

# Project Manager's 101 Guide for Test Automation Projects

*Shrini Kulkarni*

*Principal Consultant – Testing*

*iGATE Global Solutions*

*Bangalore*

[shrini@igate.com](mailto:shrini@igate.com)

[www.igate.com](http://www.igate.com)

## Table of Contents

Introduction.....	3
Background – Legacy of failed Automation Projects.....	3
Context .....	5
Automation readiness Assessment.....	6
Development Model .....	6
Concept of Automation Ecosystem.....	7
Building Blocks of an Automation Ecosystem .....	7
Process Model.....	8
Test Design and Automation.....	9
Test Case Analysis .....	11
Resourcing – Hiring for Automation skills .....	11
Economic Aspects of Automation .....	12
Bonus Tip – Big Secretes about Automation .....	13
Conclusion.....	13
References .....	14
Appendix A 10 Classic Reasons why Test automation projects fail .....	15
About the author – Shrini Kulkarni .....	15

# Introduction

Any *serious* test automation is a software development activity but of a different type. Test automation is a valuable addition to a tester's toolbox to enhance the reach of testing. It does not and can not replace a human tester, particularly at the end-user level. Most test automation efforts fail because they don't take software development architecture into account, they don't plan for maintenance, and they tend to be understaffed, and are often staffed by non-programmers. Over last 2 decades, various authors and practitioners have cautioned against taking a shortcut for deploying automation solutions. Few parameters and aspects like – “scope of work”, “staffing requirements”, “Automation tool”, “Acceptance criteria”, “Identifying and managing stakeholders”, “Approach to automation” etc., make automation projects to be treated differently than other software projects. Thus managing test automation project presents a unique and interesting challenge.

Test automation is a very cool tool or means of supplementing your manual testing. If properly applied, it can bring in some cost advantages, appropriate reductions test cycle time, working around human limitations of boredom, fatigue and many others. In today's scenario of Enterprise IT and software product engineering, managers have been looking for ways improve and optimize costs associated with software testing and are typically seen zeroing on test automation. We have a legacy of incorrect application of Test automation and history of failed or abandoned automation initiatives.

Unique challenges, structure, dependencies related to automation projects have prompted the author to study various aspects that make automation projects special. The purpose of this paper is to serve as a guide to project managers managing automation projects. The paper will touch upon various aspects of automation projects and provides tips and guidelines for successful management of the project.

## Background – Legacy of failed Automation Projects

One of the oldest references with respect to uniqueness of Test automation, myths and challenges comes from James Bach [1]. A close look at the observations made in the document indicates that the failures were due to various reasons that were external to the system of Test automation. Like business expectations, dependency on third party software products, Technical challenges from allied groups – like development and testing, etc.

Bret Pettichord [2], points out common problems associated with Test automation projects. In his experience of working software companies of different size and cultures, Bret, recognizes many common things that lead to failures. It is very important to note

the observations on people issues, business expectations and loss focus on “testing”. Reluctance to think about testing – is a noteworthy observation. Automation engineers might find automation more interesting than testing and for the management sets its eye firmly getting benefits (as promised by automation tool vendor) – while slowly drifting away from core work of Testing. Among the people issues, dumping the work of automation on test engineers as a part time job (when there is no testing happening) is the big one.

Cem Kaner [3] lists a specific list of things that a manager needs to take care of – for automation projects. Words of caution are right there at the start of the paper – “Companies abandon automation efforts after failing to achieve bug-effective and cost effective automation. Success is possible but far from assured”.

Shrini Kulkarni [5] lists down 10 classic reasons why test automation projects fail. The author presents a concise list of reasons leading to failures in automation – with the top one being “a simplistic model used for testing”.

Elisabeth Hendrickson [4] mentions about ‘the differences between automation success and failure’. A failure in test automation can manifest in a number of ways –

- Wasted time and money
- Inaccurate results
- Demoralized team
- Overall reduced productivity

Another common symptom of failure would be the “automation abandoned indefinitely” as another sign of test automation failure. Elisabeth further lists following characteristics of a failed automation project.

