



Ten Steps to Building Software Test Automation That Works



Introduction

Take a journey into the exciting world of Quality Assurance Software Test Automation. Let's define where you are today, focus on where you would like to take your organization, and build a plan to get there.

As your tour guide, we've provided you with a map and detailed instructions to help you select the right crew and choose the right means of transportation in order to reach your desired destination safely. We've also provided the tools to measure the effectiveness of your journey and how to tell when you have reached your destination.

Whether you're an Automation beginner or working to further improve your existing test automation efforts, we'll provide helpful tips and ideas in this paper.

We've described the entire road map of automation, including:

- Setting up goals and objectives
- Defining budget and ROI
- Analyzing existing QA and development processes
- Designing your test automation and transition process
- Identifying, hiring, and training automation resources
- Selecting tools
- Promoting internal sale and the automation project
- Achieving immediate results
- Measuring your progress

You need to read this paper if:

- You are pioneering automation implementation
- Your organization made an investment in automation, but it's not getting results that you hoped
- Your automation is doing well, but you feel it can be doing better
- You feel like you achieved significant results with automation, but you would like to promote the results within the organization
- You think that the existing automation process needs retooling
- You bought automation tools and but cannot effectively train the right people
- You are not sure how to chose the right testing tools
- You just started on an automation journey but have limited experience
- Your management would like to see a clear ROI before allocating an automation budget
- You have good experience but would like to learn latest tips and tricks



Table of Contents

Ten Steps to Building Software Test Automation That Works	1
Introduction.....	2
Table of Contents.....	3
Goals and Objectives	4
Defining Budget and ROI.....	6
Analyzing Existing QA and Development Processes.....	8
Designing Your Test Automation and Transition Process	11
Tool Selection.....	13
Identifying, Hiring, and Training Automation Professionals	16
Internal Sale and Promotion of the Automation Project.....	18
Achieve Immediate Benefits.....	19
Measuring Results.....	20
What's Next?	21



Goals and Objectives

Everyone knows it would not be wise to go on a long trip without a map or GPS. It is also essential to know the minor details that are required for you to find your destination. You may need to consider your budget when embarking on a trip.

This is also true in the world of QA Automation. It is, indeed, a long journey. To some software development organizations, it is a journey for the life of the company! Therefore, we think it is important to set time aside to define your goals and objectives for QA Automation.

Let us briefly reiterate the high level benefits of QA Automation:

- Your productivity will increase
- Your time to market will shorten
- Your quality of the product will continue to improve

For different people in your organization, different things will be important. But, it is very important for everyone to understand one common set of goals and objectives to get your team to the right destination in the shortest possible time.

Your Company's goals:

Improving productivity is the most important goal in delivering a software product for internal or external clients. However, you do not have to forget about other benefits of automation such as improving the quality of the product, increasing job satisfaction for internal QA and development teams, and reducing the time required for complete regression of the product.

Who are my customers?

You have to consider two major categories of customers:

- End users of the product and
- The internal development organization who will be on the receiving end of the results of your accomplishments

Your Customers' goals:

With their goals in mind, you can determine the goals of each of these categories:

End users:

- Reduced errors in the product
- Regular updates without braking existing functionality
- Product that works reliably with any third party solutions or plug-ins



Internal development organization:

- Shorten development / release cycle
- Streamline development and testing: eliminate bottlenecks, firefighting, or slipping deadlines
- Improve productivity of development and QA team
- Save money

Defining Budget and ROI

The majority of organizations require an estimate of savings or returns on investments as part of the budget approval process. Automation projects are no exception. In order to calculate a Return on Investment (ROI), let's start with the savings.

It may be difficult to determine your expected level of results without some outside assistance. Because there is no QA Automation Handbook, most companies who are investigating QA Automation are not even aware of:

- The level of success to expect
- The speed of results attainable
- The importance of an effective methodology
- The importance of designing software for the ease of automation
- The impact on the development process
- The potential benefits to the bottom line

Savings

Here are the some areas to consider for savings.

