

## **To Be (Automated) or Not To Be (Manual): A Dilemma for Small Development Shops?**

Automated software testing has long been an integral part of big software development organizations but is often thought to be too expensive or difficult for smaller companies. In many cases, automation has a proven track record in reducing the costs of testing, increasing test coverage and effectiveness, and shortening testing cycles. There are also various intangible benefits associated with automation such as reaching the market faster and improving tester moral (and productivity) through the elimination of tedious tasks.

Automation, however, is not a cure-all for testing problems and can not eliminate the need for manual testing. In fact, for most testing situations there is likely an optimum balance between automated testing and manual testing. The dilemma faced by small development shops with a QA & Testing staff of one or two (or none) is how to determine this balance. Deciding on the types of automation, the areas that can potentially be automated, and the tools and skill levels of the testing resources that will be required is not a simple black and white proposition.

So what is a QA Tester who is already buried in hundreds or thousands of test cases to do? At XBOSoft, we have developed a simple high-level “XBO Automation Self Assessment.” Its purpose is to provide a basis and framework for addressing the question “Should we automate some of our testing?” It doesn’t directly answer the question. The intention is to give people a better idea on deciding if spending more time and effort on the “automation question” is warranted.

The XBO Automation Self Assessment is a simple 4-step process:

- Step 1. Pre-assessment to determine if an automation assessment and automation effort is even worthwhile.
- Step 2. Automation Self Assessment to scope out the likelihood of automation fitting into the testing program.
- Step 3. Automation cost/benefit review.
- Step 4. Next Steps

If after the assessment, it is determined that automation should be pursued further, some factors to consider and suggestions on Next Steps are presented. For example, if there is value to be mined from automating some of your testing, do you have the skills & resources needed to successfully automate? Does outsourcing or working with automation testing experts make sense?

## To Be (Automated) or Not To Be (Manual): A Dilemma for Small Development Shops?

### Step 1 - Pre-Assessment

These are simple “yes or no” questions. A “yes” answer to anyone of them indicates the XBO Automation Self Assessment should be taken. It should be noted that for some of the questions below, test automation is no silver-bullet in and of itself. Automation, however, may be part of the solution.

1. Is your testing mix more than 90% manual?
2. Has the testing activity become deskilled?
3. People are dissatisfied with testing?
4. Is there frustration by the length of time it takes to finish projects?
5. Is there a need to hire additional testing resources, but no budget?
6. Manual testing activity has become tedious and repetitive?

### Step 2 - XBO Automation Self Assessment

The XBO Automation Self Assessment is attached below. It is a questionnaire with “yes or no” answers. This is a scoping effort. The questions selected reflect many “rules of thumb” in the industry. In general, the more “yes” answers, the more potential benefits there are to be realized in automation. In fact, six or more “yes” answers are a good indication that there is latent “automation value” in a testing program.

### Step 3 - Automation Cost/Benefit Review

If the XBO Automation Self Assessment indicates there is potential value in automation, then the next question is “are the benefits in automation worth the effort”? If you know a \$100 dollar bill is lying on the ground, you will surely pick it up if it’s at your feet and be \$100 richer. If, however, the \$100 bill you know is on the ground is at the end of a long and arduous trek, you will think twice if the time and effort to collect it are worth the trouble.

XBO’s recommended automation cost/benefit review is not a rigorous spread-sheet driven return on investment (ROI) analysis. The time and resources generally available to small development shops just doesn’t warrant such detail. We suggest four key areas be addressed; Current Testing (baseline costs), Future Estimated Testing Demand, Estimation of Automation Benefits, and Estimation of Costs to Automate. Again, these would be high-level assessments relying on a mix of actual numbers, “guesstimates”, business plans, and some intuition.

**Current Testing** – the idea here is to establish a baseline of current testing costs. Our recommended approach has two parts: Baseline = actual costs + opportunity costs. “Actual costs” would be the salaries, expenses and overhead of the current testing. This should be a good solid number. What most people often overlook, however, is “opportunity cost”! These are real, but intangible as there isn’t an invoice or salary directly associated with them. Estimating them is a little tricky and entails attaching some value to better product quality and faster testing cycles. For example, by improving

## To Be (Automated) or Not To Be (Manual): A Dilemma for Small Development Shops?

product quality is there an opportunity to grow sales, or improve customer retention, or reduce the number of customer support issues. As a rule of thumb, “opportunity costs” are generally much higher than one would expect!