- Stated goal "automate everything"
- Unstated goal "Reduce the number of Testers"
- Goals are not measurable
- Executives expect immediate payback
- Automation lead is inexperienced technically.
- Inadequate communication with executives
- Inadequate communication with manual testers
- Extremely limited test documentation
- Automation team did not appreciate the value of manual testing
- Manual testers felt threatened

We have a legacy of incorrect application of test automation and history of failed or abandoned automation initiatives. In spite of repeated focus and emphasis on holistic

approach to automation by software testing and automation practitioners, the trend continues. All of this would set context – that project manager when starts off an automation project, should be aware of the fact that the ground is slippery and there is a bad weather too. So caution is the keyword.

## Context

Typically, a project manager for an automation project can be seen working with any of following two contexts.

One – An IT organization with a diversified IT application portfolio supporting critical business functions, Software Testing function mainly understaffed, manned by contract resources, Development Team not equipped/inclined to implement/support Application Testability features, Mandate for Cost effectiveness (read offshoring), shrinking budgets to Testing and One or more costly test automation tools on the shelf (waiting to be used) or buying/Demo/Proof of concept in progress to implement automation

Second – A software product development group that has an automation team, there is a need to develop automated test suite for core set of product features. Objectives of automation are to facilitate daily build and test, test more often, facilitate easier introduction of new features and bug fixes. Automation happens typically more at unit level /API level and less at GUI level. Mostly in-house tools are used.

While the former one is more common in the industry, the latter one is seen product development companies. As we can see from these two contexts that there are no universal objectives about Automation, stakeholders, depending upon the situation, decide the objectives for the automation program. Key to successful management of automation project is to understand “stakeholders” view point.

*Due to the holistic, systemic approach necessary to be successful in QA, one should not discuss any one approach without a good understanding of how that approach ties into the others, and how you have to leverage all of them to succeed. The one thing I can pretty much guarantee is that manual testing ad hoc testing, or automated testing will not make anyone successful long term if only one of them is relied on - Anonymous*

- ***Mission of Testing Decides EVERYTHING for Testers***
- ***Testing mission sets the CONTEXT***
- ***Testers should always focus and align their Testing to the Mission given to them.***
- ***Automation engineers help Testers in achieving their mission***

## Automation readiness Assessment

“Are you ready for automation?” is the first question that an automation consultant needs to ask a client or a stakeholder who is interested in kicking off an automation initiative. As one discovers later, forcing automation on a project team which is not prepared for automation can result in problems. Automation readiness can be measured in terms of Availability of Test cases in a common repository, Test environment, Test data, Product release plan, expectations from automation, availability of Tool licenses and other infrastructure requirements etc. An alert PM would insist on doing a readiness check on both automation and client side to ensure that both are geared up for the journey. Automation readiness exercise would also enable to identify some possible bottlenecks or road blocks so that both client and automation team is aware of them and can plan for mitigating the issues arising out of them.

## Development Model

A Development model for an automation project in its simplicity would be like an IDE (integrated development environment). It is a core structure on which automation code is developed. In some cases, a development model is also referred as “Automation framework”

The project manager needs to be aware of development practices for automation, are to be built on the principles of “Holistic Approach to Test automation” as defined bellow.

1. Test automation is Software Development
2. Success of Automation is highly influenced by clear understanding of objectives of automation and Management commitment to the initiative – Start with clear Definition of What a successful Automation means.
3. Test Automation can not pay immediately
4. For Sound Test automation, sound Manual Testing infrastructure is a Pre-requisite
5. Test automation will NOT reduce the manual Testing effort but will free up manual testers “mundane” and “routine” testing tasks, hence enhancing the utilization of Manual Testing resources
6. Test Automation will not improve Testing process (In fact Manual testing process is pre-requisite for successful Automation)
7. 100% automation is not only “unachievable” and when achieved will be a costly to change and maintain.
8. Good amount of collaboration with Product development team, manual Test team and an application with built in testability feature, is key to cost effective automation.

