Evolution of Test Automation
State Driven Testing

Jan De Coster

November, 2011

Why test automation?

- Use computers to replace expensive manual testing
- Can’t do it all manually
  - Coverage of functionality
  - Coverage of platforms
  - Coverage of complex data inputs
- Catch problems earlier at less cost

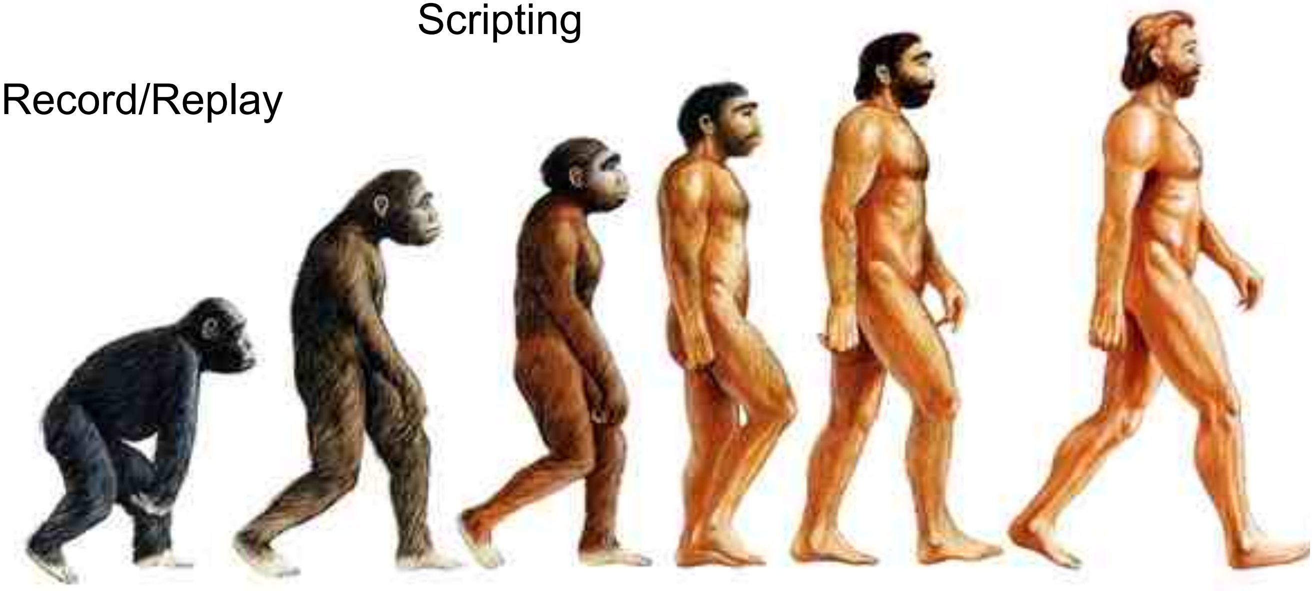
The trick is whether automation can provide an ROI

