Automation for Managers

Robby Green

QUEST 2009

Agenda

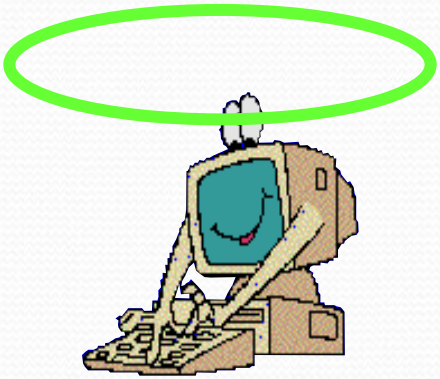
- What is Automation?
- Can everything be automated?
- Should I or shouldn't I go for automation?
- Automation Suitability
- How to make this choice?
- Information that should be taken into account
- What are the risks in automation?
- How should automation be presented to senior management?
- The Automation Process
- Advanced Automation
- The soft side - how to get the cooperation of the existing test team and overcome their fear of becoming redundant?
- Automation projects
- Summary
- Appendix:
  - tools selection and ROI calculation
  - More examples of automation projects

What is Automation?

"Test Automation is the act of converting test cases to machine executable code"

- The output of a test automation project is a (or set of) test suite/s which will be used by testers to verify the application time and again.
- Test Automation is perceived as an in-process efficiency improvement program, which will improve time to market advantage for product development organization.

Should I Automate?



No Testing

Manual Testing

- Time consuming
- Low reliability
- Human resources
- Inconsistent

Automated Testing

- Speed
- Coverage
- Repeatability
- Reliability
- Reusability
- Process Orientation
- Programming capabilities
- Employee Satisfaction

Myths about Automation

- capture / replay = automation
- Immediate pay back
- Zero ramp up time
- One tool fits perfectly
- No maintenance costs

Can everything be automated?

YES !

But…

What will it cost me?

- Fixed Cost
  - Automation feasibility analysis cost
  - Tool selection and Acquisition cost
  - Hiring skilled resources OR training existing team members
  - Time in learning the application/business processes
  - Pilot project identification effort and Proof of Concept
  - First time automation of the identified parts of application/product
  - Test suite Documentation effort
- Variable cost
  - Test script and documentation maintenance cost
  - Automated Test infrastructure maintenance cost
  - Execution cost

But…

Benefits of Automated Testing

Tangible:

- Speed and Accuracy
- Accessibility
- Accumulation
- Manageability
- Repeatability
- Availability

Intangible :

- Formal Process
- Retention of customers
- Greater job satisfaction for testers

How do I make the choice

- A scorecard should be maintained to identify the right candidate of automation.
- Automation should be given priority based on the business priority & the scorecard of each test case.

High Scoring Test Cases

- ☑ Tests that need to be run for every build of the application *(sanity check, regression test)*
- ☑ Tests that use multiple data values for the same actions *(data driven tests)*
- ☑ Complex and time consuming tests
- ☑ Tests requiring a great deal of precision
- ☑ Tests involving many simple, repetitive tests
- ☑ Testing needed on multiple OS/Browser combination

Low Scoring Test Cases

- ☒ Usability testing - *"How easy is the application to use?"*
- ☒ One-time testing
- ☒ "ASAP" testing - *"We need to test NOW!"*
- ☒ Ad hoc/random testing - *based on intuition and knowledge of application*
- ☒ Interface testing
- ☒ Batch program testing
- ☒ Back-end testing

More repetitive execution?
Better candidate for automation.

You should automate if you want to…

- Run existing tests on a new version(s) of a program/application
  - A lot of programs are frequently modified in the same environment so re-use test scripts across releases and configurations
- Run more tests more often
- Perform tests that would be difficult or impossible to do manually
  - Bugs that can be uncovered only with long runs
  - Checking multiple things at the same time
- Have better use of resources
  - Testers can do better jobs, machines can be run all the time
- Consistency, Reliability and Repeatability of tests
  - The same tests will be repeated exactly in the same manner, every time when the automated scripts are executed thus eliminating manual testing errors
- The testing elapsed time can be shortened, therefore leading to a huge saving in terms of time and money. Generally, the ROI begins to appear in the third iteration of automated testing

Usual Automation Suitability

Test Stage	Type of Testing	Amenity to Automation	Factors Influencing
Regression Testing	Black Box	High	■Focus is to test the impact on the stability of the original application ■Tests are repetitive (Same functionality need to be tested for every release)
Model Office Testing	Black Box	High	■Application will be highly stable ■Final level of acceptance tests and therefore less number of outstanding defects
Performance Testing	Black Box	High	■Application will be fairly stable ■Tests are repetitive in nature (Same tests need to repeated for various parameters)
User Acceptance Testing	Black Box	Medium to High	■Application will be fairly stable ■Depends on the # of outstanding defects from System Testing ■Number of test cycles planned (At least 2 cycles should be planned)
System Testing	Black Box	Medium	■No repeatability / No stability in terms of functionality ■High cost for automation
Integration Testing	White Box	Low	■System level integration testing ■White box and not a functional testing ■No repeatability and no stability in terms of both technical and functional aspects
Unit Testing	White Box	Low	■Component level white box testing ■No repeatability and No Stability as system is still in construction phase

And you need to make sure that you have

- An adequate Test Environment that accurately replicates the production environment

  – Can be small scale, but it must consist of the same types of hardware, programs and data
- Test Environments database that can be restored to a known baseline
- Dedicated PC's to run automated scripts
- Detailed and accurate manual test cases that can be converted to automated format
- Appropriate tools that can be effectively used

Risks

- Too Long preparation
- Not enough Cycles
- Tools Costs
- Employee dissatisfaction
- Management support
- Quality & Audits (validated dev??)

How to present to Managements

ROI !

- Effort = ROI
- Time to market = ROI

ROI Presentation I

Cost for Manual Execution

Cost for Automated and Manual Execution

Releases	Approx. Execution timeframe	Manual Testing
Campaign-1		82660
Campaign-2		82660
Campaign-3		82660
Campaign-4		82660
Campaign-5		82660
Total Cost (all releases)		413298

Releases	Cost in EURO					
	Automation (One time)	Automation (Maint.)	Auto-80% (Execution)	Man-20% (Execution)	Total Exec. (Automation)	Difference (Automation -Manual)
Campaign-1	101830	0	18677	15666	136172	53513
Campaign-2	0	5091	18677	15666	39434	-43225
Campaign-3	0	5091	18677	15666	39434	-43225
Campaign-4	0	5091	18677	15666	39434	-43225
Campaign-5	0	5091	18677	15666	39434	-43225
Total Cost (all releases)					293909	-119389

***Enhancement costs for Automated Execution only include the Automation build effort (Reuse factored with existing automation framework) and test execution effort, test planning effort not considered*

ROI presentation - II

Releases	Approx. Execution Timeframe	Manual Testing (GBP)	Automation Testing (GBP)
Campaign-1		82660	136172
Campaign-2		82660	39434
Campaign-3		82660	39434
Campaign-4		82660	39434
Campaign-5		82660	39434
TOTAL		413298	293909

Overall Savings (all releases till Sep'10)	-119389

Cost of automation testing for above releases is 293909* EURO, which is 119,389 EURO cheaper compared to manual testing providing 29% savings.

