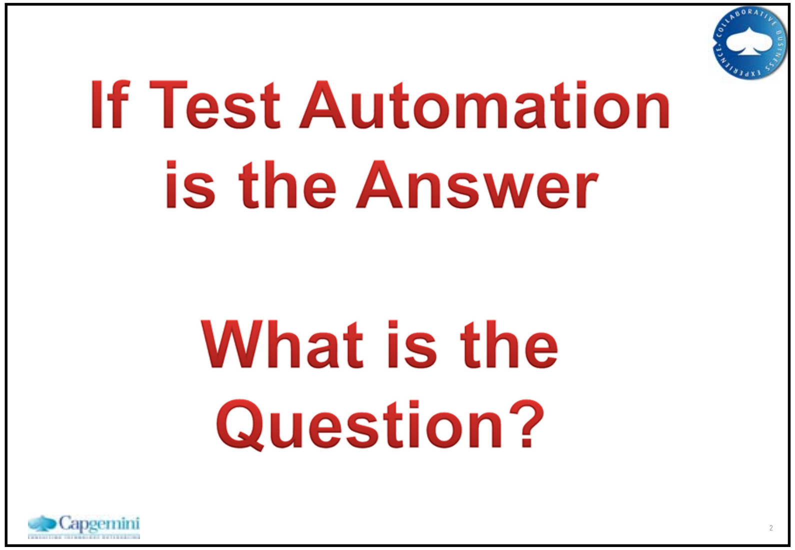
Test Automation: A Story Lost in Translation

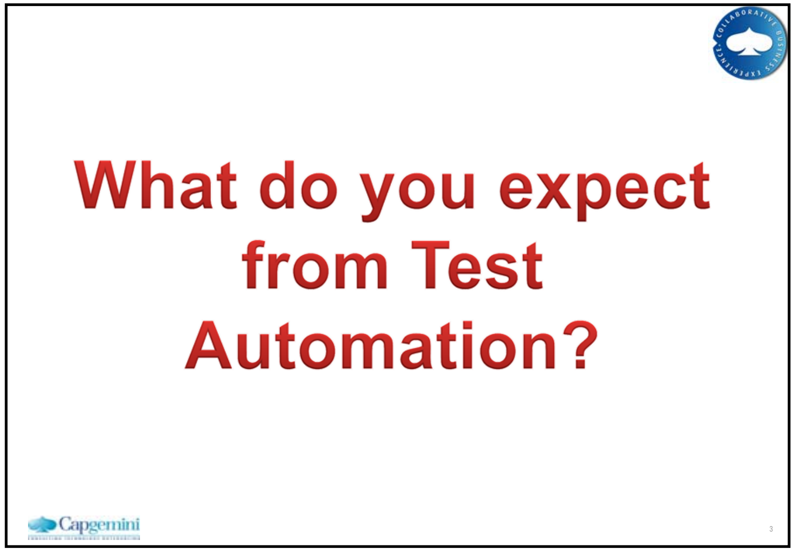
Author: Colin Cherry, *Director Testing Services*

TOGETHER. FREE YOUR ENERGIES



If Test Automation is the Answer

What is the Question?



What do you expect from Test Automation?

How Effective is your Test Automation?

Agenda

- My Background Story
- My Definition Story
- A Brief (Hi)Story of AST Tools
- The Story *for* AST
- The Story *against* AST
- The Evidence Story
- The Business Case Story
- The Requirements Story
- The AST Criteria Story
- The AST Framework Story
- The AST Success Story

My Background Story

- I wrote my first line of code in August 1973
- I wrote my first (manual) test a few days later
- I wrote my first (automated) test a few weeks later – when I needed to create & compare dates
- Developers have been creating automated tests for over 60 years… so why are we **still** discussing the merits of AST (Automated Software Testing)?

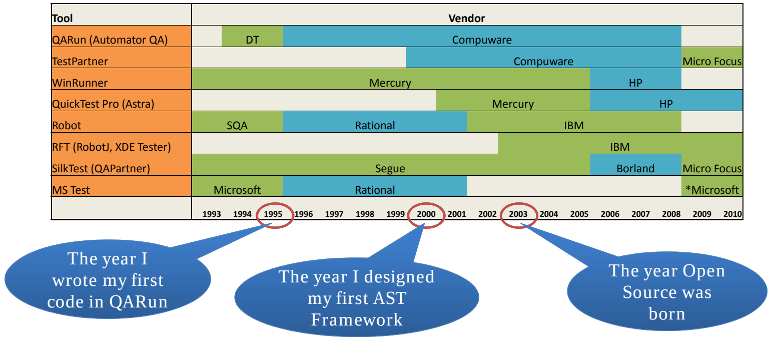
Because we've complicated the Story

My Definition Story

- AST enhances manual testing efforts by focusing on automating tests that manual testing finds hard to accomplish
- AST is software development & therefore has an SDLC
- AST does not replace the need for manual testing skills, test analysis & test techniques; manual testing is the basis for AST
- AST can't be clearly separated from manual testing; instead, both AST & manual testing are intertwined & complement each other