1. Lessen the need for repetitive manual testing
2. Compress the release cycle time
3. Slash the number of defects released to the field
4. Considerably cut the time developers spend fixing bugs reported from the field
5. Reduce the number of fixes or interim releases required by the field
6. Decrease the number of helpline and maintenance personnel
7. Increase the productivity of the development group

The reduction of repetitive manual testing is a tangible value to measure. By replacing repetitive manual testing with automation, savings in-person-hours may be realized. But this may not be an accurate realization if an inadequate amount of testing is currently being completed.

MYTH:

Manual testing will be eliminated by implementing automated testing.

REALITY:

The reality is that no organization will be able to eliminate the need for manual testing. However, if test automation is implemented successfully, major accomplishments can be achieved:

- **Manual testing team will be able to focus on the new features**



- **Manual testing team will be able to spend time writing new test plans to be automated later**
- **Manual testing team will be analyzing results of automation test runs**
- **Manual testing team will not be blamed for being a bottleneck to the release process**

It is often difficult to specific dollar savings when considering the manual testing process. Not only will reductions in repetitive manual testing produce measurable savings, but the reduction in the test time required will permit running the test more frequently. This will impact savings in other areas especially the cycle time and reduction of defects.

Investment

Developing test automation that works requires an investment. Below are some of the cost elements that need to be budgeted.

- Tool Licenses
- Additional headcount including benefits and any overhead
- Hardware
- Outside services
- Classroom training
- Mentoring of resources

Once the savings and the costs are defined, the numbers can be plugged into a ROI formula to calculate your return on investment. Below is a simple ROI calculation formula:

ROI = savings divided by the cost of automation,

Where savings = Cost of manual testing minus the cost of automated testing

For a FREE ROI calculator developed by using data from a number of actual client cases, please visit the following link: <http://www.qasignature.com/resources>

Analyzing Existing QA and Development Processes

First and foremost, your existing situation should be evaluated in order to determine what needs to be modified or added to today's QA and Development processes. We advise you to put together a matrix or a questionnaire and engage in a number of short conversations with key people in your organization. Depending on your organization's size and structure, the levels and titles of these people may vary, but we generally recommend talking to:

- VP of Engineering
- Development Manager / individual contributors
- QA Manger / individual contributors
- Release Manager
- Tech Support Manager / individual contributors

The answers to the questions below should be collected.

1. How does your development cycle look?
 - Flip to the [next section](#) and see which of the diagrams is a better representation of your development cycle: "Diagram 1" or "Diagram 2"?
2. If the answer is the "Diagram 1," then most likely you will hear "yes" when you talk to the members of your QA and Development teams. Ask them the following:
 - Does QA feel that:
 - They always find themselves at the end of the process?
 - The application is given to them for testing when all possible deadlines for development are blown?
 - There is never enough time for testing the release properly?
 - They are the ones who are being blamed for a late release?
 - Do developers feel that:
 - The defects that are given to them to fix should have been caught much earlier in the development life cycle?
 - There are bugs coming from the field that by all means should have been caught during release?
 - They are being asked to redo their work many times for no obvious reason?

- The problems that QA reports are very often impossible to re-create?

3. Specific data (averages):

a. Releases:

- Frequency of releases (number of releases per year)
- Duration of development cycle (from start of development to turning it over to QA)
- Duration of testing cycle (from start of testing to giving it back to development)
- Number of iterations (how many times the “b” and “c” above are repeated before the product is released)
- Duration of the whole release cycle
- Deadline slippage (average number of days / weeks a release is late)
- Number of emergency releases / patches per year

b. Builds:

- Do you have an established version control system and process in place?
- How often do you build your application (daily, weekly, monthly, every three months)?

c. Resources:

- Who is doing testing?
- Do you have a dedicated QA team?
- Do you have or plan to have a dedicated QA Automation team or resource?
- Are your QA resources being asked to work in other areas not related to QA?
- Are your Test Automation Resources being asked to do manual testing?
- Are you using other team members (not QA – such as developers, tech support personnel) to test your application?

d. Environment / Hardware:

- Do you have a dedicated test system?
- Is the above test environment being used exclusively for testing?
- Does your testing ever happen on a live environment?
- Do you have or plan to have a dedicated Test Automation System?
- Is the Test Automation System ever used for anything other than test automation (for example – for manual testing)?

e. Statistics: collect the following information on a consistent basis:



- Bugs found during release:
 - Total Number
 - Effort to find them - QA (resource–days)
 - Effort to fix them - Development (resource–days)

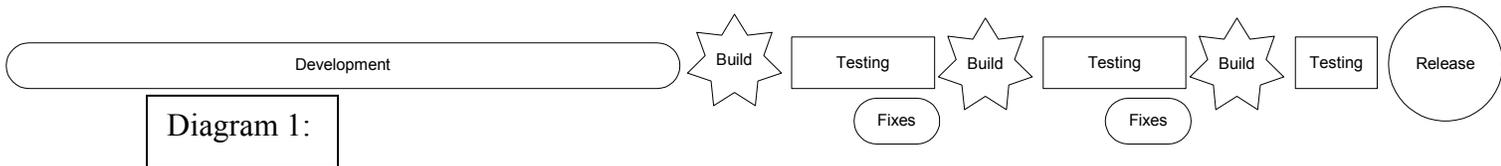
- Bugs from the field found after release:
 - Total Number
 - Effort to find them - QA & Tech Support Team (resource–days)
 - Effort to fix them - Development (resource–days)

The collected information will indicate symptoms of problems you can attack with effective adjustment of your processes. It will also show the direction to proceed. This information will be your benchmark to compare against when you are at the point where test automation is successfully implemented and starts bringing results. The consistent collection of data will increase your ability to justify your efforts and to calculate your project's ROI.

Designing Your Test Automation and Transition Process

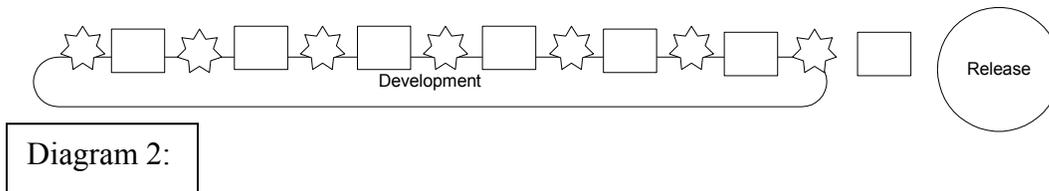
Traditional Software Development Environment

Diagram one shows a traditional software environment where the bulk of development is completed prior to delegating the application to the testing team. This team is often made up of volunteers, subcontractors, or borrowed personnel from other departments in the organization. The crash exercise is performed just before the product is released to the field. The test team does whatever it takes to ensure a smooth automation process. This may require working double shifts, week-ends and holidays. Defects are recorded and fixes are provided for the defects found. The lack of time allows, at best, a quick verification of the fixes. However, there never seems to be time to retest the entire product to determine if the fix affected any other function in the software. The sad reality is that clients end up finding many of the problems.



Automated Test Environment

In an automated testing environment as shown in Diagram two, the product is tested whenever there is a new build or, ideally, daily. The frequency is often increased as the release date approaches. This is dependent on the development method being deployed. Defects are posted, fixed, and re-tested on a daily basis. The complete test is run to flush out any problems introduced by the fixes.



The end result is that product quality is not compromised by the inability and lack of time to perform adequate testing.

Where does your process fit in the examples above?

The first step in the transition process is to determine where you are today based on the diagrams above. Making the transition to an automated test environment requires change not only to the way testing is performed, but to the development processes as well.

The 3 major areas of change are to:



1. People
2. Processes
3. Technology

People

The roles and responsibilities of the people will change. The fear of the unknown is minimized as you begin to define future changes.

Processes

Automating the testing will impact the overall testing processes. Laying out the high level changes to the testing processes will ease the transition process.

Technology

The new technology imposed by automating the testing may change the communication of product defects to the developers.

Here are several steps that can be taken to get the process started:

1. Organize a team made up of product management, development, and quality assurance
2. Document, at a high level, the current development and testing processes
3. Prioritize the test cases to be automated
4. Modify roles and responsibilities
5. Map out transition processes realizing the change will take place over several months.

Tool Selection

The next logical step in your transition to test automation is to select the right tool for your people, processes, and technology.

The process of determining the tool best suited to your needs is not an easy one. Each organization will have different needs that should be met by software test automation. The testing of various applications and operating systems used requires a complicated search for the appropriate tools. A simple set of steps to evaluate vendor tools can be followed to determine the correct tool for the job.