**Does not include QTP Tool cost*

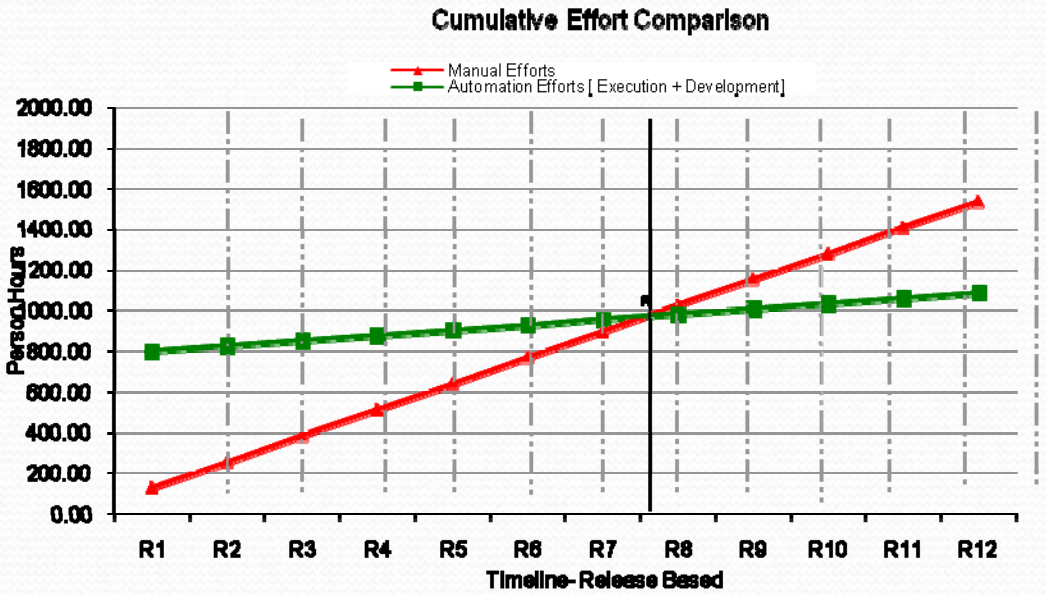
High level Return On Investment Analysis

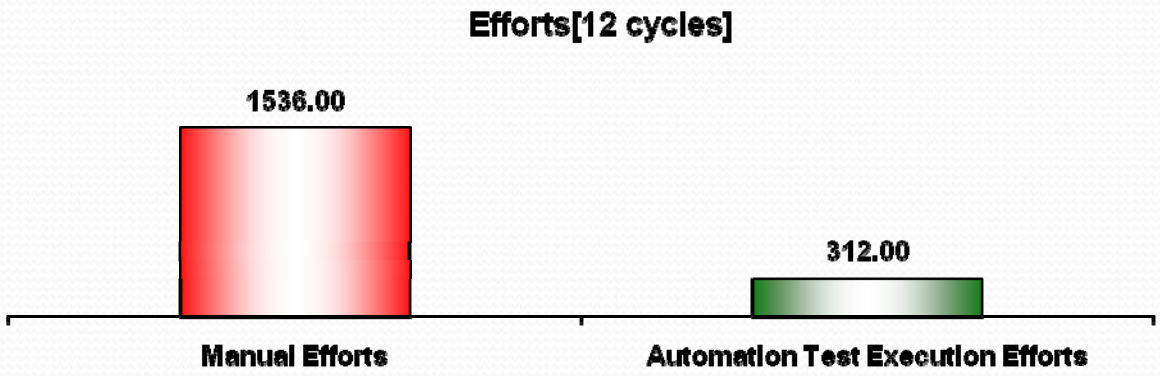
Inputs for calculation

- 41 % of available regression suite (160 test cases) considered for automation.
- Approximately 12 cycles of the automated suite per year may be planned.
- ROI is calculated assuming only automated test cases will be executed as regression in next 12 cycles.
- Approximately 740 hours considered for Automation development effort.

