

Test Automation Checklist

From Spring WIKI

Contents

- 1 Analysis
- 2 When to Automate
- 3 Selecting the TOOL
- 4 Strategy
- 5 Implementation

Analysis

- What problems are you trying to solve?
- Have you segregated test cases which can be automated?
- Do you need permission to automate?
- What tools are already in place?
- Are elements identifiable - Identify objects from the screen using Location inspector tools?
- Have you checked test environment feasibility?
- What skills do you have access to?
- What is the benefit of Automation? Did you calculate ROI?

When to Automate

- Check status of the development of application? Is it near end of product life-cycle?
- Is the product stable?
- Aren't any planned major enhancements in core functionality?
- Is test data available?
- Do we have defined expected test results?
- Is the core functionality and navigation flows approved accepted by client?
- Are test conditions and preconditions detailed?

Selecting the TOOL

- Functionality -
 - Evaluate tools which falls in the line for Automation?
- Usability -
 - Is it easy to learn and adapt?

- Availability of training?
- Maintainability -
 - Does the company have support for this tool?
 - Is there any online used community?
- Affordability -
 - Cost Factor
- Compatibility -
 - Is the demo version of the tool available?
 - Verify its compatibility with OS ?
 - Does it identify all object within the application?
 - Could you identify all possible events for each object(Mouse hover, drag, Selecting custom dates etc)
 - Is there Recording and playback option available?

Strategy

- How large is the code base?
- Do you have Functional knowledge of application?
- Have you identified and prioritized modules based on their importance?
- How will you maintain automated test scripts?
- How many platforms/technologies does the software need to be tested to?
- How many test environments will you need?
- Are all test environments same?
- Will you need to use distributed testing?
- How do you document?
- Do you need coding standards?
- Where do you report bugs?
- Are you planning to use default reports or customized reports?
- How will you manage the code base?
- Do you use a version control system?

Implementation

- Required Access/Privileges for the application is granted to automation tester or not
- Vertical and Horizontal Decomposition for feature assignment
- Did you select appropriate Framework as per need?
- Are you creating MAVEN or JAVA project?
- Is design principle (POM) being followed?
- Keep everything that changes or which is bound to change in a separate file
- Is test data kept separately?
- Have you created a utility class which will cover all generic functions [Ex- File Read/Write, DB Connection]
- Is timeouts used to handle unexpected situations (control not present) ?
- Have you given any hard-coded sleep value?
- Is there any hard coded assignment?
- Are you reading data from configuration file?

- Are assertion check there to justify your test scenarios?
- Have you closed the FILE/DB after operations?
- What tests or code can you re-use?
- Have you prioritize test cases in TestNG?
- What happens with the test results?
- Are we going to execute tests in parallel?
- Can test scripts be executed on Multiple environments?
- Are we running all classes as a suite using XMLfile?
- Will test results (passes and failures) be logged automatically?
- What happens if the tests break (or don't work)?
- How will the tests recover from failure?
- How will failed tests effect future tests?
- Have you prepared flow diagrams?
- Have you done code cleanup?
- Are you keeping record of test cases which are failing in test run?

Retrieved from "http://wiki.sct.lan/index.php?title=Test_Automation_Checklist&oldid=3147"

- This page was last modified on 3 May 2018, at 16:13.
- This page has been accessed 27 times.