9. Test Design needs to be aligned with Automation – Retrofitting available manual Test cases is not an effective way to Automate.
10. Automation tool is only one entity in overall ecosystem – Supplement it by building a Framework.
11. Identification and sensitivity to the factors that are external to automation and can still have a significant impact on the success of automation
  - a. Application Testability Features – Poor application testability reduces Automation coverage
  - b. Developers Willingness and support
  - c. Manual Testers Support
  - d. Test Design methodologies
  - e. Re-engineering efforts for Test cases
  - f. Creation and maintenance of Test case Repository

## Concept of Automation Ecosystem

Test automation does not happen in any isolated system – there are many related entities to be considered so that Automation as an improvement initiative gets implemented and can sustain changes. Tool needs to integrate with rest of components. Just like a biological ecosystem, Test automation works in an ecosystem consisting of various interdependent entities with dynamic interactions. Also any action taken in the automation ecosystem (e.g. pesticide in food chain) has a potential domino effect on other entity of the system. Entities Automation eco system will have symbiotic relationships among themselves.

A Test Automation ecosystem is assemblage of Software Engineering entities like components, Platform, Tools, People and processes (development and Testing) living together and having symbiotic relationships

It is this ecosystem concept that makes any Test Automation Program – complex to design, implement and manage.

### Building Blocks of an Automation Ecosystem

Testing tool alone will not solve Testing problems. In most of the cases, one tool will not be sufficient for the purpose. What we need a “Test Automation Tool box” – An automation Framework, a collection of wide variation of tools.

#### Project Management Block

1. Project Plan
2. Quality Plan

3. Configuration Management Plan
4. Schedule/Work break down structure
5. Issue Tracker
6. Project status Review
7. Estimation model
8. Productivity and Progress Tracker

### **Development Block**

1. Script/Module Templates
2. Coding Guidelines
3. Code Review Process/Guidelines
4. Test analysis template
5. TALC model - Automation Lifecycle

### **Deployment Block**

1. Target Test Platform - Environment Setup
2. Reporting System
3. Scheduling system
4. Test data Repository
5. Execution Engine
  - a. Remote Execution
  - b. Centralized control
  - c. Integration with Test Management Software

## **Process Model**

Test Automation projects are software development projects – but of different type. When executed separately, need to follow some structured process for successful completion. Like in software development, Automation project also will have phases, milestones and deliverables. Key to success in automation is to design, develop, implement and maintain one such model for a given organization. The model needs to specifically address unique characteristics, requirements, culture and other things. It is recommended to not to just blindly follow an “industry” standard model without evaluating the suitability of the model to the group and organization.

A process model would help the automation team to plan the project activities, provides a wireframe structure on which specific project artifacts can be laid out and delivered. The process model would define the roles and responsibilities of resources in the project and the protocols for communication with external teams and individuals. A

process model would also help the team with a tool kit for performing routine tasks of the projects, there by bringing a consistency and uniformity of deliverables.

Following are few general requirements that the automation process model needs to address.

1. A process module shall identify various phases in automation development, corresponding milestones, gates, deliverables and entry/Exit criteria at each of these phases.
2. The model shall be customizable to cater to specific needs to a client situation, application under test and project environment.
3. The process model shall explicitly identify the dependencies for the project and provide a mechanism to track these dependencies throughout the project
4. The model should be flexible to suit the needs of changing project environment like - change in tool, change project stakeholders, change in requirements and project charter etc.
5. The model needs to explicitly call out the parameters that impact the project success and suggest ways to track these parameters.