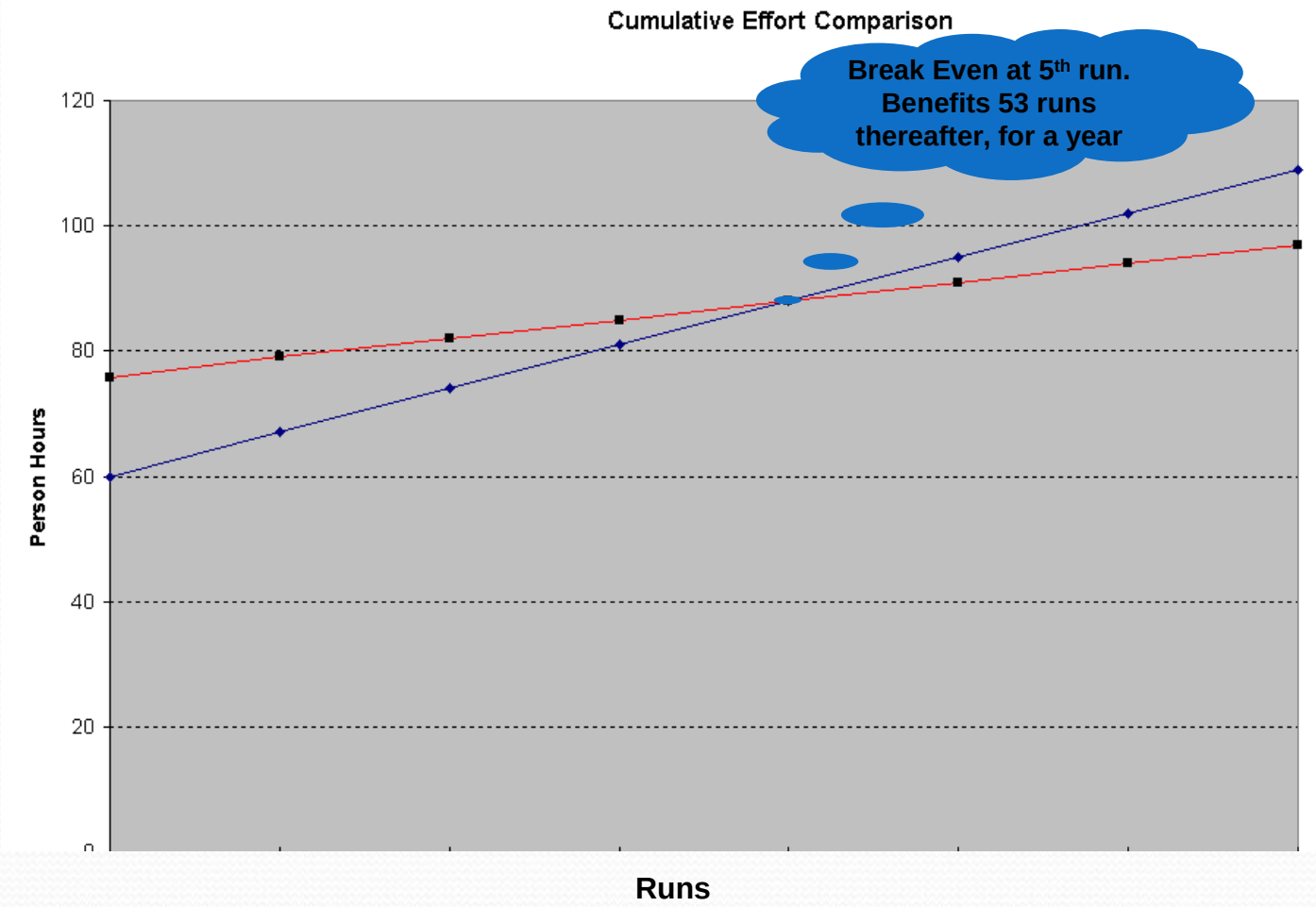
Benefits

- Approximately 75% effort saving in Test execution
- Approximately 70% savings in Test execution time
- Flexibility to plan more regression cycles



