.

**TEST MANAGEMENT
SYSTEMS (TMS)**

TestRail:
it could be the optimal option
since it is convenient for manag-
ing test cases and well-integrat-
ed with automated tests written
in Java with a combination of
TestNG and Selenium WebDriv-
er. However, we didn't choose
TestRail because the customer
found it quite expensive.

Zephyr:
this TMS was not selected since
it did not provide convenient
management of test cases.
Zephyr allows grouping test
cases into sets, but you cannot
create a tree-like architecture for
test case distribution.

Xray:
unlike Zephyr, it supports a
tree-like test case design, which
made it the option we settled on.
Xray has its own API, which is
quite complex to work with, so
integration with this tool can be
difficult and time-consuming.

**TEST AUTOMATION
TOOLS**

Cucumber:
it works best with projects follow-
ing the BDD development meth-
odology, otherwise it creates
extra work for the QA automa-
tion engineer and limits possibili-
ties in terms of writing test cases
and managing automated tests
in comparison with other tools.
This is why we didn't choose
Cucumber.

Nightwatch:
this tool was not selected since
we could not afford to spend
extra time studying it.

**Selenium WebDriver +
TestNG + Allure:**
this combination of tools was
our choice because it could
cover most of the automation
requirements and provided a
powerful set of capabilities for
writing automated tests.

WHAT TMS / TOOL COMBINATION TO CHOOSE FOR A FLEX-BASED APPLICATION

This project required us to automate the process of calculating indicators in gas transmission pipelines. We worked with a web application that received all data and calculated the main indicators. The distinctive feature of this project was the fact that it was written in Flex.

**Test Management System /
Test Automation Tool Selection Process**

During the selection process,

the following options were considered:

Selenium + Pytest:
the most suitable combination for web testing. However, we dismissed this option because Selenium did not support Flex/Flash objects. Besides, various kluges available at that time were not reliable and crashed often.

Robot Framework + Sikuli:
RF is an easy-to-use and versatile testing tool that offers many plugins (the Sikuli plugin in particular) and allows writing BDD test cases. This combination supports Flex (via Sikuli), so we were using it for the first couple of months but switched to another one later because of difficulties in supporting another level of abstraction (BDD wrappers) and issues with debugging tests at this level.

Sikuli + Pytest:
The most common testing framework; provides an image recognition tool. We chose this combination because it allowed more flexibility in working with the code and debugging it.

Alternatives we used:

Selenium + FlexMonkey:
added the ability to work with Selenium.

Ranorex:
an excellent test automation tool that supports Flex objects. However, we could involve only as an alternative due to its cost.

PYTEST OR CUCUMBER? THINGS TO CONSIDER WHEN CHOOSING THE FRAMEWORK FOR YOUR TEAM

This project was a modularly built tool, specifically created for the design and expansion of telecommunication networks. It automated to a high degree the process of delivering permit drawings, technical drawings and bills of material for construction purposes.

**Test Management System /
Test Automation Tool Selection Process**

During the selection process,
the following options were considered:

**Selenium Webdriver +
Pytest (analog of TestNG) + Allure:**
this combination was chosen as the most popular and easy-to-maintain option for creating a test framework from scratch.

Cucumber:
For a project where developers were in charge of software testing, Cucumber would not be a good option due to the complexity of test maintenance.

Ranorex:
the project did not have desktop components; dismissed for being too expensive.

TestRail:
the majority of the features could be found in JIRA.

Nightwatch:
could be an alternative, as it is an uncluttered test runner framework. However, Cypress.io, in our opinion, works much better.

WEB+MOBILE+API TEST AUTOMATION: CHOOSING THE RIGHT TMS / TOOLS COMBINATION

The next project we would like to present is the platform for car sharing that offers travelers, who leave their vehicles at an airport, the ability to rent out their cars to other travelers. The project realization included the automation of acceptance criteria scenarios for the new functionality and maintenance of the existing automation framework and CI automated tests for the iOS and Android applications.

**Test Management System /
Test Automation Tool Selection Process**

During the selection process,
the following options were selected:

Selenium WebDriver for Web UI Automation.

Karate and REST Assured for API Automation:
Karate for API automaton and REST Assured for test prerequisites.

Appium + Cucumber for Mobile Automation:
the classic combination when it comes to BDD test automation for mobile.

Alternative options:

Try **Selenide** for WEB UI.

For API automation, use **Retrofit** as an alternative, but REST Assured is easier and more popular.

There are no alternatives to **Appium** when it comes to mobile test automation.

WHY SELENIUM+TESTNG+ ALLURE IS OUR FAVORITE COMBO

This one was the ad-verification platform for advertisers, with next-gen ad-fraud prevention, brand safety enforcement and viewability control, where our team automated regression tests for the entire product functionality

**Test Management System /
Test Automation Tool Selection Process**

During the selection process,
the following options were selected:

Selenium+TestNG+Allure:
the most in-demand and popular combination so far. It provides the broadest options for logging and reporting. In addition to @Step, we can add detailed logging to the report (which is available in the console), as well as screenshots and attachments. Due to numerous annotations, Allure supports flexible reporting.

TestRail:
thanks to the aforementioned trio, TestRail is used solely for creating, storing and editing test cases and test suits. We opted for this tool due to its convenience, flexibility and scalability. There was only one reason why TestRail was connected to the core combination — we had to indicate the test case number in each test declaration, in the test name field. Thanks to TestRail, we could quickly navigate unfamiliar parts of the project to find necessary tests.

ABOUT QATESTLAB

QATestLab is a provider of independent software testing and QA (Quality Assurance) services.

We offer manual testing and test automation services worldwide for multiple industries (smart software, e-commerce, banking, healthcare, business solutions, game, etc.) in many directions:
- **web:** quality control for websites and web software;
- **mobile:** testing mobile software on various devices;
- **desktop:** verifying different aspects of desktop software products;
- **cloud:** cloud solutions testing based on project specifications.

Our portfolio of test automation services includes:
- mobile (native, hybrid) test automation
- desktop (apps, extensions) test automation
- web (UI, functionality) test automation
- cloud test automation
- API test automation

The QATestLab team of **40 QA Automation Engineers** has successfully accomplished **250+ projects**, writing test scripts in **Java, JavaScript, C# and Python.**

Faster time-to-market with enhanced efficiency of quality checks can be achieved with automated testing. Try our tailor-made automation solutions and see tangible results.

ABOUT TEST IT

Test IT is a focused software development company on a mission to make QA teams' life easier with the help of Test IT — a cutting-edge test management system. Test IT TMS unites manual and automated testing in a single sleek interface with clear and flexible reporting.

Deep integration with JIRA, flexible API, webhooks for CI/CD systems make Test IT unparalleled in terms of infrastructure and test environment. No need to set up complex integrations between TMS and automation frameworks - Public API and framework libraries make it super easy to connect autotests, launch test runs and collect data on failed autotests.

When you consider alternatives to **TestRail + Ranorex** and want a more cost-effective solution — consider Test IT test management system.

We help hundreds of SMBs and large Enterprises in many industries:
- Banks and fintech
- Enterprise Software Development
- Web and Mobile Development
- Game Development
- Online Retail and E-commerce
- Telecommunications
- EdTech

Get a personal 30-min online demo with a walkthrough of the core features and address your needs. Or you can get a 30-day free trial for up to 25 users to check Test IT out on your projects!

CONTACTS

QATestLab
quality is a rule

Website: **qatestlab.com**

E-mail: **contact@qa-testlab.com**

TEST IT
software testing system

Website: **testit.software**

E-mail: **info@testit.software**