Evolution of Test Automation

Keyword Driven Testing

Scripting

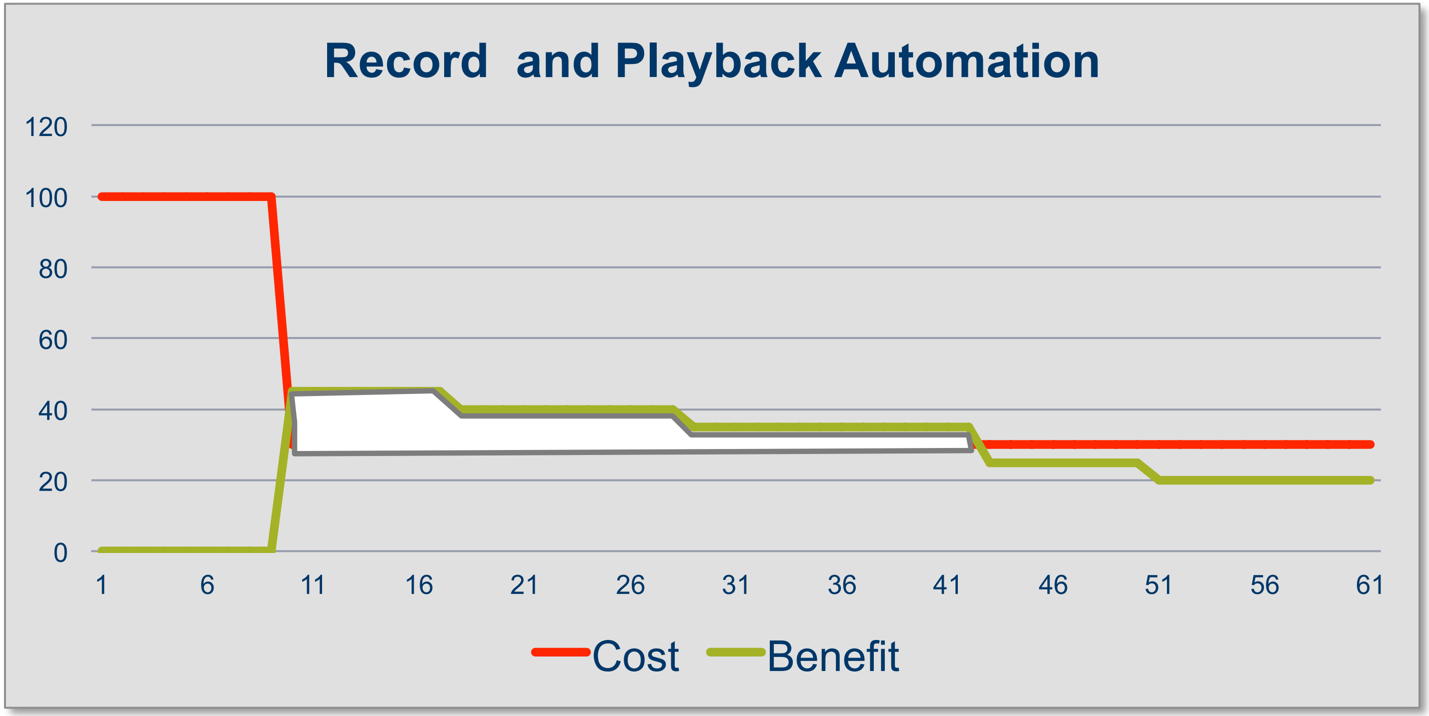
Record/Replay



Early Approach: Record and Playback

- Create long recordings of user actions and make tests from them
- Problems
  - Expensive to build scripts with nothing to reuse
  - Application changes tend to break lots of scripts
  - If not caught maintenance issues cause reliability problems

Cost Benefit - Record and Playback



- Fragile and time consuming
- Very high maintenance
- Tough to achieve benefits

What do I need? - An Automation Framework!

How do I use this automation language?

How do I incorporate test data into my scripts?

How does the technology in my application work?

How can I be more efficient and get more reusable code?

What business logic and use cases should I be testing?

How am I going to maintain this over time?

Image: sippakorn / FreeDigitalPhotos.net

What does an automation framework provide?

- Structured way to add new automation
- Capability to drive data into test cases
- A roadmap and process to follow
- Provides a way to reuse pieces of automation
- Isolates change due to application change
- Brings down the costs of test automation

Keyword Testing Frameworks

Action	Object Type	ObjectID
ClickSubmit	Button	Submit
TypeName	TextField	UserName
VerifyName	TextField	UserName
SelectRole	DropDownList	Roles
VerifyTitle	Window	Browser