## Test Design and Automation

Test design and documentation is the single most parameter that decides the coverage of Automation. Ideally, writing test cases especially for automation would be good thing so that test designer can think of ways to facilitate the programmatic implementation of a test. However in many cases, due to project constraints and other context related issues, it would not be possible follow this route. In that case, an automation team would be typically handed over a list of test case designed previously by a manual test team and automation team would automation a subset of these test cases based on the criticality and importance of test as rated by various stakeholders. Thus the approach of test/automation design shifts to “selection of test cases”

Selection of test cases for automation is the major activity in Automation lifecycle. A mature PM of Automation would spend adequate time is set for proper analysis of test cases and check for each one the value of automation vs. the cost of automation once the technical feasibility is cleared.

**A Good Manual Test is closer to Humans and represents their usage pattern.**

**A Good Automated Test is closer to Computers and is similar to application code.**

Manual Tests suffer from following problems when taken up for automation

1. A style that is not close to what computers can understand - Mostly (forcefully) written in a language that any unskilled tester – can understand
2. Some of them are OUTDATED - Most of them need updates to suit with current set of Application features.
3. WRITEN with lots of manual element – phrases like “Valid or proper results are displayed”
4. Extensively documented to make regression easier – Test design style and steps are not optimized for automation. Results in lots of repetitive steps and verifications and weaker automation suite.
5. Test Data issues – manual Tests do not identify the test data to be used
6. Highly Brittle – Too much tied to the flow, UI etc – difficult to maintain and change
7. Not Suited for Automation
  - Not Data Driven

In order to design a manual test that is useful for automation, you have to design it to be not like a realistic manual test...which means when you are doing it/thinking about it, you have to be thinking or operating in a way different than a good manual tester would. When manual testers try to create manual tests that are good for automation, they end up creating manual tests that aren't good manual tests.

*- David Gilbert*

*Life seen through the eyes of an automation engineer has very little in common with life seen through the eyes of a real user (or a manual Tester)*

*- David Gilbert*

# Test Case Analysis

The word Test case analysis is applicable in those contexts where the automation is given a set of test cases and asked to automate a subset of them. Test case analysis phase in Automation lifecycle aims at building a robust Automation test suite based on the sound programming principles of modularity, re-usability and extendibility.

In a typical GUI regression automation scenario, following are the tasks performed as the part of Test case analysis

- Manual Execution to understand the flow
- Identify and tabulate the Test data Required for the Test case
- Logical Break up of Test case steps into Navigation and Verification routines
- High level ideas for navigation and utility routines
- Identify Setup and clean up routines – for Non Read-only Type actions
- Create pseudo code – skeleton for the test
- List of Navigation and Utility functions
- Record/Identify various GUI controls in the test case and record issues with them, if any

## Resourcing – Hiring for Automation skills

One of the major challenges of setting up of an automation team for the project manager is hiring human resources for the team with appropriate skills. While skills specific to usage of the automation tool selected for the project is essential, an automation engineer is also required to have other supplementary skills in development methodologies and Testing in general. Hiring resources with only tool experience can severely impact the team's ability to develop maintainable code and also the team's ability to look for solutions in case there are tool limitations.

Following list is intended to be used as a guide for hiring automation resources for the team.