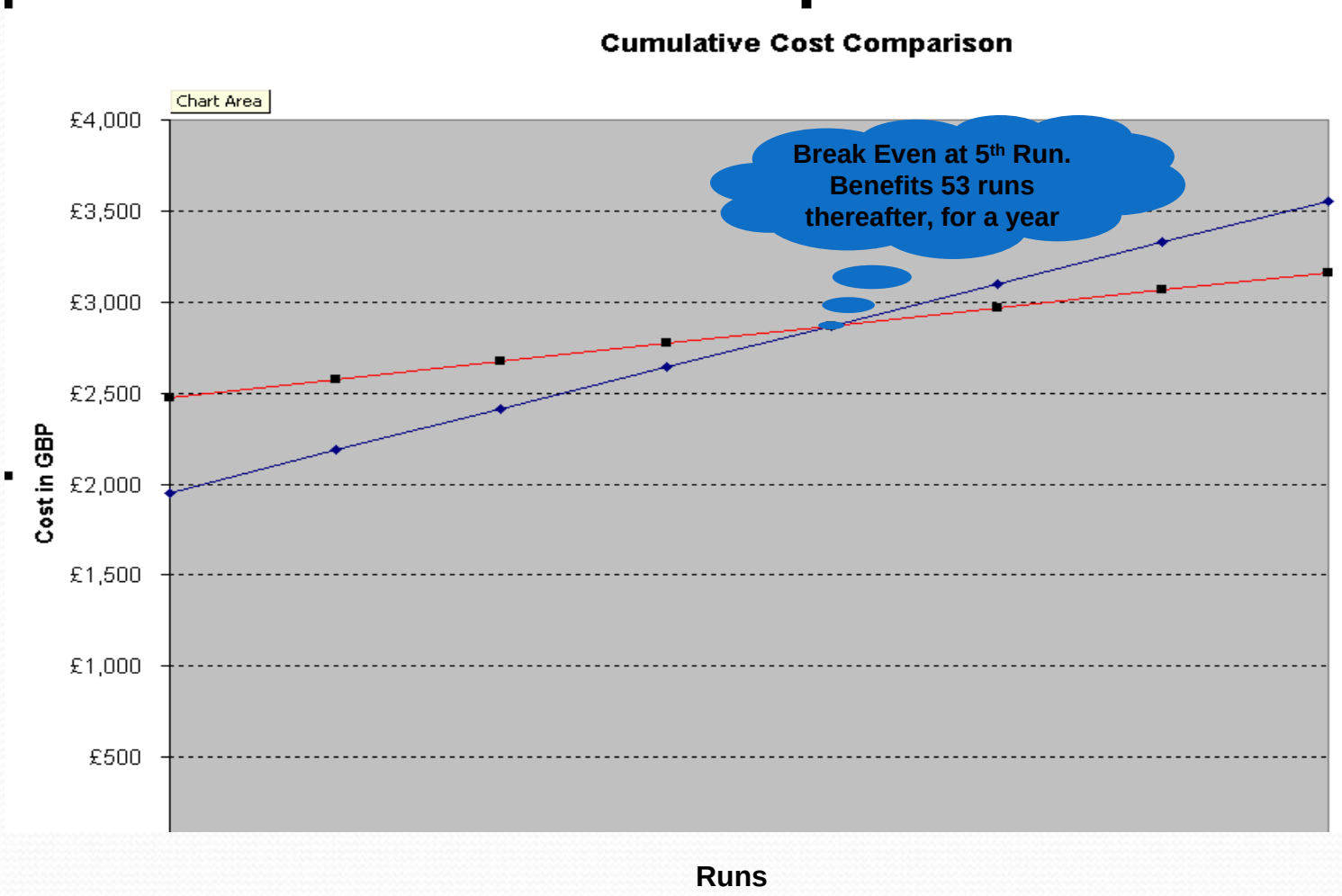


ROI – Effort Comparison



Note: Efforts shown in above graph are cumulative

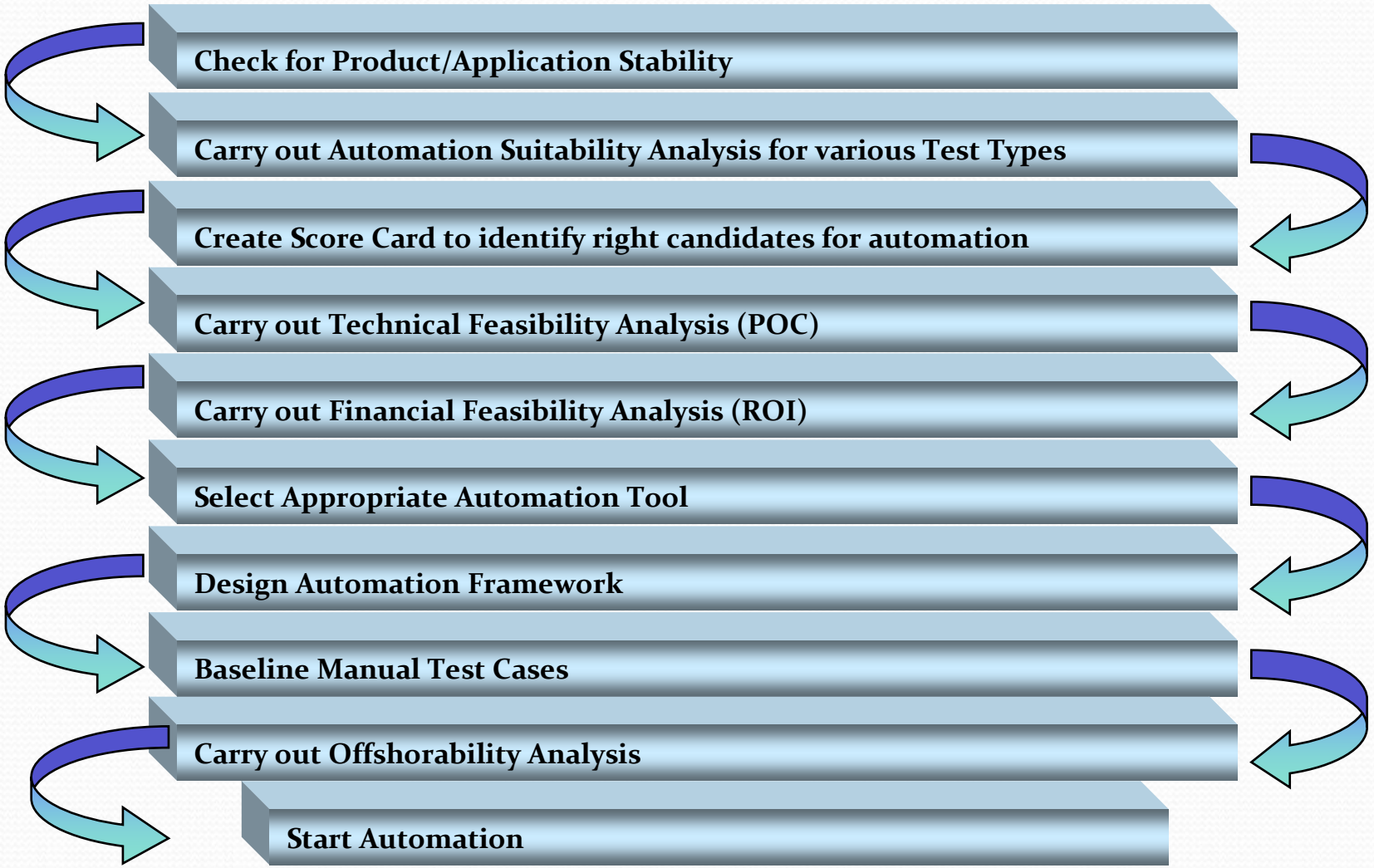
ROI – Cost Comparison



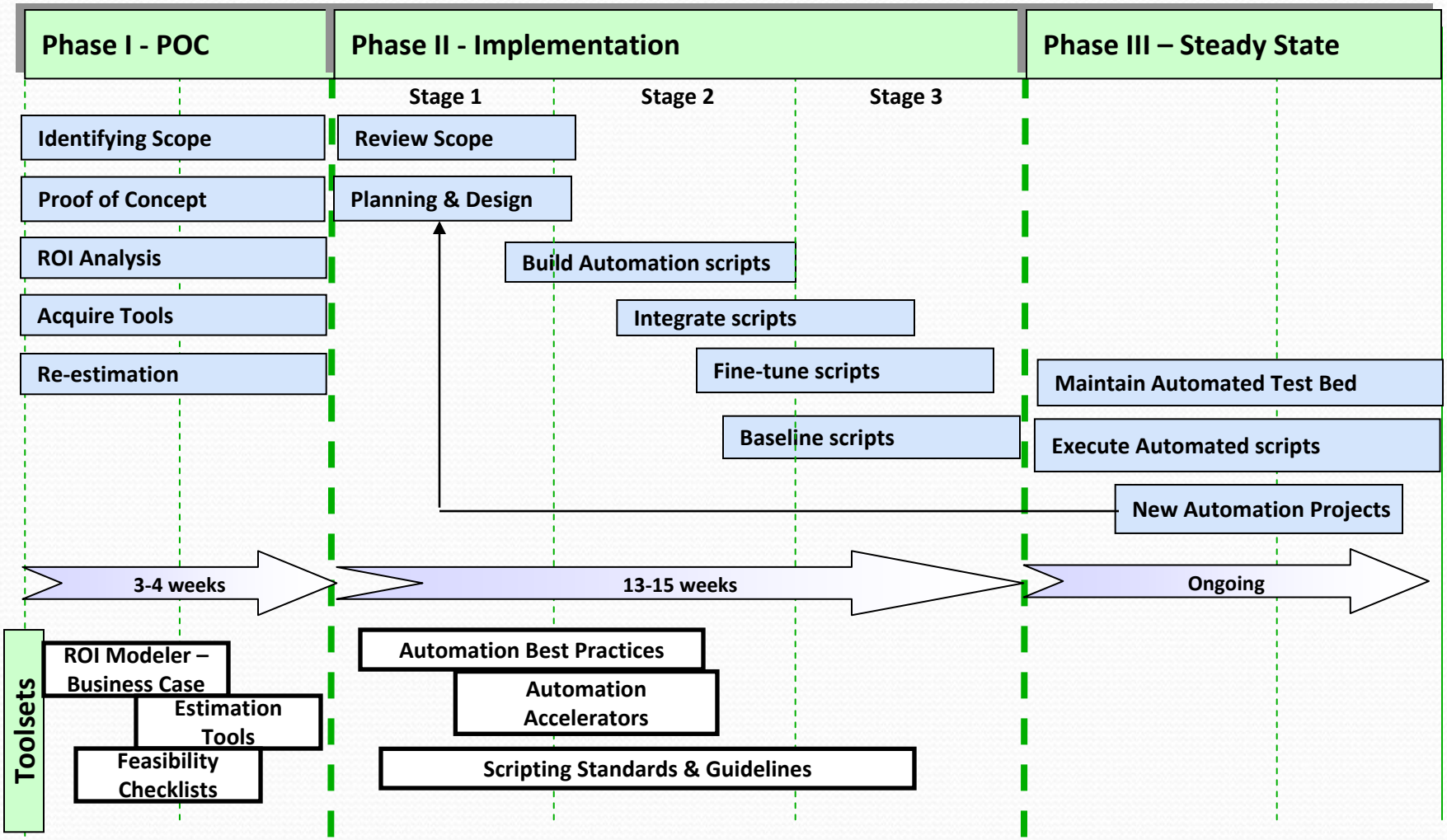
Note: Costs shown in above graph are cumulative

So How do we start

Automation Methodology



Firms have employed following Implementation Framework to ensure returns from automation are defined & agreed and risks minimised