Keyword Approach

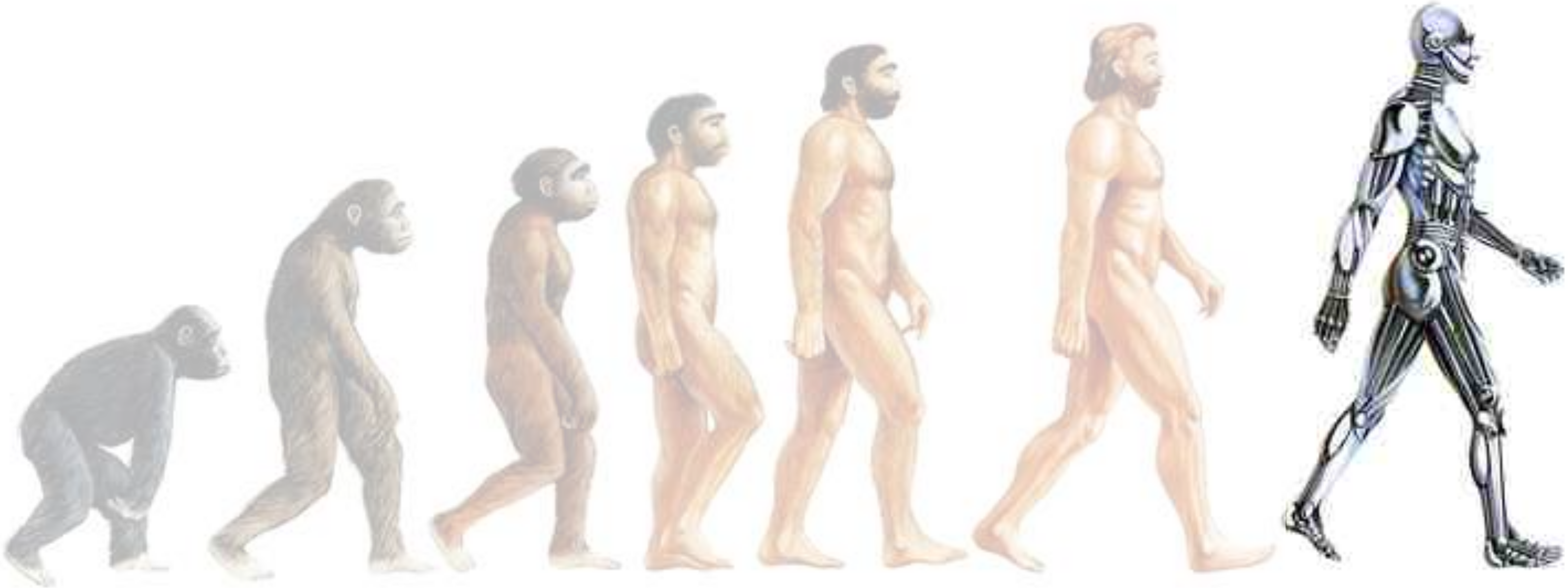
- Maps an Application Under Test to a set of action based keywords
- Business savvy people assemble test cases using keywords
- Develop test cases without programming knowledge

Shortcoming: Keyword-driven Testing

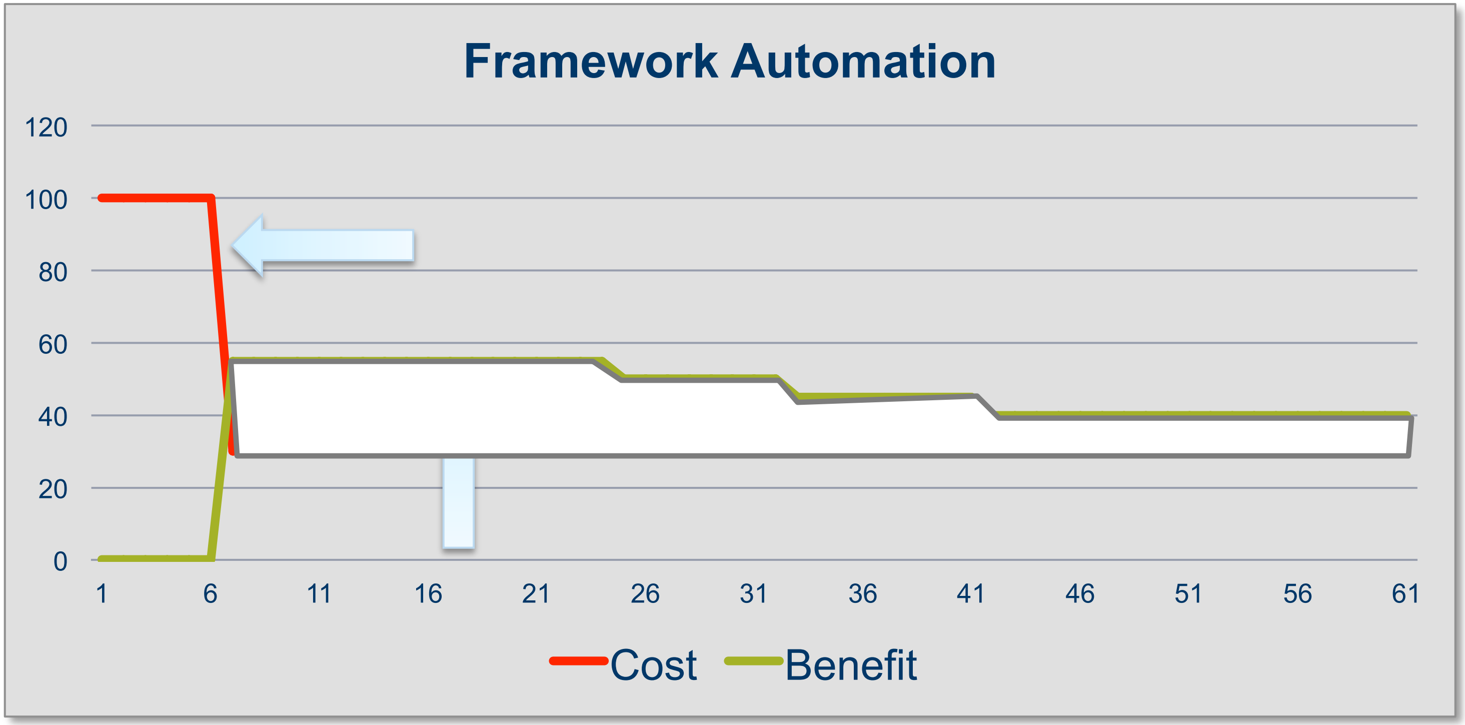
> *"While keyword-driven sounds wonderful, it is not a magical methodology that will solve all automation problems and cure world hunger. I worked on a keyword-driven project while I was an employee of a big corporation. We had an elaborate in-house tool, that could compose the keywords into larger blocks of actions, which were also reusable in tests. The project was a failure. The library of keywords became so huge that no one could figure out which keyword should be used in which context."*

