Jalasoft Automation Methodology

Contributors

This document has been created by people with many years doing automation testing for Jalasoft's projects

- Gonzalo Alba
- Raul Garvizu
- Nedsayde Escalera
- Jafeth Garcia
- Wilge Vargas
- Silvia Valencia
- Danny Flores

Jala-oft

Content

- Introduction
  - What is an automation project?
  - Common Mistakes
- Automation Project phases
  - Phase 1 – Mission/Requirements/Analysis
  - Phase 2 – Designing/Implementing Framework
  - Phase 3 – Automation Plan and Execution

Jala-oft

What is an automation project?

- A test automation project is a software development project. As with any software development effort, if we use poor requirements, design, and coding practices, we will get bad results.
- It is **code**, so we have to treat it as a programming project.
  - understand the **requirements**;
  - adopt an **architecture** that allows them to efficiently develop, integrate, and maintain features and data;
  - adopt and live with **standards**. (it makes sense for two programmers working on the same project to use the same naming conventions, the same structure for documenting their modules, the same approach to error handling, etc..)

Jala-oft

COMMON MISTAKES

1. Don’t underestimate the cost of automation.
2. Don’t underestimate the need for staff training.
3. Don’t expect to be more productive over the short term (no immediate payback).
4. Don’t spend so much time and effort on regression testing.
5. Don’t use instability of the code as an excuse.
6. Don’t shoot for “100% automation.”
7. Don’t use capture/replay to create tests.
8. Don’t create test scripts that won’t be easy to maintain over the long term.
9. Don’t fail to treat this as a genuine programming project.
10. Don’t insist that all of your testers be programmers.

Cem Kaner

Jala-oft

Cost of automation

Don't underestimate the cost of automation.

- It will be needed to train the staff.
- There won't gain back much of that time during the Release (such as a product Release 2.0) in which it is done the initial automation programming.

Jala-oft

Don’t spend so much time and effort on regression testing

- Skilled manual testers often run new variations as they retest the program. The new variations address the same areas of the program, but they use different data or order tasks differently or combine tests that were run separately before.
- There is value in being able to conduct smoke tests or in being able to run a final qualification test at the end of a release. But there are limits to this value. In allocating resources for regression testing (redoing old tests), it should be allocated significant resources for tests that have never been executed before i.e. for exploratory or ad hoc testing. Otherwise, the tester would be missing many problems.

Jala-oft

Don’t use instability of the code as an excuse

- Some people advise testers not to do any automation until the program’s user interface has completely stabilized
- Saying that code isn’t ready for automated testing until the UI is frozen seems like saying that code isn’t ready for testing until all the bugs have been taken out.
- Change adds complexity that will have to be managed, but that doesn’t give an excuse to not automate (if it is needed automation) or to automate badly.

Jala-oft

Don't shoot for "100% automation."

- Some people recommend that testers automate 100% of their test cases. This is not recommended because many of y black box tests only once. Many, many bugs are found during these exploratory tests.
- To automate these one-shot tests, the tester would have to spend substantially more time and money per test. In the same period of time, the tester wouldn't be able to run as many tests. Why should the tester seek lower coverage at a higher cost per test?

Jala-oft

Don’t insist that all of your testers be programmers

- Many excellent black box testers have no programming experience. They provide subject matter expertise or other experience with customer or communications issues that most programmers can’t provide. They are indispensable to a strong testing effort. But you can’t expect these people to write automation code.
- Therefore, you need a staffing and development strategy that doesn’t require everyone to write test code.
- You also want to avoid creating a pecking order that places tester-programmers above tester-nonprogrammers.
- This is a common, and in my view irrational and anti-productive, bias in test groups that use automation tools. It will drive out your senior non-programming testers, and it will cost you much of your ability to test the program against actual customer requirements.

Jala-oft

Automation Project Phases

- Phase 1 – Mission/Requirements/Analysis
- Phase 2 – Designing/Implementing Framework
- Phase 3 – Automation Plan and Execution

Jala-oft

Phase 1 – Mission/Requirements/Analysis

- Mission
  - Automation Mission
  - Goals
- Requirements
  - Framework
  - Software Under Test (SUT)
  - Project tools and development process (Client Process)
  - Tests to be automated
  - Client Expectations
- Analysis
  - Requirements Document
  - Automation Tool
- Strategies

Jala-oft

Automation Mission

- Questions that help to identify the test automation mission
  - What kinds of bugs are you looking for?
  - What concerns are you dressing ?
  - Who is your audience?
- These are possible missions
  - Improve test coverage
  - Reduce testing costs
  - Automate regression tests
  - Find Important bugs fast
  - Assess the software stability, concurrency, scalability
  - Verify Key features
  - Measure and document product quality
  - ….

Jala-oft

Requirements - Framework

- Framework
  - Do we need a Framework?
  - Is there a framework already implemented?
    - Is it still being used?
    - What problems does the framework have?
  - What is the framework scope?
    - Enterprise-oriented - Test automation to support different product lines and projects in the organization.
    - Product-oriented - Test automation activities focused towards specific product line of applications
    - Project-oriented - Test automation effort focused towards specific project and its test process

Jala-oft

Requirements – System Under Test (SUT)

- Type of Application
  - Is it a Web application?
  - Is it a Desktop application?
  - Is it a Mobile Application?
- Architecture
  - Is it a Client/Server?
  - What is the DataModel?
  - What are the Layers?
  - Is it using Web Services?
  - Does it interact with other products?