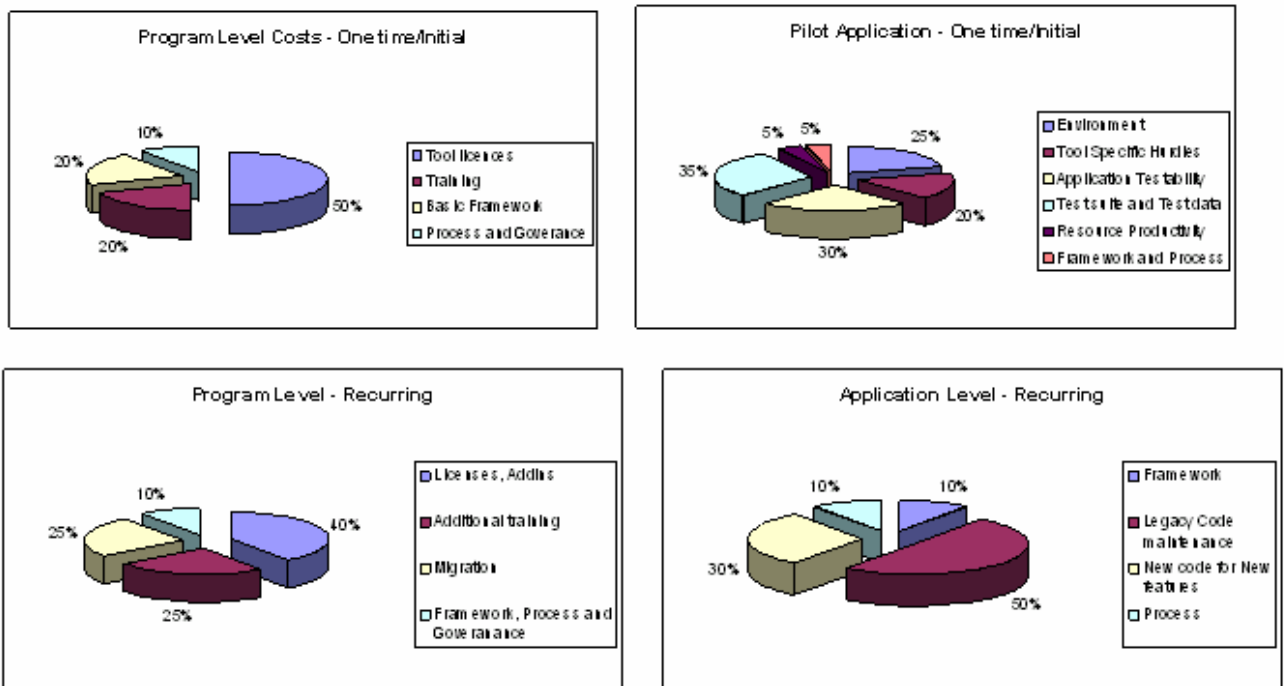
1. Good automation is software development – An automation engineer, at a high level should understand and appreciate general programming and development principles and should be able quickly grasp the concepts.
2. Prior programming experience would be a huge advantage
3. Should have a good analytical/out of the box thinking abilities in order to solve and work around typical “non standard” test features and

4. should possess basic testing skills and ability to think through the eyes of a tester and should be able to tailor the solutions suited to various testing needs
5. Should be good at test tools – a wide variety of them, should know about major drawbacks of each of them. Affinity to a particular tool should not become a hindrance to development of variety of solutions using alternative methods.
6. A project lead or a PM, in addition to above qualities needs to have general project management skills like project tracking, issue resolution, communication to stakeholders, deliverables management etc.

## Economic Aspects of Automation

Automation is initial investment heavy exercise. If there is one single most component that impacts the “economics” of automation that would be “Maintenance cost of Automation”. It will be more accurate if this term is generalized to include all “recurring costs” associated with owning an automated test suite or solution.

Following picture shows various cost categories associated with Automation. The project manager needs to factor all these into overall business case associated with the automation and make sure that stakeholders understand ROI model and set realistic expectations with respect to benefits of automation.



## Bonus Tip – Big Secretes about Automation

1. It does same thing again and again and again with same sprit, energy and speed (of course limited to the capabilities of Computer, network and other platform components). Is it Good or Bad?
  - a. Good – for that part of Testing that is Repetitive (highly) (Ask yourself what part of your Testing is that way? 5% or 10% or 20%)? You can reap benefits only to that extent.
  - b. Good for Accuracy – what part of testing requires an accurate yet VERY CLEARLY defined criteria that can be effectively translated into computer instruction?
  - c. Good For speed and luxury of executing in ODD/OFF hours
  - d. Bad – it can not find bugs for you.
  
2. Automated Tests are poor at making observations, many of them, simultaneously. Effective manual tests involve observing the entire system, from colors to response times to CPU usage, all more or less simultaneously. Automated tests are very poor at this sort of observation

Automation is just a TOOL. It'll get you some places, but not all. You can use Mercedes to haul dirt and use an excavator or a Road roller to go office – but are they good use of them? Automation is a HOW. Arguably a really, really good one, but it is not an objective in and of itself.