A Brief (Hi)Story of AST Tools

The problem started when the Vendors arrived on the scene



* Thanks to Sam Warwick for the timeline

The Story *for* AST

1. Improved build verification (Smoke Tests)
2. Improved Regression coverage
3. Multi-platform compatibility & configuration testing
4. Opportunity to replace mundane manual tasks
5. Improved ability to reproduce tests finding defects
6. Opportunity to perform after-hours (lights out) testing
7. Focused Regression testing
8. Improved Performance testing
9. Opportunity to perform memory-leak testing
10. Improved documentation & traceability
11. Distributed workload & concurrency testing

The Story *against* AST (1)

1. Approximately 37%* of AST initiatives fail due to lack of time; AST will not get a poor product to market quicker
   - If your software doesn't work, it doesn't matter how fast or cheaply you deliver it
2. Approximately 20%* of AST initiatives fail due to lack of experience
   - As in all code development, experience counts; use those with less experience on simpler/non-critical activities
3. Approximately 17%* of AST initiatives fail due to an insufficient budget
   - The Business Case needs to present a realistic budget & if the project (or BAU) can't afford to invest sufficient funds, AST should not be undertaken

* Source: Worldwide survey of over 700 organisations conducted by **IDT** in 2009 (Elfriede Dustin et al.)

The Story *against* AST (2)

4. Approximately 11%* of AST initiatives fail due to the incompatibility of the tool
   - A Proof of Concept & Pilot is essential before committing to a tool
5. AST is not effective on highly configurable software
   - Unless you create a manageable subset your automation suite will be unwieldy & costly to maintain
6. AST can change the priority of bugs that need fixing
   - A minor bug in an AST suite becomes a high priority if the suite is stuck
7. AST will not reduce the size of your Test team
   - Manual Testing is still be required & most Test teams are under resourced anyway

* Source: Worldwide survey of over 700 organisations conducted by **IDT** in 2009 (Elfriede Dustin et al.)

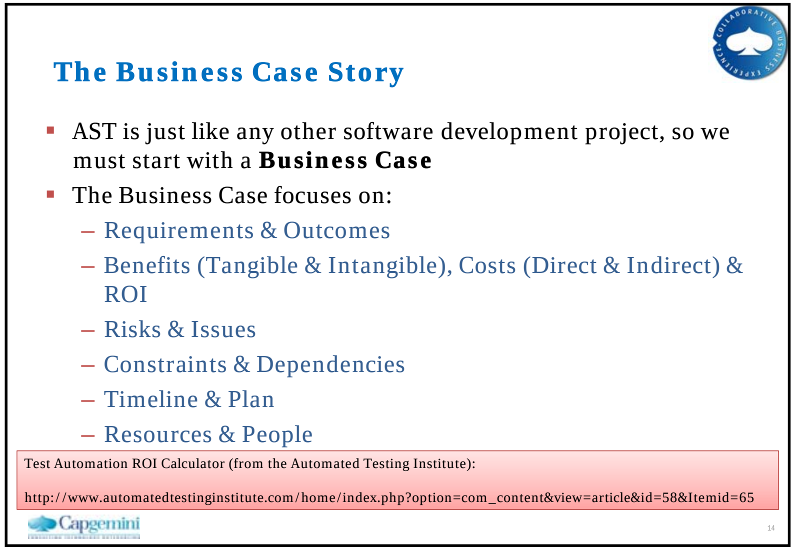
The Evidence Story

- The **TRUTH IS** more AST initiatives fail than succeed
- The **TRUTH IS** the majority of organizations that I have worked with have AST *Shelfware* - so, I have also failed *(at times)* to influence successful outcomes with AST
- The **TRUTH IS** *Capture/Replay* is responsible for more misconceptions about AST than any other *feature*

* Source: **Global Software Test Automation** (Hung Nguyen, Michael Hackett, Brent Whitlock)

The Road to a Better Future Story

TOGETHER. FREE YOUR ENERGIES

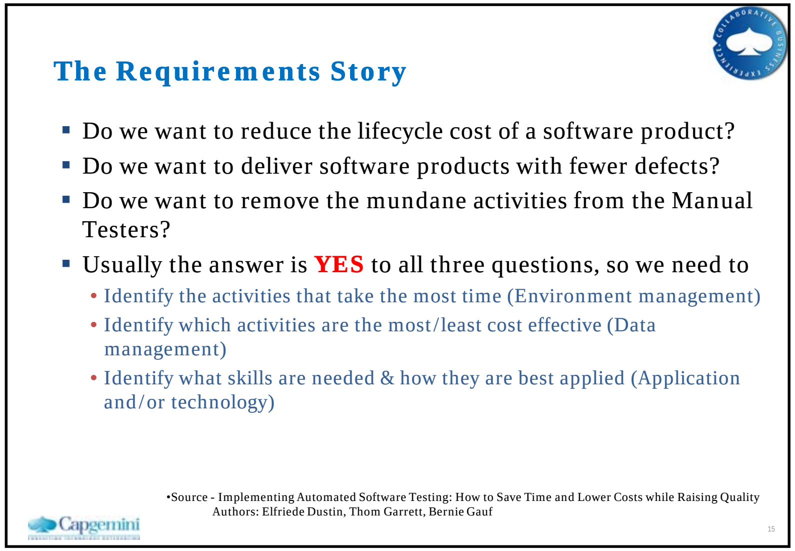


The Business Case Story

- AST is just like any other software development project, so we must start with a **Business Case**
- The Business Case focuses on:
  - Requirements & Outcomes
  - Benefits (Tangible & Intangible), Costs (Direct & Indirect) & ROI
  - Risks & Issues
  - Constraints & Dependencies
  - Timeline & Plan
  - Resources & People

