



# MART Pro-Service Solutions

Why Automation Projects Fail





# Software Automation

- MART Pro-Service Solutions is a leading independent test and interoperability testing facility offering a complete suite of testing services.
  - Test Automation Consulting (Voice, Data, GUI)
  - Test Automation Architecture Design (Voice, Data, GUI)
  - Test Script Development (Voice, Data, GUI)
  - Outsource test augmentation (Voice, Data, GUI)
  - Interop testing in our facility (Voice, Data, GUI)



# Software Automation

- Automation is a new mandate in today's fast past world of software development. Traditionally testing has long been seen as the bottleneck in getting software releases out the door and software test automation as the solution to removing that bottleneck. However software automation more often than not fails to live up to expectations and in fact seldom returns the investment put forth.



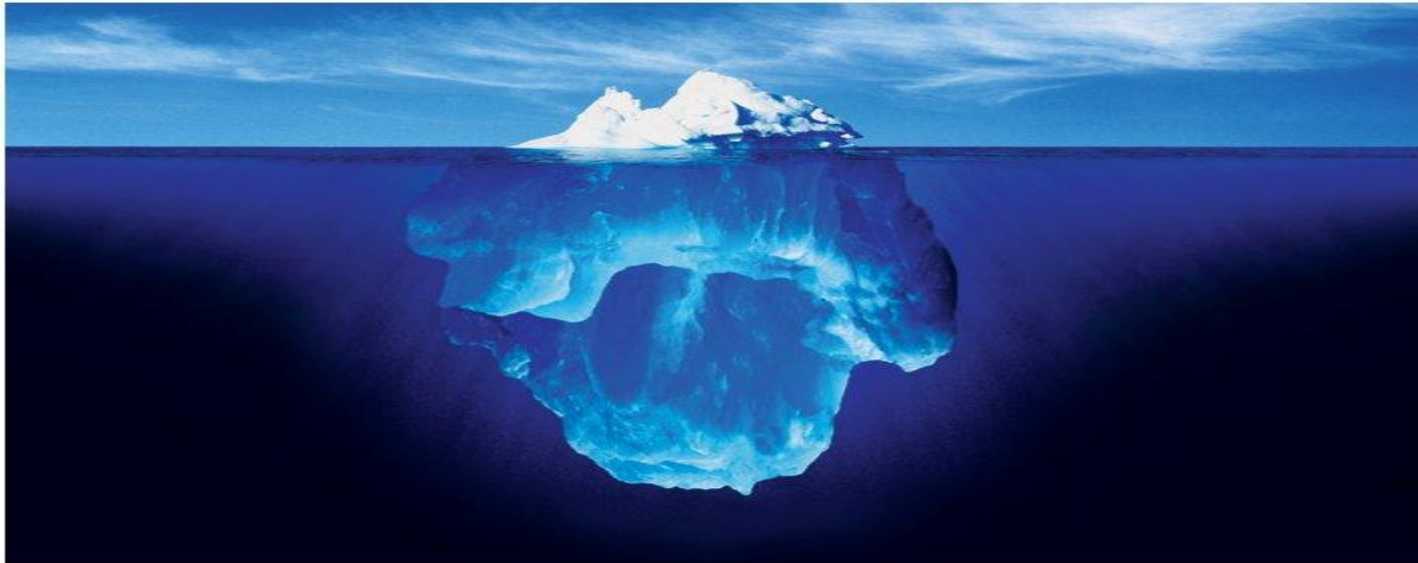
# Test Automation Failure

- The majority of software test automation projects fail. In fact most if not all automation projects fail to live up to peoples expectations and many projects are abandoned.



# Automation Barriers

- Automation (like the tip of an iceberg) seems like a small problem superficially however later (often too late) it is realized that the problem domain is much bigger (like the true size of an iceberg) than first realized.





# Automation Barriers

- The impression among many is that automation is simply a matter of purchasing the “right” tool and assigning a resource or two to the project. However, companies soon realize (to their surprise) that their testing needs are not met by the “tool” in fact they need significant (unforeseen) investment to save the project from failure.