Jala-oft

Requirements – System Under Test (SUT)

- Platforms/Scenarios
  - All in one?
  - Distributed?
  - What are the operate systems supported?
  - What are the browsers supported?
- Technology Stack
  - What technologies have been used to implement the UI interface?
  - Are there third party tools used on the UI Interface?
  - What is the database technology used for the SUT's database?
- What is the product's maturity?

Jala-oft

Requirements – Project tools and development process

- What is the technology used for the virtualization?
- Is there a process for continuous integration?
  - Which is technology used?
- What is the test management tool?
- What is the bug tracking tool?
- What is the code versioning tool used?
- How many weeks does an iteration have?
- What is the process during the iteration execution?

Jala-oft

Requirements –Tests to be automated

- What kind of tests have to be automated?
  - Acceptance Tests?
  - Regression Tests?
  - UI Tests?
  - Non-Functional Tests?
  - End to End Tests?
  - Backend Tests?
- How often would the test cases be executed?
- How are the test cases written? Ask for a sample
  - This helps to verify if the test cases follow a structure that will make them easy to understand and automate.

Jala-oft

Requirements – Client Expectations

It is very important to know the client's expectations.

- What is he expecting In Short Term?
- What is he expecting in Long Term?
- What is the percentage of test cases to be automated?

Jala-oft

Analysis – Requirement Document

- After analyzing the requirements a document should be created documenting what would be supported by the automation framework.
  - The Functional and Non-Functional requirements

Jala-oft

Analysis - Automation Tool Selection

- Selecting the correct automation tool for the project is critical, there are many points to consider where the most important is the compatibility with the SUT and the tests to be automated.
- Below is a summary of the points that need to be considered when evaluating different automation tools:
  - Technology Stack
  - Type of tests to be automated
  - Scenarios Supported by the SUT
  - Licensing vs Budget
  - Documentation available about the tool
  - Integration with project tools

Jala-oft

Analysis-Automation Tool Selection

- With all those points create a matrix which will help you to make a decision.

Requirement	AUT tool 1	Comment	AUT tool 2	Comment	…
SUT					
Type of Test to be automated					
Scenarios/Platforms supported					
Licensing vs Budget					
Integration with project tools					
Documentation available about the tool					
…..					

- When doing the evaluation it is good to have a basic knowledge about the SUT

Note: each point mentioned has to be expanded

Jala-oft

Strategies - Overall

- **Project Strategy**
  - Explain on a document the Strategy that will be applied in order to reach the Goals and the Mission
- **Test Strategy**
  - Explain on a document which would be the best test strategy for the project for instance:
    1. Automate first Smoke Test
    2. Acceptance Test for the newer features
    3. Acceptance Tests for old features
    4. Non-Functional Test

Jala-oft

Phase 2 – Designing/Implementing Framework

- Design
  - Type of Framework
  - Design Pattern
- Proof of Concept (POC)

Jala-oft

Design- Type of Framework

There are four well known types of frameworks

- Modular
  - This involves the creation of independent scripts that represent the modules of the application under test
- Data Driven
  - The test input and the expected output results are stored in a separate data file (normally in a tabular format) so that a single driver script can execute all the test cases with multiple sets of data.
- Keyword Driven
  - Keyword driven testing is an application independent framework utilizing data tables and self explanatory keywords to explain the actions to be performed on the application under test
- Hybrid
  - Use two or more type of frameworks

Design – Design Pattern

Design Pattern is used to resolve common problems when designing an application or system.

- Singleton
- Factory
- Builder
- PageObject
- Strategy
- Facade
- …..

Jala-oft

Development Strategy – Proof of Concept (POC)

- A proof of concept (POC) is a demonstration, the purpose of which is to verify that certain concepts or theories have the potential for real-world application. POC is therefore a prototype that is designed to determine feasibility, but does not represent deliverables.

Jala-oft

Development Strategy – Proof of Concept (POC)

- Select few Test Cases to be automated
  - The test cases selected has to be the ones that work with different components in UI, has different type of verifications (UI, DB).
- Once implemented, demo it and get feedback (from Client) which will help to
  - Identify similar projects that has to be supported by the framework
  - Identify type of tests that has to be supported by the framework
  - Identify scenarios that still need to be supported
  - Clarify client's doubts
  - Clarify automation team's doubts
  - Approval from the client.
- Implement the final version of the framework based on the feedback

Jala-oft

Development Strategy – Framework Qualities

- As any other development project the code in this case the framework has to have the following qualities
  - Maintainability
    - How easy is to maintain the test cases in case there are changes
  - Independence
    - The degree to which test case stands alone. If there are dependencies the planning the order of execution becomes more complex
  - Efficiency
    - Best usage of the resources to reach its propose.
  - Extensible/Flexibility
    - How easy is to extend the framework to other areas supported by the SUT (API, Web, Mobile, Desktop)
  - Reusability
    - How easy is to reuse the methods created
  - Readability
    - How easy is to read the test cases and methods created
  - Robustness
    - How the framework handles the unexpected errors.
  - Modularity
    - Scripts that can be efficiently assembled to produce a unified system without redundancy or omission
  - Documentation
    - How the automated test case is documented in order to make sure that any engineer can understand it

