Why Test Automation Fails and what to do about it

(Test Design and Implementation Tips)

Barry Solomon
PNSQC 2020
Who am I?

I have been involved in test automation since like forever

- Principle Sales Engineer (Mercury Interactive)
- Director of Technical Alliances (Mercury Interactive)
- QA/Testing Instructor (ALP International)
  - Testing best practices Manual/Automated
  - Test Automation Framework workshops
- CTO Web Search Engine Company
- Full-Stack developer since like forever
- Senior Solutions Engineer (TestIM)
Overview

- Why Test Automation efforts often fail
- Challenges to Test Automation Success
- Time - Test Automation’s Nemesis
- Anatomy of Successful Test Automation
- Test Automation - Other Use cases
- The Right Tool for Test Automation Success
As a seasoned developer and test evangelist...

• Ask me how many tests I created?
Squat

• Well maybe a few more than squat.

• Hundreds of unit tests for TDD while developing backend logic.
  • This was successful due to it saved a lot of time while testing
  • It was efficient and maintenance free
  • Though once the code was written it was reasonably static and the testcase sat idle

• I did create a test suite for a temporal feature that had to be automated
  • The consequences of not automating would have been waiting hours/days for a single test.
  • In the end it took several weeks to create the tests and after several releases they broke.

• Blackbox testing seemed to always be abandoned after a few product iterations
  I always started out with the best intentions using Selenium and/or other tools, but as time went on, developers were less available and QA had more broken tests than working
  As tests broke and deadlines approached, getting help fixing tests became more trouble than it was worth.
Why Does Test Automation Fail?

- Unrealistic Goals For Automation
- Selecting the Wrong Automation Platform
- Lack of Proper Planning
- Contaminated Test Environments
- Underestimating Time, Cost and Effort
- Lack Of Training
- Failing Test Automation Frameworks
- Inadequate Resources to acquire the right tools
- QA Team Makeup
- Test Maintenance
Challenges to Test Automation

• Moving to continuous deployment but hindered by test authoring
• Current test automation framework is not evolving fast enough to accommodate new systems and assets
• Poorly Written User Stories
• No or Incomplete Acceptance Criteria
• Changing Requirements

Why Test Automation Fails and what to do about it
Time is Not on Our Side
Question

• You have 1000 manual test cases that HAVE to be run, how long does it take?
It’s Complicated

• Given
  • You have \(q\) QA Engineers and \(m\) manual tests that take \(t\) minutes to run on average and there are only \(h\) productive hours in a given day

• Question:
  If there are only 1-3 days for release cycle testing, how long does it take to test?

Ooh, that’s a tricky one. You have to use calculus and imaginary numbers for this.

Imaginary numbers? You know, eleven, thirty-two, and all those. It’s a little confusing at first.
Follow-up Question

• You have 1000 **automated** test cases that HAVE to be run, how long does it take to test?
It’s Complicated

How many tests broke THIS time?
Time is our Enemy

• Time is the biggest contributor to success or failure
• Key to success is to reduce wasted time to a minimum
• Time Reduction Targets
  • Time planning tests
  • Time designing tests
  • Time creating tests
  • Test execution is free (or at least it should be)
  • Time maintaining tests
Time to Test Catch-22

- No Time to Automate
  - Tough! There never is, never was, and never will be enough time. Deal with it!
- Test Automation theoretically should help, but there is no time to even test what HAS to be tested now, so how can I automate?
- Management is on my back to get my tests done yesterday!
How do You Eat an Elephant?

PLEASE DON’T EAT ME.
Why Test Automation Fails: Test Design and Implementation Tips
How do you succeed w/ Test Automation?
One Test at a Time

• Find a test that is time manually consuming and easy to automate
  • Automate one, just one (or maybe a few)
    • This saves a time during the next test cycle
  • Do not use (all) the time saved for more manual testing
    • Take some of it for running an extra testcase
    • The remainder use to automate a test or two

In this way, you can start automation and get out from under the mountain of non-run tests and live a happier, healthier life
Simple Tests

- Tests that are created to kickstart automation, but have a short shelf life and may be replaced with more advanced tests later
  - Record every manual test during regular testing
  - These tests are simply recorded with no effort at modularity or reusability
    - Full business process with one set of data (that was recorded)
  - Quantity vs Quality

It is even more important here to use a test tool that is fast and accurate and with little to no maintenance

**If it breaks, toss it!**
Top Automation Candidate Ideas

- Iterative Tests (Every Option)
- Tests with Diverse data
- Repetitive tests with lot of tedious validation (Grid Sorting/Filtering)
Test Prioritization

- Initial Tests
  - Manual Test Conversion
  - Obnoxious Tests
  - MVP Tests
- Secondary Test Targets
  - Bug Tests
  - Semi-Automated Tests
  - High Value Functionality
Test Planning - Initial Test Plan