The Implementation Phase of Automation includes three well defined stages

The Implementation phase is further structured in following three stages

Stage 1 – Test Automation planning and design

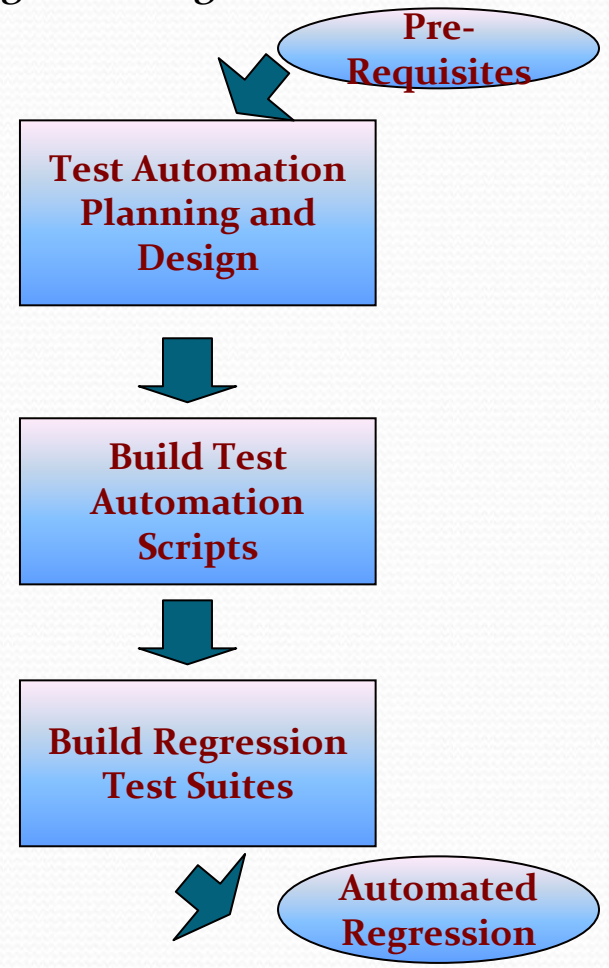
- Study existing system and application
- Scoping of work
- Project Plan and Strategy
- Detailed Framework Design

Stage 2 – Build Test Automation scripts

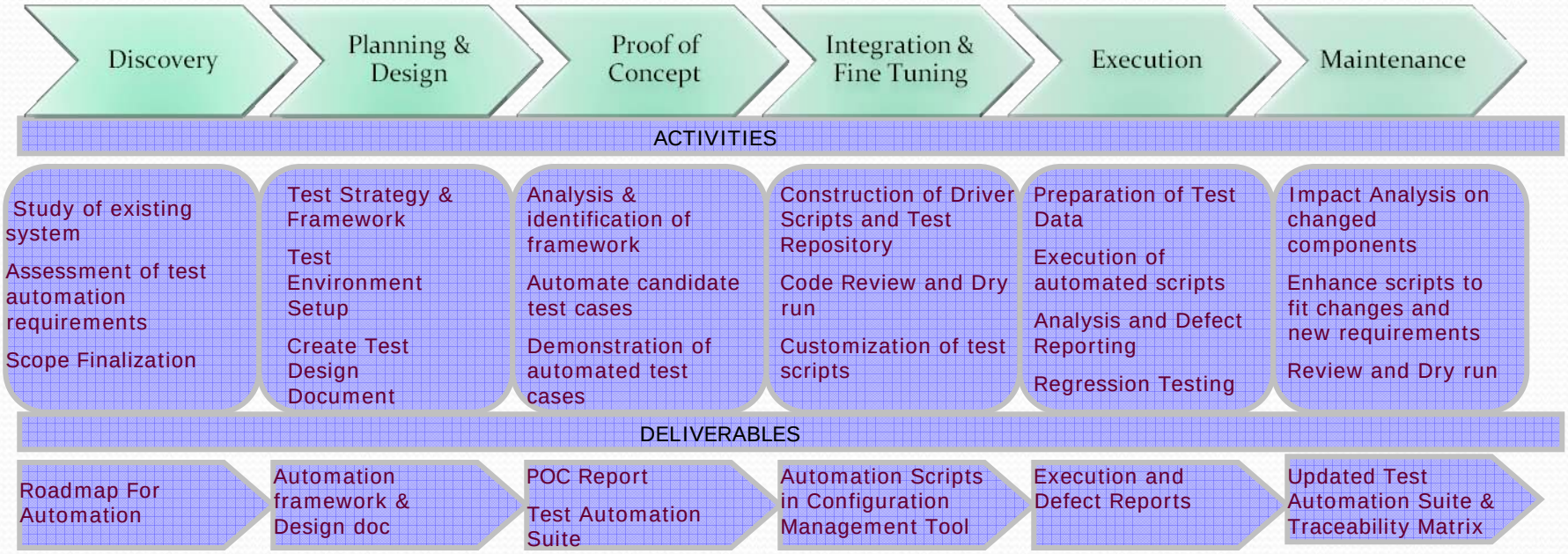
- Creation of automation test scripts.
- Implementation of framework design
- Build libraries
- Dry run of automation scripts

Stage 3 – Integration with Test Management tool

- Integration of scripts with Tools (QTP, QC)
- Execution and baseline of test suites
- Analysis of test execution results



Automation –Approach Summary



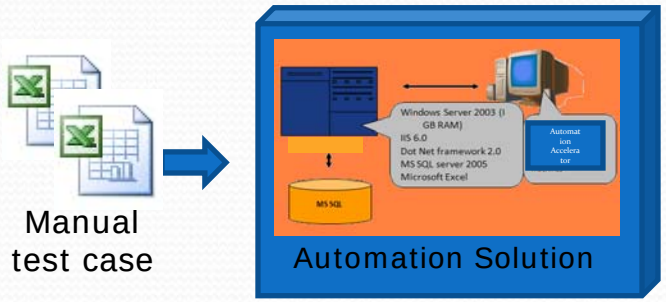
Advance Automation

Automation Methodologies - Comparison

Approach Feature	Record - Playback (with data parameterization)	Keyword driven Framework Approach	Automation Accelerators
Investment / Time to Market / Complexity of Initial Implementation of test automation suite	**Low**	**High**	**Medium**
Allow automated test case scripting of functional test cases prior to delivery of a stable application	X	✓	✓
Simplicity of use	✓	X	✓
Division of Labour between Automation Expert, Test Specialists and Business / Domain Subject Matter Experts	X	⇔	✓
Reuse of common business functionalities to achieve Reduction in Cost-to-Market and Time-to-Market along with Quality Improvements	X	X	✓
Cost / Time to Market / Complexity of Future Maintenance of test automation suite	**High**	**Medium**	**Low**
Ease of Future Maintenance / Ease of Scalability / Ease of incorporating changes in GUI Objects (Window and Control), functionalities and test data	X	⇔	✓
Support for Iterative and Agile development model	X	X	✓
Provision of additional Exception / Error Handling over and above the in-built "Recover Handling" mechanism provided by Automation Tool	⇔	⇔	✓
Capability of generating English Language Documentation of Automated Test Cases	X	X	✓