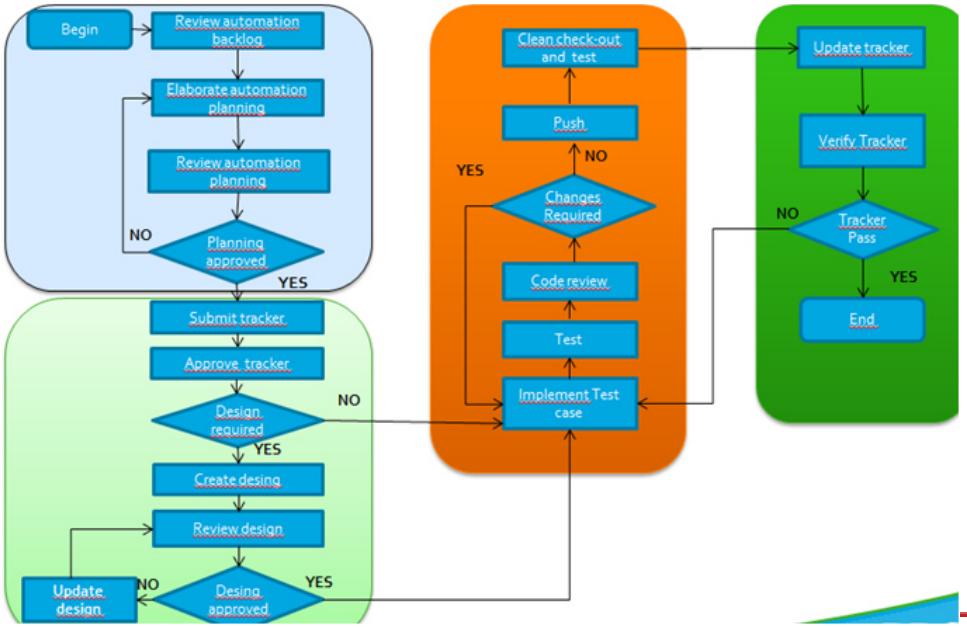
Jala-oft

Phase 3 – Automation Plan and Execution

- Development Process
- Automation Plan
- Automation Plan Execution
- DevOps (Common Work)

Development Process

As any other development project it has its own process from selecting a manual test case and having it automated. For example:



Jala-oft

Automation Plan

The automation plan as any other plan has to contain as minimum the following points described

- Scope
- Strategy
- Scenarios
- Resources
- Milestones
- Metrics
  - What we want to measure

Note: The automation plan can be different depending of the project

Jala-oft

Automation Plan – Test Cases

- The manual test cases have to be clear otherwise they will have to be re written
- Categorize the test cases into levels of difficulty/effort which is basically the cost in man/hours. E.g: Very Low and Low, Medium , Very High and High.
- Team analyses all test cases and define the cost for every category (e.g. very high costs 1 man/day, etc.)
- Team identifies a strategy to automate the test cases and save execution time e.g. merge test cases, use data driven, use backend to load data…
- Team puts all the test cases within their corresponding categories. With that done plus the vacation/holiday/capacity plan we come up with an estimate and let stakeholders decide what to defer

Jala-oft

Automation Plan - Execution

- Agile/Scrum
- Put your plan in the project management tools.
- Assign responsible.
- Follow scrum for daily follow up.

Jala-oft

Automation Plan - Execution

- Reports
  - Test Case execution
    - # Passed, Failed (SUT issue or changes, Automation Framework)
    - SUT build, Framework builds
  - Status of the SUT
    - Functionality got or not broken
    - New defects have been found
  - Progress of test cases automated
    - % of test cases automated
      - Smoke Tests
      - Acceptance Tests
      - Regression Tests
      - Per Feature
    - # of test cases automated per day, per week, per sprint

Jala-oft

Automation Plan - Execution

- Framework Maintenance when
  - UI changes
  - Backend changes (rare)
  - Configuration that was not considered
- Regression analysis
  - What was passing yesterday, and now is failing
  - What was failing yesterday, and now is passing
- Update the automation plan
  - The automation plan can be updated depending on different factors:
    - Priorities changes
    - Features that are hard to be automated
    - Automation velocity – reach a more accurate number of test cases automated in a determined milestone

Jala-oft

DevOps (Common Work)

- Continuous Integration
  - Coordinate the integration of the automated test cases in the CI practice

Jala-oft

References

- http://www.cs.colorado.edu/~kena/classes/5828/s12/presentation-materials/ghanakotagayatri.pdf
- http://softwarequalitymethods.com/papers/autoarch.pdf
- http://www.softwaretestpro.com/ItemAssets/4772/AutomatedTestingHandbook.pdf
- http://www.kaner.com/pdfs/shelfwar.pdf