Source: http://testautomationblog.com/2010/05/16/keyword-driven-automated-testing/

Next Generation of Test Automation

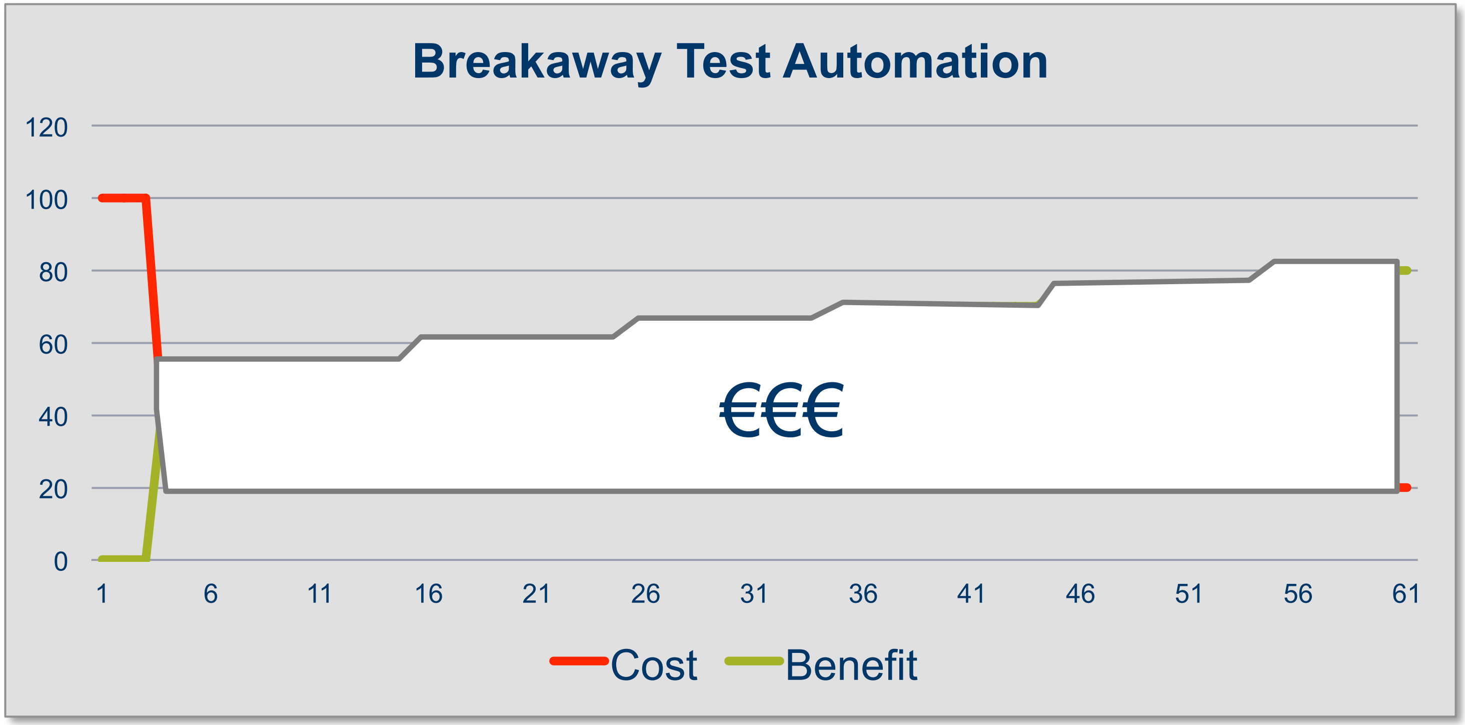


Cost benefit – Framework Automation



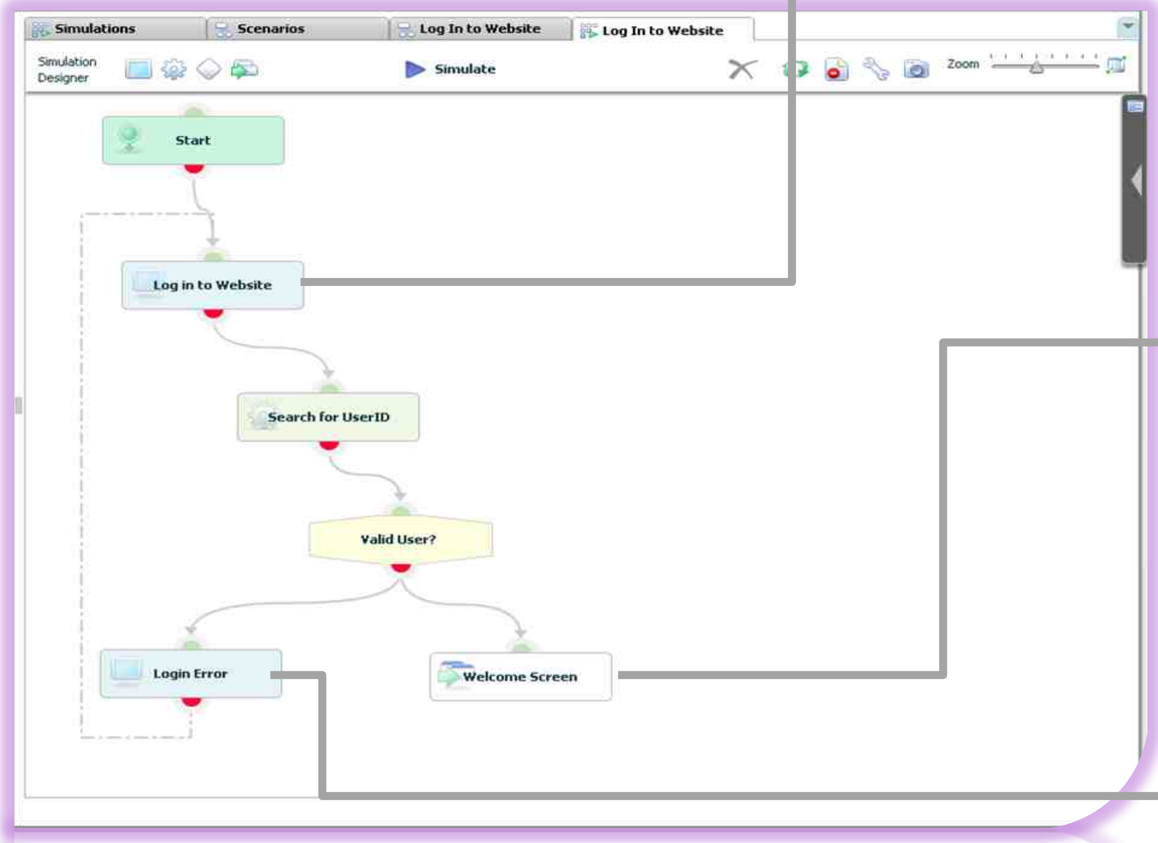
- Maintenance and extension doesn't scale
    - Application complexity
    - People and process complexity - hard for BA's to participate
- ROI is there but limited

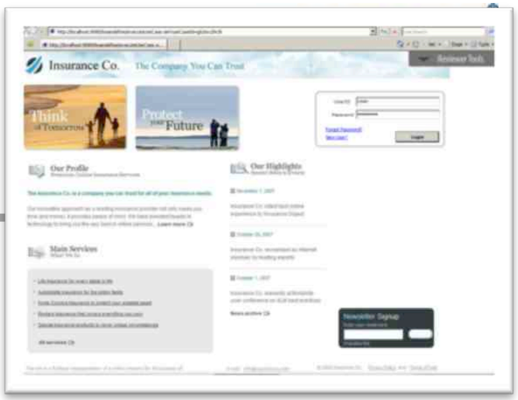
What’s needed? – A breakaway benefit!