The following are recommended steps to follow for a software test automation tool selection.

1. Define your requirements (technology, hardware, and software constraints)
2. Compile a possible list of candidates
3. Review vendor requirements
4. Make your selection

Before looking at testing tools make a shortlist of the requirements you have for software testing. What problems will the tool solve? What technical capabilities will the tool need to be compatible?

Test Requirements:

What types of testing problems do you want the tool to address? Do unique issues exist? Make a list of these special requirements to be addressed in the evaluation.

Technology:

Any testing tool you choose must be compatible with:

- Operating systems your application supports
- Development environments used to create your application
- Third party software with which your application integrates
- Object oriented testing capabilities
- Open Source vs. Commercial choice of tool

****Note**

Many managers are tempted to save on the license cost and pick one of the free or open source tools that are readily available. Often, the consequences of this decision are detrimental.

- Most of these tools have very little or no support.

- Sometimes the installation and configuration of a tool takes a significantly larger amount of hours as opposed to the relatively easy, fast, and friendly procedures offered by commercial tools.
- We have evaluated more than a hundred of such tools and have found: free or open source tools are less user friendly.
- Few of the free tools are as powerful as commercial tools. However, they are flexible making it possible to modify and expand.
- Open source tools can require a significantly larger amount of effort to achieve desired results. A resource may spend several months building the tool to meet the required functionality. One or two licenses of a commercial tool may be a fraction of the money spent by the company compared to the development time.

List of Possible Tools

You can develop a short list of vendors by consulting with experts who have automated software with similar requirements. Find an objective expert who is not a re-seller of any particular tool.

Test Tool Choice Considerations

Review the following:

- Vendor references
- Other customers with similar requirements
- Skill level required to use the tool
- Multiple user access
- Support and help documentation
- Integration with other tools
- Budget requirements for:
 - Tool purchase
 - Training, implementation and on-going support

Test Tool Selection - Feature comparison and Proof of concept

Run all your tests with each of the tools on your short list using the same test scripts. If you are not comfortable running the tests, an objective third party vendor may be a viable option. The vendor should have experience with the short list of tools.

The *proof of concept* will make the selection quite clear. With the selection made, you are ready to take the next step.

MYTH:

There is a "perfect" tool out there to solve all my problems and will result in automation without any need to write additional code.



REALITY:

The reality is that there are more than 120 widely known tools on the market with new tools coming out every day. It is a lifelong project to know all of them. Realistically, there is no perfect tool. Any tool that you choose will still require effort and a proven methodology in order to build solid automation. Focus on the goal and results will come!

Identifying, Hiring and Training Automation Professionals

Find a unique combination of skills and attitudes when you look at potential test automation resources.

QA automation skills required:

- **Technical proficiency** – Writing test scripts is essentially a development task and all code development practices and standards are more than applicable here. The script code has to be easily managed, maintained, and well commented. There should be a number of established processes and procedures in place, such as - name conventions, code reviews, design meetings, source / version control, scripts, and back-up. If your test automation specialist is also responsible to set up all the above, then that person should have a significant technical background. A bachelor's degree in computer science with business skills is recommended.
- **QA engineer mentality** - Quite frankly, not many developers are eager to take on the load of being responsible for a QA automation effort. A good QA person focuses on finding the most efficient coverage of the test and is tasked with locating hidden problems. Developers, in general, are more inclined to think in terms of code efficiency (in this case, the test script code efficiency). When it comes to automated testing, the goal is to develop test scripts that will be very easily modified to quickly accommodate changes in the application under test. It is not the goal to test the efficiency of the script code itself.

Hiring outside vs. training internal candidates

Based on the suggested skill requirements you will make the determination of whether to hire an outside vendor or train internal resources. In either case, adherence to and the training on an automation development methodology is crucial to long-term viability of the **automation**.

Training - The importance of a Development Methodology

The importance of an automation development methodology cannot be understated. It is recommended that developers of automation are proficient in the capabilities of the tool set being deployed. Training on the tools is a pre-requisite to the training to follow. The tool vendors do a good job training on the use of tool sets but lack in the training necessary to build efficient and maintainable automation and development architecture. The architecture needs to support development efficiency but, more importantly, must be efficient to maintain.