Legend: X Not Possible ✓ Possible ⇔ Possible, but cumbersome

Automated script generation & Test Automation Accelerators



Manual test case

Automation Solution

Automation script

Object Repository

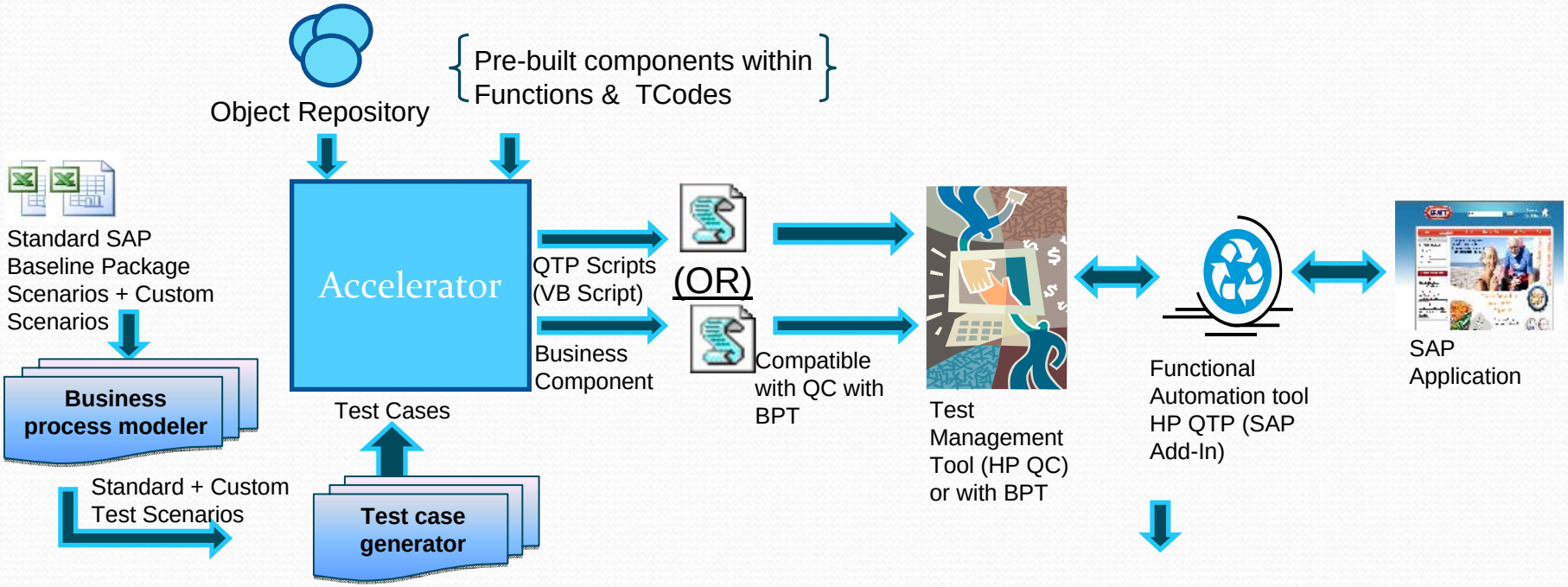
Application Under Test

Automation tool

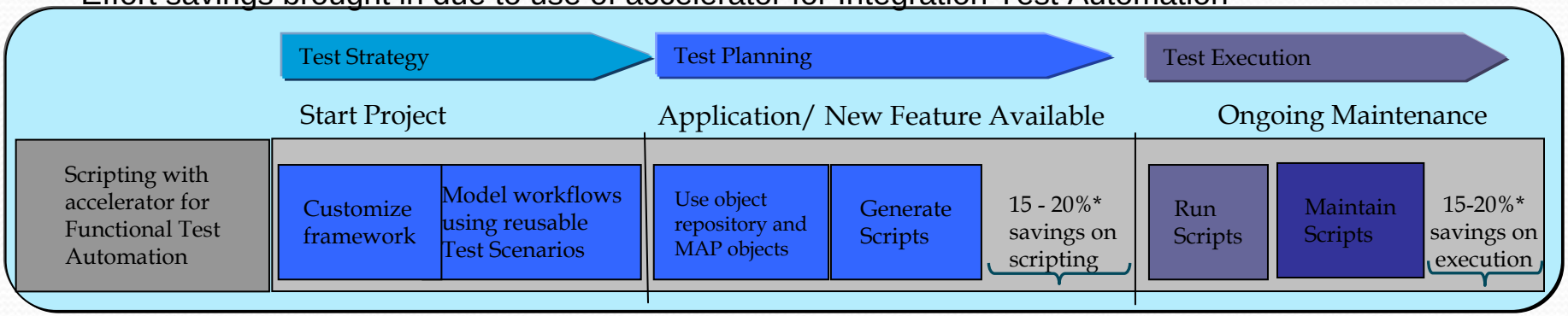
Test Management Tool

- •Accelerator is a friendly web-based application, which generates the automation script automatically
- • Automation can be done by a team having minimal automation experience
- • The automation script generated is modular and structured, making maintenance very easy
- • The framework generated by the solution can be maintained and executed using QTP. There is no further dependency on the solution
- • The solution also provides robust error handling, test case documentation and user friendly execution reports
- • The solution can be integrated with Quality Centre / Test Director

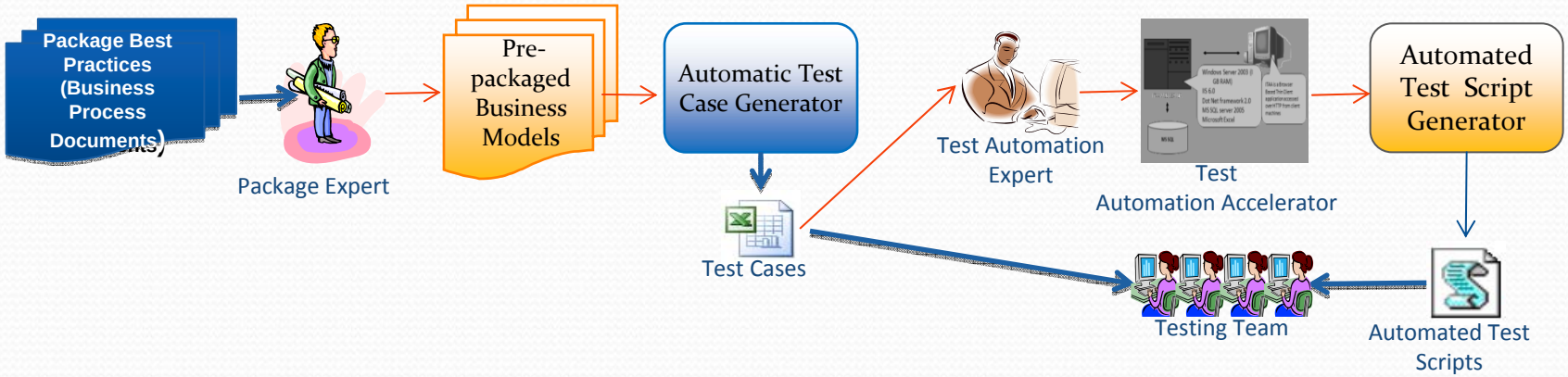
Test Automation Accelerator for ERP testing



Effort savings brought in due to use of accelerator for Integration Test Automation



Model Based Testing tools and Automation Tools are key to drive cost and time to market efficiencies