Test Automation ROI Calculator (from the Automated Testing Institute):

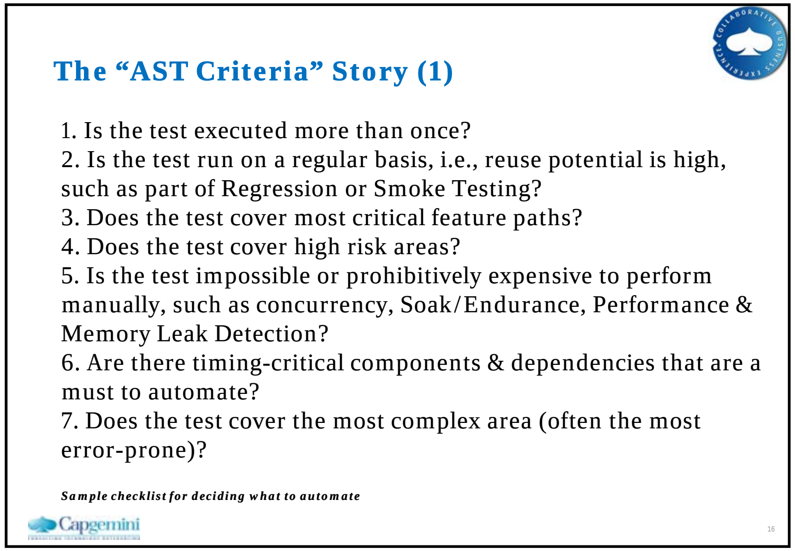
http://www.automatedtestinginstitute.com/home/index.php?option=com_content&view=article&id=58&Itemid=65



The Requirements Story

- Do we want to reduce the lifecycle cost of a software product?
- Do we want to deliver software products with fewer defects?
- Do we want to remove the mundane activities from the Manual Testers?
- Usually the answer is **YES** to all three questions, so we need to
  - Identify the activities that take the most time (Environment management)
  - Identify which activities are the most/least cost effective (Data management)
  - Identify what skills are needed & how they are best applied (Application and/or technology)

•Source - Implementing Automated Software Testing: How to Save Time and Lower Costs while Raising Quality
Authors: Elfriede Dustin, Thom Garrett, Bernie Gauf



The “AST Criteria” Story (1)

1. Is the test executed more than once?
2. Is the test run on a regular basis, i.e., reuse potential is high, such as part of Regression or Smoke Testing?
3. Does the test cover most critical feature paths?
4. Does the test cover high risk areas?
5. Is the test impossible or prohibitively expensive to perform manually, such as concurrency, Soak/Endurance, Performance & Memory Leak Detection?
6. Are there timing-critical components & dependencies that are a must to automate?
7. Does the test cover the most complex area (often the most error-prone)?

Sample checklist for deciding what to automate

The "AST Criteria" Story (2)

8. Does the test require many data combinations using the same test steps (i.e. multiple data inputs for the same feature)?
9. Are the expected results constant, i.e. do not change or vary with each test? Even if the results vary, is there a percentage tolerance that could be measured as expected results?
10. Is the test outcome analysis overly time-consuming, such as expected results analysis of hundreds of outputs?
11. Does the test need to be verified on multiple software & hardware configurations?
12. Does the test automation ROI look promising & meet any organizational ROI criteria?

Sample checklist for deciding what to automate

The AST Framework Story

- Effective AST Frameworks should
  - Support the integration of multiple commercial testing tools – my first framework (in 2000) was developed to incorporate all the tools available at the time
  - Support the integration of commercial, open-source & in-house tools
  - Allow for new tools to replace out of date ones
  - Be adaptable for new applications
  - Support the entire SDLC (from UT to E2E); however, it is the re-use factor that is most important, not that every phase uses AST
  - Support distributed testing across multiple computers & environments

The AST Success Story

- Keys to a successful AST implementation
  - Be clear about your requirements
  - Develop an AST Strategy
  - Test the Framework
  - Create a benchmark, set targets & monitor progress
  - Implement & continuously review/improve processes & procedures
  - Identify the required skills & get the right people onto the team
- Don't underestimate the COST

Total Cost of AST =

Test Tool Cost + Script Creation Costs + Script Maintenance Costs

Colin Cherry
Director
Testing Services

Capgemini Australia Pty Ltd
Level 2, 477 Collins Street
Melbourne VIC 3000 Australia

Tel: +61 3 9613 3138
Mob: +61 412 214 240

colin.cherry@capgemini.com