MYTH:

Send the entire QA department for tool training and you will have test automation in no time!

REALITY:

Having a good tool and people that know it inside and out is one of the components of success. However, if other aspects such as maintainability are forgotten, in 6 months you might find yourself looking for another "better" tool.



The first tool may never be implemented and become “shelfware”. According to the tool vendors, about 80% of automated tools sold are never implemented. This paper is designed to help you become one of the 20% of organizations who succeed.

Training on the Development of Test Automation

The student should learn to create test scripts that are more robust, durable, and easy to maintain. In addition, they should be trained extensively to interpret the results.

Automation Essentials Course Outline

Pre-requisite: Tool Overview Training includes:

1. Automation architecture
2. Approach and development methodology
3. Essentials of automation
4. Execution of automation scripts
5. Interpret the results

On The Job Training

On the job training should begin after Tool Overview Training and the Automation Methodology Essentials class. The training should start off with mentoring sessions and evolve into one-on-one assistance. This schedule will allow the user to develop automation while utilizing the support of experienced developers for on the job assistance.



Internal Sale and Promotion of the Automation Project

It will be easy to convince senior management to endorse an automation project if you have solid budget and ROI numbers in hand. The quality assurance team may pose more challenges. Their resistance is easily addressed by communicating the changes ahead. Redundant manual testing can be traded in for more exciting work.

Start by automating the “critical path” or the main highway through the product and prioritize by measuring the risk and time. Automate the redundant tests first and give a higher priority to the long test time items because when time deadlines approach, these tests are more likely to be cut short. As the automation coverage is expanded, the QA professional’s time is freed up to concentrate on writing test scripts and testing new functionality in the product.

In an automated test environment, once the test is written the computer performs the testing and the manual test requirement is significantly reduced. The first run of any test should be preformed manually – then the tests are automated. The test can be run as frequently as necessary with no additional cost. The QA professionals become the Champions of the test process by eliminating the bottleneck.



Achieve Immediate Benefits

One of the weaknesses of standard QA Automation Methodologies is the upfront time and effort required before being able to write a single line of automation code. Most Automation Consultants believe they must spend months learning the product before they can get started. They may not even care how long it takes if they are billing on a time and material model.

Development and Maintenance of Automated Test Scripts Should Be Done in Parallel with Product Development.

The Test Automation Methodology should support live test automation within the first two weeks of the project. There is no need to spend weeks or months learning the software application. The methodology should support the development of test automation simultaneously with the development of the product. It is of paramount importance that the methodology supports the ease of maintenance of the automation scripts as more scripts are developed.



Measuring Results

An unlimited number of metrics could be put in place to measure the corresponding improvement from automating the quality assurance testing process. The metrics below were used by a client over a 24-month period. The numbers were input on a release frequency. Progress was observed over this period.

	Release Number						
1	Frequency of Builds						
2	Frequency of Automation						
3	Frequency of Bug Fixes						
4	Duration of Release (Weeks)						
5	Total Number of People (Dev/QA)						
6	Count of Modification Requests (MRs)*						
7	MRs Per Month						
8	MRs Per Person Per Month						
9	Testing Time/Release						
10	Coverage %						

* Modification requests include defects, enhancements and new features.

With automation you will begin to see the improvement in the productivity of the teams as well as the quality of the product.

By using this table and ROI calculator that you download from <http://www.qasignature.com/resources> you will be able to measure your success over time.



What's Next?

In order for your journey to be a successful, you need to make the right choice of whether to go on this trip alone or to rely on outside help to navigate you successfully through turbulent waters. In either case, no matter how you do it, on your own, using outside consultants or a test automation company, the basic steps remain the same.

Just make sure that whoever is implementing automation for your company is focused on the results that are important.

Safe journey!

Contact us:

Since 2001 qaSignature helped more than 50 software development organizations to implement automaton that works.

Please call us today if you would like have an evaluation done on how fast test automation that produces a 200% ROI in the first year can be implemented.

(857) 229-1060 ext.10
qaSignature, Inc.
200 Wells Ave., Suite 201
Newton, MA 02459
www.qaSignature.com