- Login/Logout
  - Positive
  - Negative

- Registration
  - Email Validation

- Shopping Cart
  - Add
  - Remove
  - Checkout

- Product Listing Validation
  - Prices are correct
  - Data is accurate
Modular Test Design

• Like Lego blocks, build your tests
  • Login
  • Navigate to Functional Area X
  • Enter data for target single page/transaction
  • Validate result
  • Logout
Example - Product Listings Test Case

- Login (Shared Test Case)
- Product Page Navigate (Shared Test Case)
  - Product Search (Shared Test Case)
    - Parameters
      - Product Search String
      - Expected Product Count
      - Array of Product Data for Validation
    - Validate (Can be done incrementally)
      - Target Product Name is found
      - Expected Product Count
        - > 0, Expected Product Count
      - Product Information
        - One, Samples, All
      - Sorting/Filtering
- Logout (Shared Test Case)
Bug Tests

• Install SaaS based bug capture tool on everyone’s computer
  • Create a Jira project for incoming bug tests
  • Ask everyone to record bugs and save to this project
  • Pickup your free test cases
• These tests are time negative
  • Record steps to reproduce a bug, save into Jira
  • Improves communication between QA and Dev
  • Creates reusable test cases for free
• If your test automation tool has no ability to capture bug tests
  • Time to find another tool
It Takes a Village

- Install SaaS based automation tool on everyone’s computer
  - Create a Jira project for incoming raw tests
  - Ask business analysts to simply record and save to your project
  - Pickup your free test cases
  - Added benefit, screen recordings captured can be reused for documentation & training
- If your test automation tool has no ability to capture bug/user tests
  - Time to find another tool
Manual Test Conversion

• Easy method for manual to automation conversion
  • Take manual tests and create an automated test that simply prompts a user to run an excel file and hit a button to note the results
  • Simply record your manual tests

• Benefits
  • Allows for test results to be entered into the new test management paradigm
  • Creates automated tests for free
Semi-Automated Tests

- Create automated tests that login and navigate a manual test user to the target functional area to be tested
  - Saves time of manual tester navigating to the starting point every test
Not All Manual Tests Are Bad

- Some test cases are just too complex and dynamic to automate
  - Too logically complicated
  - Too dynamic
    - UI layouts that constantly change constantly break
  - Too much maintenance

- If it takes days/weeks to create a test that takes 2 minutes to run, what’s the point?
  - Besides, it will probably break next release anyway
High Value Target Data

- Data that causes the most carnage
  - -Infinity, Min, -1, 0, 1, Max, +Infinity
  - Blanks
  - ‘ and “
  - Text in number fields
  - Numbers in text fields
  - Huge text entries, thousands of characters
  - SQL Injection Data

"Choose your data wisely, for it may be your last..."
Test Automation Best Practices

• Create Structured, Short Single-Purpose Tests
• A Test’s Initial State Should be Consistent
• Compose Complex Tests from Simple Steps
• Add Validations in Transition Points
• Avoid Sleep to Improve Stability. Use Wait for or similar
• Use Abstractions
• Use Setup and Tear-down
• Test Early and Test Often.
• Divide your Automated Testing Efforts.
• Use data-driven instead of repeated tests
• Use a stable test environment.
• Create Automated Tests that are resistant to small changes in the UI.
• Name your tests wisely
• Take screenshots for failure investigation
100% Test Coverage Unicorn

• Never say never, but most likely never achieve 100% test coverage
  • I have heard rumor of some company claiming 100% coverage, but that is a unicorn.

• Unicorns are annoying

Personally, I would be happy if I could just test what I need to test before the next build drops without being yelled at. 😊
Choose The Right Tool For The Job

• Pay attention to the level of your team’s skills.
  • Non-Technical Manual Testers
  • Test Automation QA Engineers
  • Annoyed Developers

• Excellent functionality vs steep learning curve
  • Easy to use but functionally bankrupt
  • It may be a challenge to master some tools for non-technical users

• Reliability of created test
  • How much maintenance is required before we punt?

• Extensibility
  • If tool does not support something out of the box, how easy is it add needed functionality?
Choosing the Best Tool for Automation

- Ability to run repetitive tests with minimal to no manual intervention
- AI based dynamic locators. Images suck
- Integration with CI/CD and Bug Tracking
- Codeless Tests with the ability to code as needed
- Coded for those who prefer coding everything
- Even better, find a tool that lets you do both at the same time!
Last Words - Test Migration

• They say the first step to recovery is admitting you have a problem...
  • I have all these test already built, we can’t just throw them out
    • Tough, proverbial square peg round hole
  • We have this really great automation framework
    • Then why are we talking?
  • We already spent a lot of money of this other tool
    • And ‘we’ are happy to throw good money after bad?
• Selenium is free
  • Free is not really free. Time is money afterall.
Recap

• Test Automation failure causes
• Challenges to Test Automation Success
• Time - Test Automation’s Nemesis
• Successful Test Automation Guidelines
• Test Automation - Other Use cases
• The Right Tool for Test Automation Success
• Unicorns are annoying
Thank You