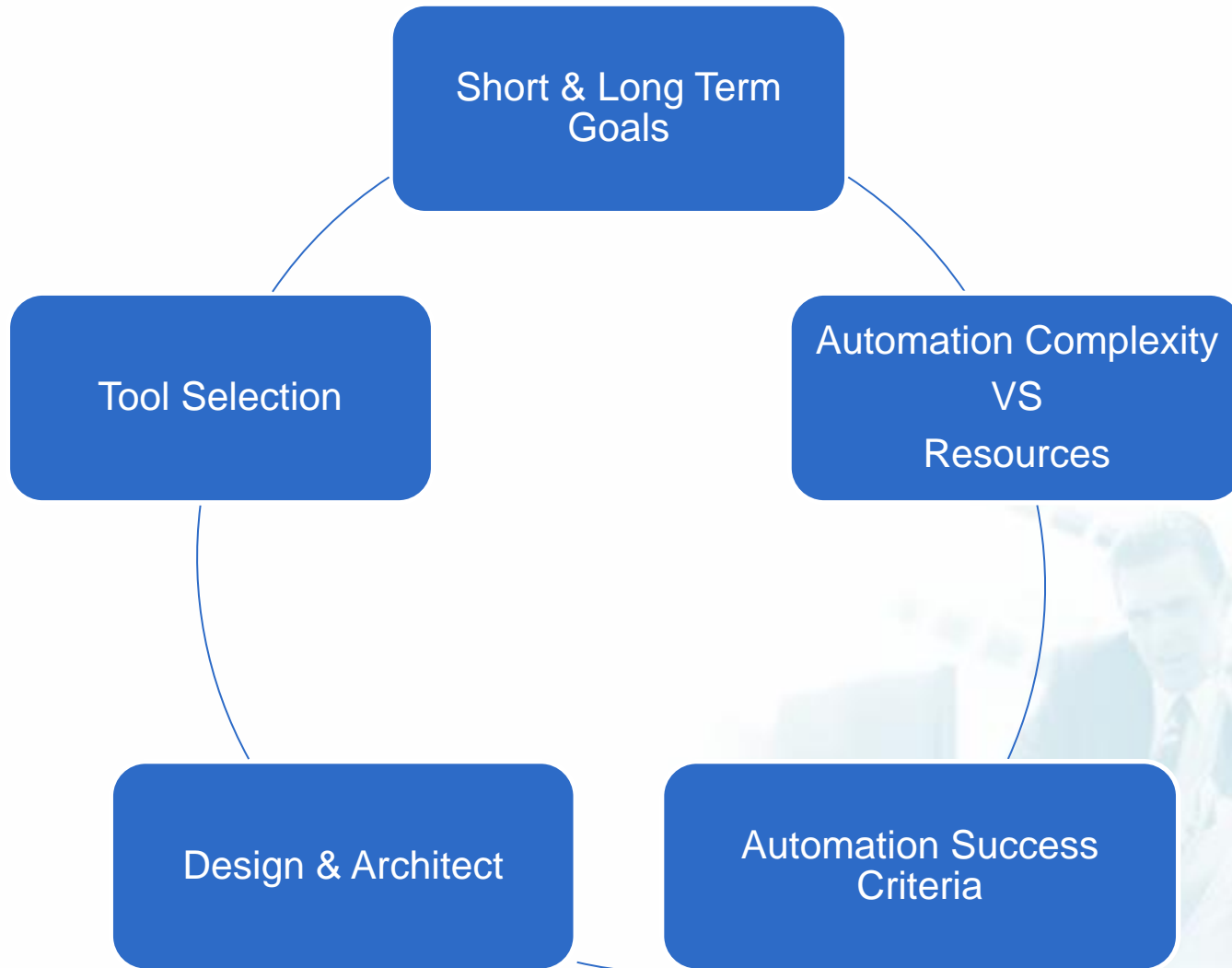


# Automation Barriers

- Tool companies are selling the ingredients to a solution. It is up to you to build it. They provide the rope to hang yourself with. Have you built an automation solution before?
- Most tools are built for the programmer not the Manual tester. However the manual tester has the most knowledge of the application under test.
- Automation is not seen as a important enough to be integrated into the other parts of the business process. E.g. programmers agree not to change application field names (which would break automation scripts).
- Automation does not mean your work is automatically done for you.