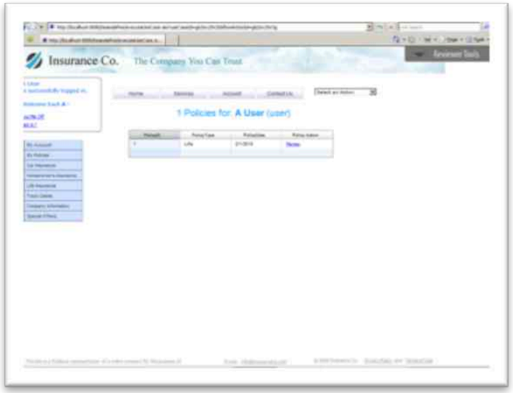


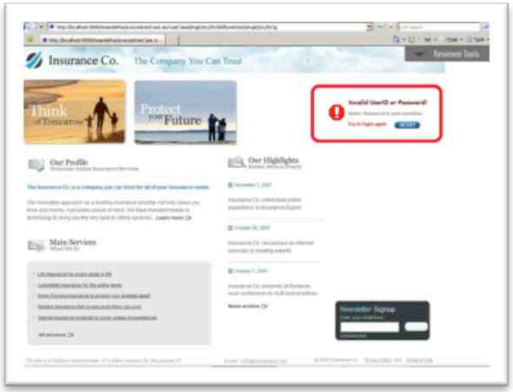
- All team members can participate
- Handles application complexity well
- Benefit continues to grow

