# BayTechServices

**When** *Good Enough*
**is** *Not an Option*

# When Test Automation Makes Sense

## and

# When it Doesn't

**www.BayTechServices.com**

# Background

With development cycles become shorter by day, the testing window is constantly getting squeezed.  Web-based software and Software-as-a-Service have shrunk product releases from months to weeks, even days.  This requires software testing to be completed in an even shorter time.  Can automated testing help?  The intuitive answer is yes.  The real world answer is yes – but.

In this paper, we will discuss:
1. Why automate – The benefits of automation.
2. Some reasons to be careful.
3. Types of tests to automate.
4. Test that are not good candidates for automation.

# Why Automate

Done right, test automation can be a powerful tool to accomplish a wide variety of testing and business goals.  Let us recap some of these:

- **Reduced execution time and cost**.  Once adequately designed, automated tests execute much faster and can run unattended.  This save testing resources (time and money).
- **Increased test coverage and improved quality**.  With automation, you can increase the breadth and depth of testing possible for each release.  Increased coverage will allow you to uncover bugs earlier.
- **Repeatability**.  Majority of the tests done for each release are regression tests.  In other words, the same tests are executed with every release – an area that automation is really good at.
- **Flexibility**.  In real-life development environments, development delays can and do happen.  Quiet often, this leads to a squeeze on testing resources and time.  With automation, repeat tests can be automated, so testers can focus on testing new features in the short window they may have.
- **Increase effectiveness of manual testing**.  As described later in the paper, much of the testing is still done manually.  By freeing testers from having to execute repetitive and mundane tests, test automation frees them to use their creativity, knowledge, and instincts to discover more important bugs.

# Reasons to be Careful

With so many powerful benefits, why is that most testing is still done manually. The reason, like with anything else, is that test automation is not free. Let us look at some of these costs and why it pays to be careful:

- **Cost of automation tools.** The due diligence to find the right tool for your need is by itself a major cost. It takes research to ensure the tool will work with all your current applications. The tools themselves are not cheap. Besides the initial purchase and maintenance, it requires time and training to learn to use them. And you may need more than one tool for different types of automation.
- **Time and cost to implement automation**. It takes time to implement automation, and yet more time to maintain the library of test scripts. Testers may not have the luxury of necessary time for proper test automation.
- **Skillset**. Not all testers are adept at automation tools. A test automation project can sometimes be as complicated as a development project – and your testers may or may not have the proper skills.
- **Lack of planning**. For test automation to work, proper planning is a must. Without it, test automation will not produce the desired results. For example, if companies develop features without updating requirements documents, automating tests could be difficult.
- **Missing the big picture**. Pursue test automation to achieve more effective and efficient testing of your software, with the resources you already have. If pursued for unexpected goals such as eliminating testers altogether, eliminating all manual testing, you are likely to be disappointed.

# Types of Tests to Automate

Good candidates for test automation include simple, routine and repetitive tests and tests that are impossible to cover manually. Specific examples include:

- **Regression tests**. By definition, you re-test existing features with every release. Planned right, these can be effectively automated.
- **Data driven tests**. These are tests where application functions need to be validated with lots of different inputs and large data sets (i.e. login, search etc.)
- **Load or capacity testing**. These tests typically have to be tested against loads that can only be achieved through simulation. Items that can be simulated are also good candidates for automation.
- **Smoke tests**. These are simple tests developers run with every build to make go/no-go decision on deeper testing.

- **Random tests**.  These are tests that fire large amounts or long sequences of data, transactions or other inputs at a system, in a random search for errors.  These can be easily and effectively automated.
- **Highest Priority features**.  You may want to test these often – making them a good candidate for automation investment.

# Tests that are Not Good Candidates to Automate

Based on industry experiences, the following tests generally do not make for good automation candidates.  In some cases, the business case does not make sense; in others it is just too complicated or not possible at all:

- **New or fast changing applications**.  For applications still under development, or frequently changing, creating automated test scripts may be a waste of time.
- **Subjective validation**.  For application functions that require subjective validation such as usability, simplicity or look-and-feel, manual testing is more appropriate.
- **Localization.**  Testing localized content requires an understanding of the language, culture and local norms.  These are best performed manually.
- **Installation, operations and maintenance.**  These tests typically involve physical installations, adding and removing hardware and interpreting errors codes from multiple vendors.  As would be intuitive, these would be difficult to automate.
- **One-offs.**  The investment in developing test scripts pays of, if the test is repeated many times.  It may not be worthwhile for one-offs.

# Summary

Test automation is an excellent way to achieve more testing and catch more bugs in less time.  Test automation is an excellent complement to manual testing.  The right combination of two will help you release products on time, with improved quality, and without increasing costs.  Given the quick fire product releases of today, some level of test automation may be appropriate most environments.

But test automation brings another level of complexity to the testing process.  It adds costs & time to get started, and requires proper planning, new skills and training.  Testing organizations should do their homework and a cost-benefit analysis before committing themselves to test automation.

# BayTechServices

### When *Good Enough*
### is *Not an Option*

BayTech Services

45 Coventry Lane

North Andover, MA 01845

Phone: 978-852-7019

Send comments to comments@BayTechServices.com

**www.BayTechServices.com**