# Five Key Areas of Automation





# Successful Test Automation

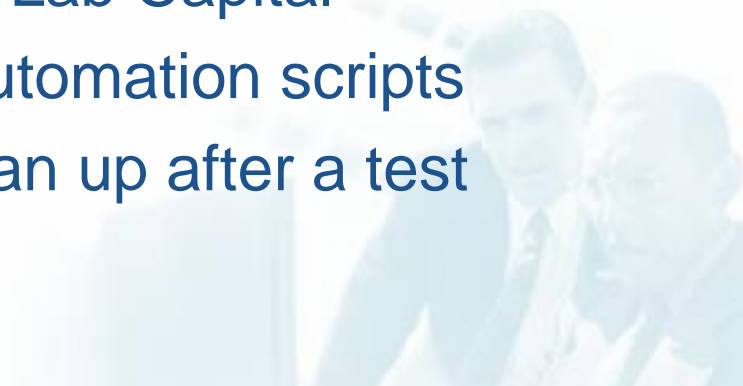
- Building Automation solutions is akin to building any (complex) Software system and requires the same effort to implement effectively. The same lessons learned in building systems should be applied to your automation project.
  - Inspections
  - High Level Designs
  - Requirements
  - Coding Standards
  - Versioning
  - Re-use
  - Efficiency





# Design & Architecture

- What are the high level goals of the automation system?:
  - Performance Testing
  - Unit Testing
  - Feature Testing
  - Regression Testing
- Will the Automation system need to interact with other software?
- Audience for the automation system? Manual Testers or Programmers?

- Develop a requirement list of the features you need in the automation architecture:
    - Logging Capabilities
    - API for Remote Access
    - Reporting
    - Resource management of Lab Capital
    - Ability to run concurrent automation scripts
    - Ability to automatically clean up after a test
    - Test Script versioning
    - Test Script Management
- 



# Design & Architecture

- The architecture needs a common framework
  - The framework lays the foundation and rules for everyone to live by.
  - Allows the automation system to be extensible
  - Standardized library and test scripts means easier maintenance and less training.
  - Support easy analysis of failed automated tests
  - Your Scripts and libraries used by your scripts should be modularized.
  - Capture Statistics and Metrics
  - Allow scripts to be re-used. For example a feature test could be re-used later as a performance test.

- Open-Source vs. Commercial vs. In-House, whichever path you choose ask yourself?
  - Does the tool support your short and long term requirements
  - Open-source or In-House tools are not free.
  - Is the automation tool leveraged by other groups in your company or will your group be the only group to use the tool?
  - Is the tool well supported and have an active user base? Can you Google for help? Do you need consultants? Books or tutorials?
  - Will tool issues and enhancements be supported
  - If in-house or open-source will the people developing and supporting the tool be focused on it in 2-4 years.
  - Easy to find new people skilled in the chosen tool? E.g. I can find people with QTP skills but perhaps not the open-source tool.



# Complexity VS. Resources

- Every choice in building an automation platform comes with consequences. Some might feel fine in the short term but be project killers long term. Keep in mind you have limited resources, time and capital.
  - Focus on the low hanging fruit at first. The automation team will be able to establish and enforce the correct processes for an effective automation program.
  - Some scenarios are too much work to automate. 1 test case run once a year but 40 hours to automate. ROI is 40 years.
  - Some scenarios introduce a much greater level of complexity to the automation system. This could hamper the stability and the scalability of the automation system.
  - Don't over-engineer your automation test cases just because...



# Complexity VS. Resources

- Full-Time Personnel must be qualified and fully committed to build and maintain an effective automation program
  - Carefully vet your team from a list of job requirements. Many automation projects are spearheaded by co-ops, interns, or unqualified/unmotivated staff with little or no experience with automation.
  - Keep the staff focused on automation. Don't divide between projects. Create a core staff of people to run and maintain the automation.
  - Don't automate more test scenarios if you don't have the staff to create, maintain, execute, and debug all your tests.



# Automation Success Criteria

- Create a list of criteria for determining success/failure and revisit these criteria at specific intervals over the life-span of the automation project
  - Low rate of False Positives
  - Reasonable rate of defects found
  - Reducing or Maintaining staff growth costs
  - Reducing Manual testing time

- Successful Automation involves:
  - Careful planning
  - Disciplined execution
  - Dedicated staff
  - Periodic re-evaluation of the automation system
  - Management support (with money, time, inclusion into the business process, and labor)



# Q & A





## Contact Information

Art Beall

[Art@martproservice.com](mailto:Art@martproservice.com)

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

Tel. 972-989-1195

Fax. 866-683-0678