**Future Estimated Testing Demand** – the objective is to try and determine a “delta” in testing between the current year’s (should be known) and that expected over the next 2-4 year’s. This would be based on information in company business plans, product development plans, etc... Is the company growing such that new QA & Testing specialists will have to be hired? Some key factors to consider when estimating future testing man-hours are:

- Releases (new products, existing product upgrades) planned
- Required regression cycles per year
- Change in existing regression test cases
- Change in configurations to be tested
- Cost of the testing resources (will there be salary raises?)
- Competitive pressures to improve quality, add features, functionality
- Business pressures to reduce cycle times, speed up delivery dates
- Impact of new PC operating systems releases (i.e. Windows 7)?
- Change in the total number of test cases per year

**Automation Benefits** – this is relatively a straight forward proposition and involves estimating the “direct” benefits that can be realized through automation. First, estimate the percentage of test cases that can be considered for automation. A good idea can be gleaned from the results of the XBO Automation Self Assessment. For practical purposes, this should generally be between 10% - 50% of the testing program. Second, adjust the *future estimated testing demand* by the percentage of test cases that can be automated. This should give a ball-park estimate of the direct automation benefits.

It should be noted that automation benefits will also come through the following intangible factors:

- Matured testing process
- Maximized test coverage which will assure the quality of the deliverables consistently
- Elimination of the risk of over sight of quality in repeated tests
- Faster Execution of the tests in multiple environments
- More focus on new features
- Enhanced product quality

As a note: these intangible benefits should be accounted for as “opportunity costs” (i.e. the costs would be lower).

## To Be (Automated) or Not To Be (Manual): A Dilemma for Small Development Shops?

**Costs to Automate** – The key components in estimating the costs to automate are:

1. Test automation tool selection and cost.
2. Time required for customizing the automation tool, building reusable components, automation frameworks and batch scripts
3. Hourly cost per test automation resource

Provided the testing team has the experience to scope out the automation tool(s) to be purchased, assigning a cost is straightforward. The challenge is estimating the resources needed in #2 above. One approach for ball-parking is to reach out to a software testing company and request a proposal. Of course, you need to develop a reasonable set of requirements and scope.

### Step 4 - Next Steps

If there is value to be mined from automating some of your testing, do you have the skills & resources needed to successfully automate? If a team is good at managing testing risk and test coverage and applying the right testing techniques, then talking about automation makes sense.

1. **Test Tool Selection.** The selection of a test tool requires understanding the scope of testing. One should know the expectations from the tools and type of automation targeted. While selecting a test tool, reusability, reliability, and cost need to be factored in the decision.
2. **Automation Framework.** It's vital to consider the effort required for designing test automation frameworks, which will help in reusing components through out the automation and thus help in reducing the script development time.
3. **Test area prioritization.** Identify high risk areas which may be your most critical test cases. Since these may also be the most complex, and thus the most difficult to automate factors such as software functionality being tested, maturity of the feature/function, level of difficulty to automate, must be carefully considered.
4. **Team member assessment.** Lastly, an honest evaluation must be made if there is a team member(s) who is suitable as a tester/programmer who can write test scripts.

If you determined there was real value in automating, but calculating your Costs to Automate really didn't work out or if you know that your testing team really doesn't have the skill and experience to design, build and execute an automated testing program, the next step is to ask ***"Does outsourcing or working with automation testing experts make sense?"*** In many cases, the costs to automate will be much lower working with test automation experts than trying to do in-house. While the hourly cost per test automation resource will be higher through outsourcing, there may likely be no upfront test tool cost and the set-up charges should be much lower due to the efficiency and productivity gains inherent in being a software testing company.

To Be (Automated) or Not To Be (Manual):  
A Dilemma for Small Development Shops?

## Appendix 1

### XBO Test Automation Self Assessment

	Test Automation Self Assessment Survey	Yes	No
1.	Is the test executed more than once?		
2.	Does the test cover an often used feature path?		
3.	Does the test cover a high risk area?		
4.	Is the test impossible or prohibitively expensive to perform manually, such as concurrency or performance-related testing?		
5.	Are there timing-critical components and dependencies that are a must to automate?		
6.	Does the test cover a complex or error prone feature or part of the code?		
7.	Does the test require many data combinations using the same test steps to thoroughly test the feature?		
8.	Are the expected results constant?		
9.	Is the test result analysis time-consuming, such as hundreds of outputs?		
10.	Does the test need to be verified on multiple software and hardware configurations?		
11.	The product being tested is considered “mature” or has certain level of stability.		
12.	Executing the current manual testing is considered difficult.		
13.	Testing includes smoke tests and repetitive verification of the same requirement in multiple places.		
14.	Testing includes “Monkey Tests” that utilize large amounts or long sequences of data, transactions, or other inputs at a system in a random search for errors.		
15.	Testing includes Date and Time handling.		
16.	Functionality testing is currently manual (and stable).		
17.	Testing includes load testing and traversing large amounts of paths through an application.		
18.	Testing involves addressing risks related to concurrency problems, error handling, and feature interactions.		
19.	Testing includes functional regression and confirmation. Rerunning a test against a new release to ensure that behavior remains unbroken—or to confirm a fix did indeed fix the underlying problem.		