Software state transitions



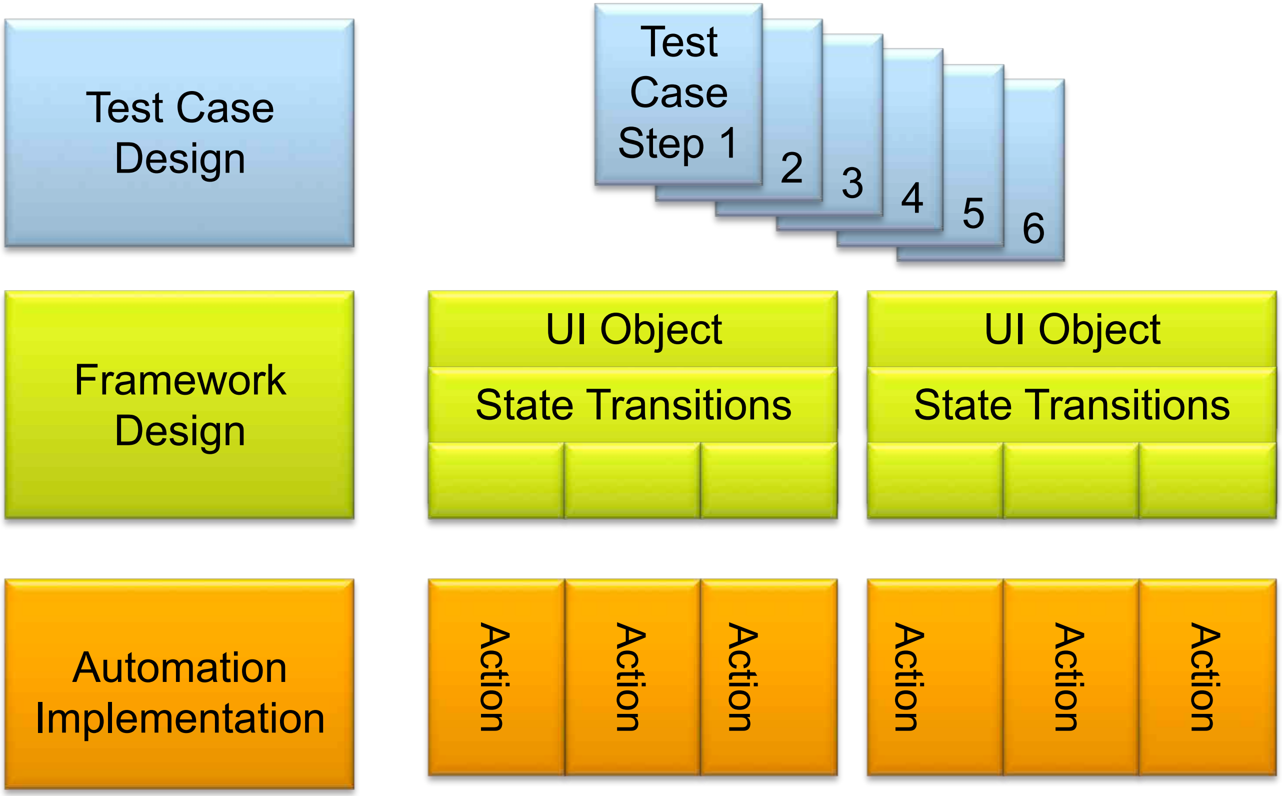




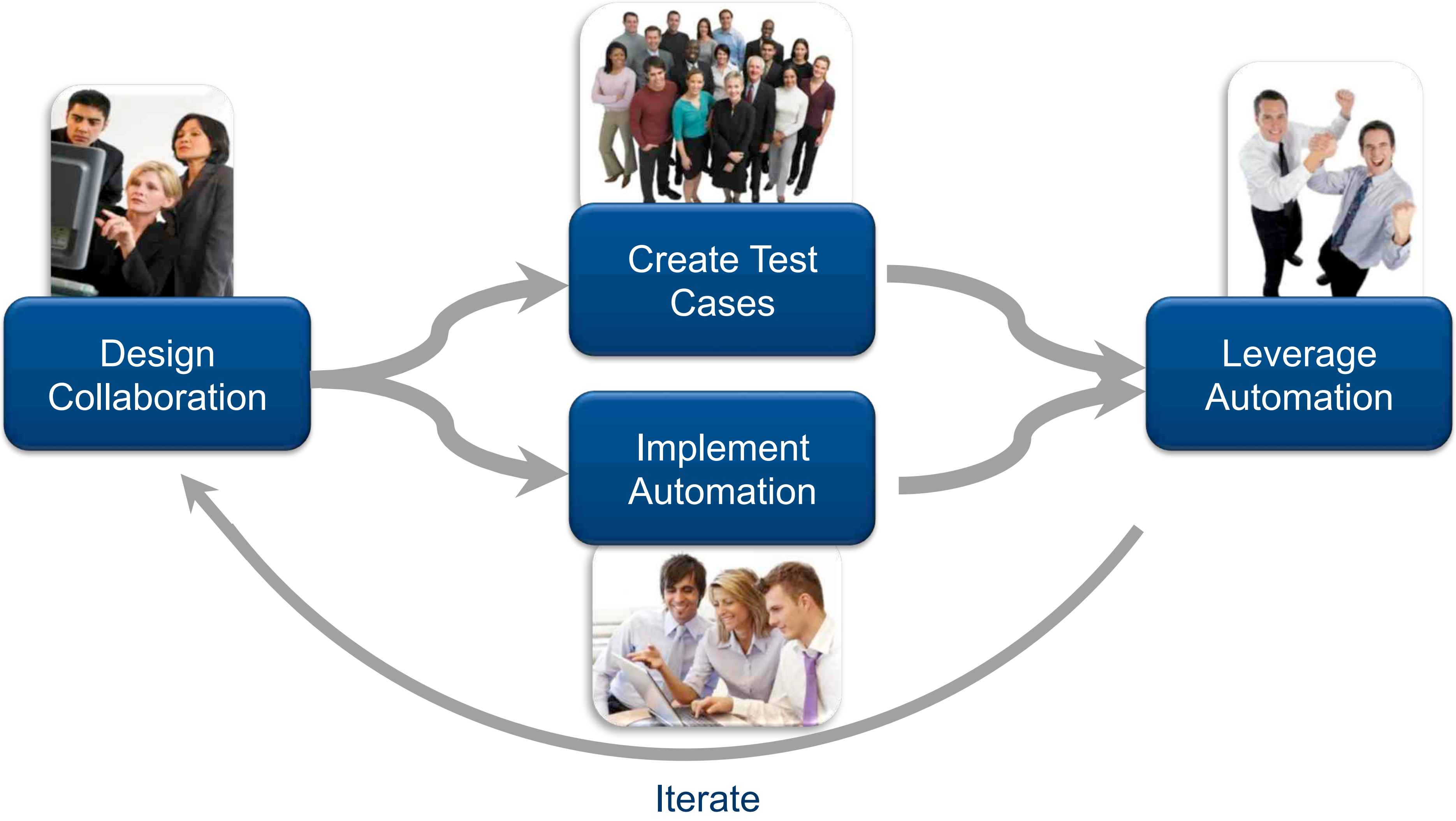


Next Generation Test Automation
State Driven Testing

- Organized and scalable
- Open for different types of test automation
  - Mobile
  - SOA
  - App specific