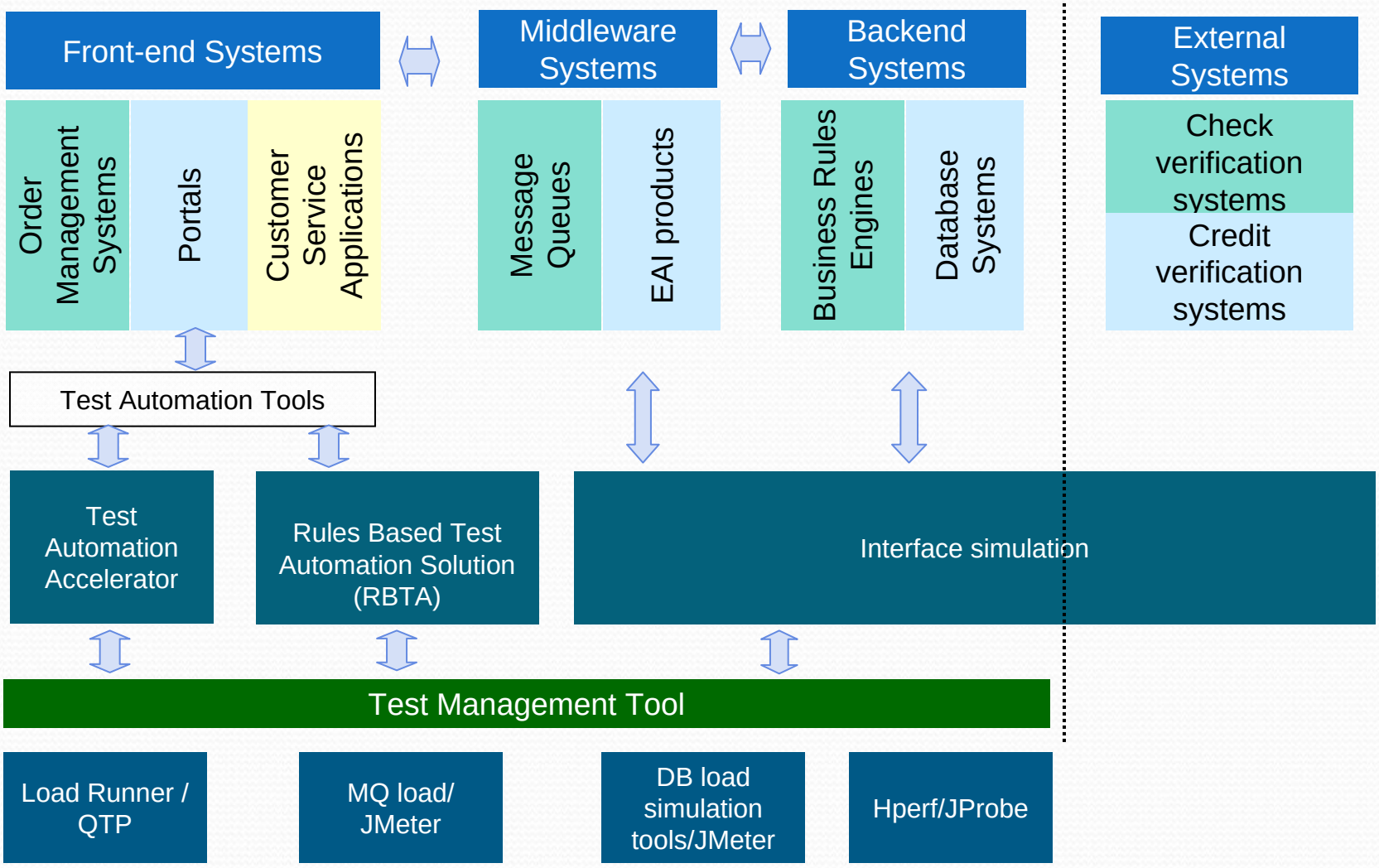
Issues

- Implementing ERP/CRM Packages as part of transformation initiatives requires functional and technical expertise in relevant package modules which may not be readily available in testing teams
- Transfer of Knowledge to the Testing team requires the core package implementation team to spend a lot of time
- Test Automation effort is duplicated across different business processes involving the same package functionality
- Complex Package functionality often requires elaborate test scripts which consume substantial test planning effort

Solutions

- **Use pre-packaged business model based testing solutions** to reduce dependency on package expertise and reduce Knowledge Transfer duration
- Use **automation tools** which can **support early automation even when the test environment is still not ready**
- Use flexible & robust test automation tools which can give the capability to build **automation components, that can be reused across business processes**
- Use **Test Case generation tools** which can **automatically generate test scripts based on pre-defined business model** provided by Package vendors