## Conclusion

Managing Test Automation project, for a project manager, presents a unique set of challenges and hence opportunities. Understanding the project environment, tools, people and infrastructure issues will key to successful management of automation projects. And then there are business and economic aspects of expectations from Automation. With the legacy of many failed automation projects and challenges observed in them, a project manager needs to be aware overall dynamics of project environment and be sensitive to the various dependencies in Automation ecosystem. The PM needs ideas, tools, references and supporting structure for managing such complex system. This paper presents some of the tools and ideas in that direction.

## References

1. Bach, James (1995, October) "Test Automation Snake Oil," *Windows Tech Journal*.  
[Web link](#)
2. Pettichord, Bret (2001, January) "Seven Steps to Test automation Success" [web link](#)
3. Kaner, Cem (1998) "Avoiding Shelf ware: A Managers' View of Automated GUI Testing" [Web link](#)
4. Hendrickson, Elisabeth (1998, October) "The differences between Automation Failure and Success" – [Web link](#)
5. Kulkarni, Shrini (2006, Feb) "10 classic Reasons why Test automation projects fail"
6. Merick, Brian "When should a test be automated" [Web link](#)
7. Bach, James (2006, July) "Manual Tests Can not be automated" – Blog post  
<http://www.satisfice.com/blog/archives/58#comments>
8. Pettichord, Bret. 1996. "Success with Test Automation." Presented at Quality Week (May). <http://www.io.com/~wazmo/succpap.htm>
9. Dustin, Elfriede. 1999. "Lessons in Test Automation." *Software Testing and Quality Engineering* (September): 16–22.  
<http://www.stickyminds.com/sitewide.asp?ObjectId=1802&ObjectType=ART&Function=edetail>

## Appendix A 10 Classic Reasons why Test automation projects fail

Sl No	Reason/Issue/Notion/Myth
10	No clear objectives for automation and Lack of Management commitment
9	Expecting Immediate Payback.
8	Lack of methodology to assess the success of the project.
7	Notion that Turn Key automation - Scripting or Just record and playback - Non programmers, Non Testers can do automation
6	Wrong selection of Test cases or using a Test design that is not suitable for Automation
5	Notion that Automation replaces Human Testing
4	No clarity about real cost involved in automation.
3	Aim for 100% Automation (Notion of More is better)
2	Not treating Automation as software development.
1	Treating Software Testing as Sequence of Automatable actions and as "Last phase in development cycle" or "some formality" to be completed so that Application can be deployed in production

### About the author – Shrini Kulkarni

Shrinivas Kulkarni (a.k.a Shrini) is Test evangelist, speaker and a passionate tester. Shrini is currently working as Principal Consultant – Testing at iGATE Global Solutions, Bangalore for their Independent Verification and Validation (IV&V) division. In an IT career spread over 10+ years, Shrini has worked in companies like Microsoft Hyderabad, Aditi Technologies Bangalore, i2 technologies Bangalore. He consulted, architected and delivered many software Testing and automation projects in various business domains like Banking, Finance, Legal. Shrini worked in many roles covering entire spectrum of IT

-like Software developer, Project lead, Configuration Manager, SQA lead, Test Lead, Test Manager.

As a certified Java programmer (SCJP2) and a certified Software Test Engineer (CSTE – QAI), Shrini holds a blend of software development and testing. He has an M Tech in Mechanical Engg from IIT Chennai, India. Shrini has been a regular speaker in Software testing conferences in India and abroad.

Shrini's interests in software testing include – Test automation, Test process maturity, Security Testing, Test management, Test Estimation etc. Shrini can be reached at [shrini@gmail.com](mailto:shrini@gmail.com) and read about Shrini's ideas and thoughts on testing and automation at <http://shrini.blogspot.com>