State Driven Test Automation – Iterative Process



Design Collaboration

Create Test Cases

Implement Automation

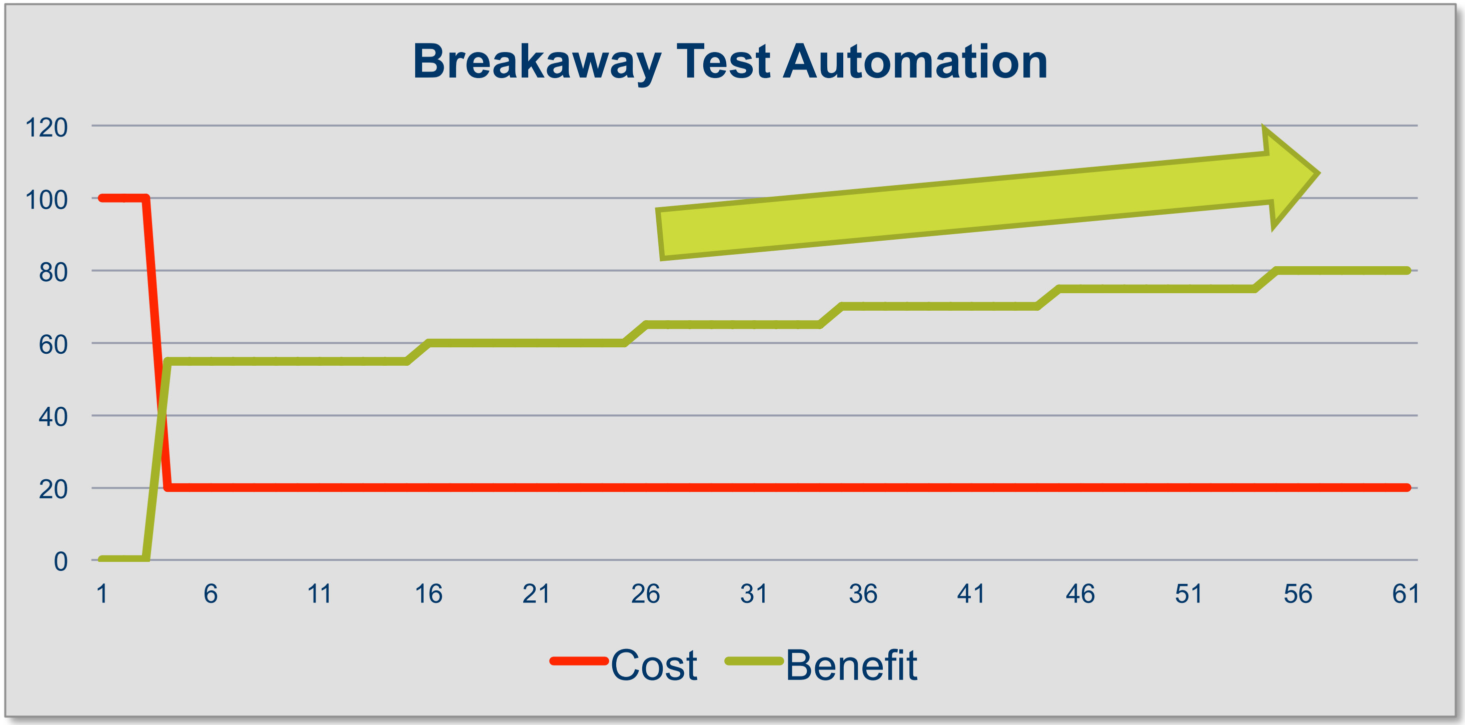
Leverage Automation

Iterate

State Driven Testing value

- Minimizes technical automation effort
  - Automates framework creation
- Maximizes productivity of business experts
  - Accelerates the creation of test cases
  - Produces clear easily readable test cases
- Minimizes maintenance costs
  - By reducing duplicate test automation
- Enables agile approaches
  - By decoupling test design and test automation

State Driven Testing – Breakaway benefit!



MAXIMUS CANADA
A MAXIMUS Company

- Situation
  - Needed to automate an application quickly

> Within two weeks of beginning Automated Test Script development, the automation team developed over 2200 transaction based tests (this equates to approximately 150 tests per tester per day – manual test development is approximately 15 test cases per tester per day).
>
> Janine Roy – MAXIMUS Test Team Lead

  - Test case development underway
  - MAXIMUS very satisfied

MICRO FOCUS®

Thank You!

I CAN'T DEVELOP AN AUTOMATED TESTING SYSTEM BY THE ARBITRARY DEADLINE YOU SET.
www.dilbert.com scottadams@aol.com
TRY WORKING SMARTER, NOT HARDER, WITH A SENSE OF URGENCY, AND A BIAS FOR ACTION.
7-30-07 © 2007 Scott Adams, Inc./Dist. by UFS, Inc.
OR MAYBE *YOU* COULD DO SOMETHING DIFFERENTLY.
I'M NOT THE ONE WHO CAN'T GET HIS WORK DONE.