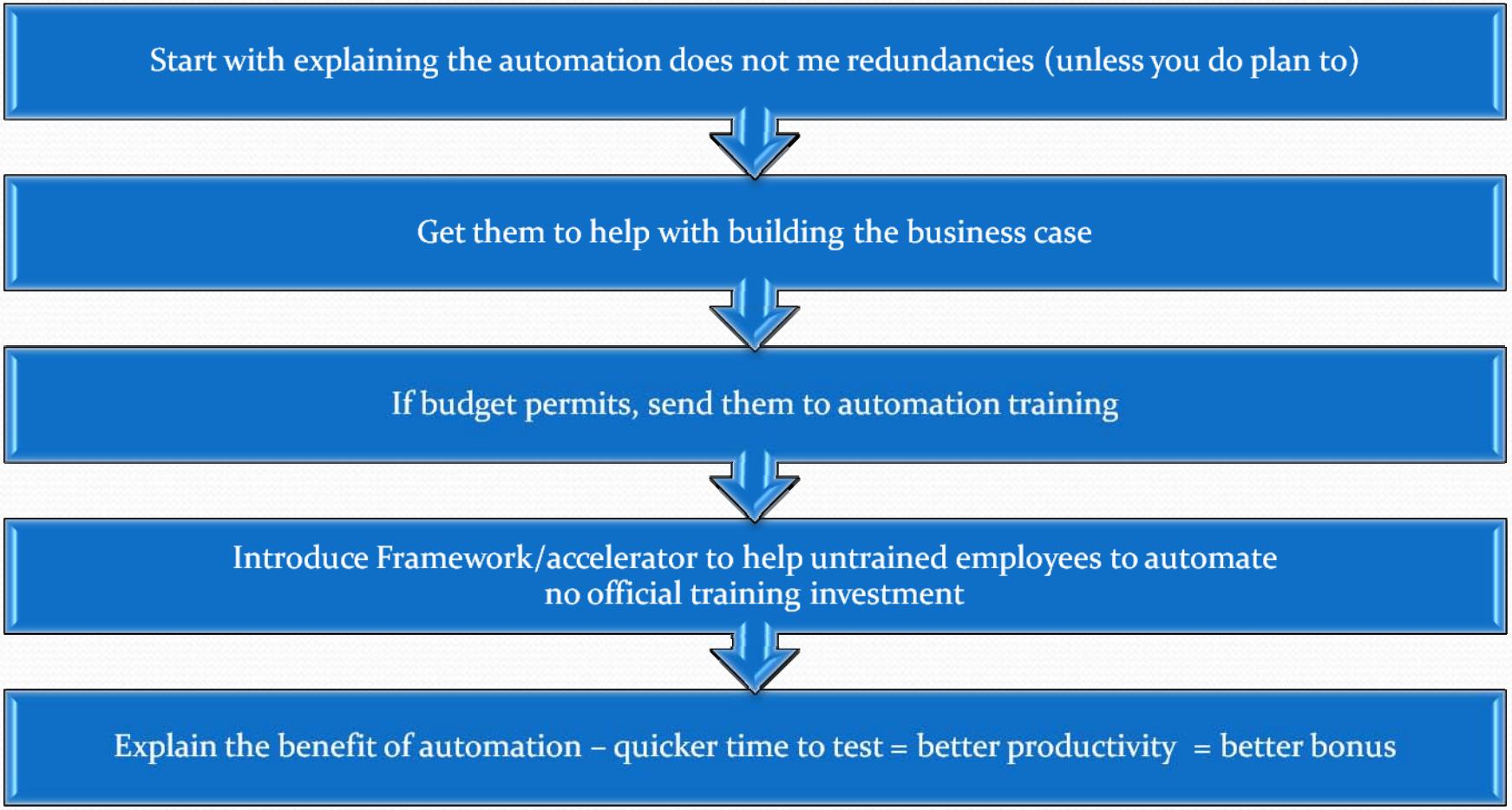
End-to-End Automation



Benefits of Framework

- **Reduced Time to Market:**
  - The availability of a rich, robust in-built framework
  - Allows test case design with logical names
- **Easy to Use:**
  - The **user-friendly User Interface**
  - Well defined framework
- **Low Cost Maintenance:**
  - The Maintenance of scripts due to changes in functionality requires minimal effort
- **Quick Test case Documentation:**
  - Manual **test documentation** generated at the click of a button
- **Integration with Test Management Tool:**
  - Automation Accelerators **integrates seamlessly** with test management tool.
- **In-built Error Handling:**
  - Automation Accelerators provides **Investigation-friendly error messages**
  - **Screen Shots** are captured at **point-of-failure** and saved at pre-defined location

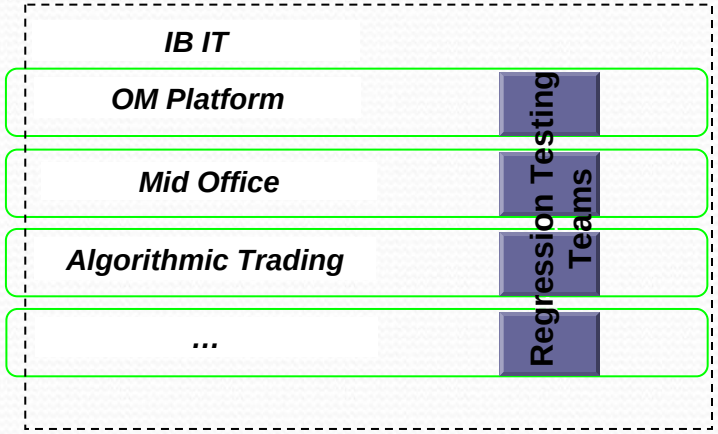
Existing employees



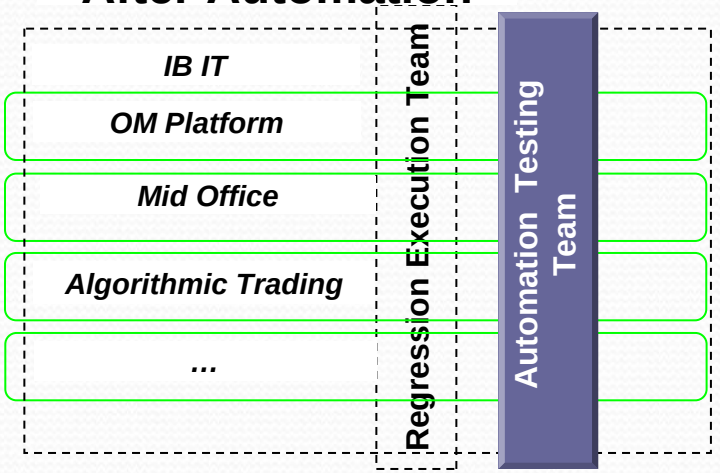
Example Projects

The client has employed automation across various applications and returns are being delivered since last 5 years

Before Automation



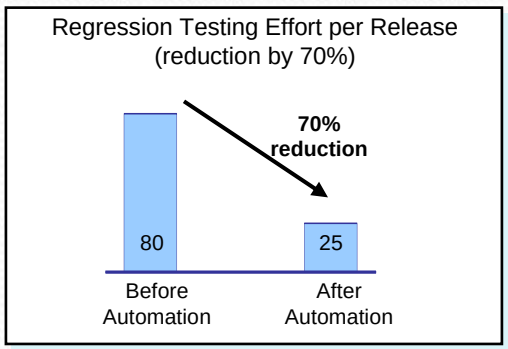
After Automation

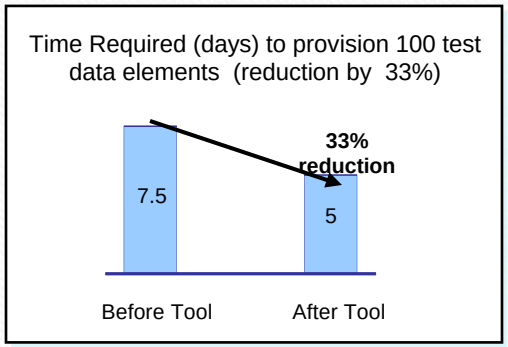


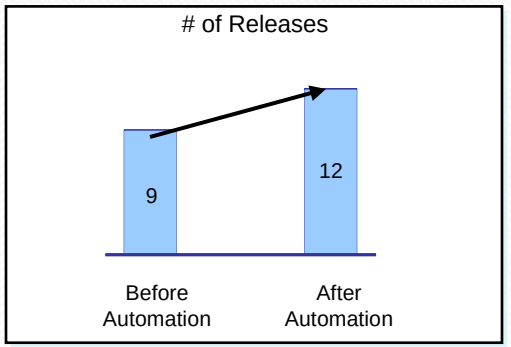
Automated helped reduce effort for every release and hence reducing the release cycle timeline by 2 weeks …

…Automation tool helped reduce data provisioning effort for every release…

…Thus allowing more releases to be delivered within the same time window…







Bad Project

In a giant international retailer, a new test manager wanted to impress...

→ Asked for automation analysis for an ever changing mainframe system

→ Analysis showed – ROI after 83 cycles, if repeated

→ Test Manager decide – Go Ahead !

High Scoring Test Cases	
☑ Tests that need to be run for every build of the application *(sanity check, regression test)*	✗ Yes. BUT ... not the same scripts
☑ Tests that use multiple data values for the same actions *(data driven tests)*	✗ *NO*
☑ Complex and time consuming tests	✗ *Yes and also high consuming to maintain*
☑ Tests requiring a great deal of precision	✗ *No*
☑ Tests involving many simple, repetitive tests	
☑ Testing needed on multiple OS/Browser combination	✗ *No*

Results

- Spent $150,000 on automation which didn't work
- Spent another $50,000 on another company to analyze – assuming the first one wasn't good.
- Although the analysis results were similar – decided to try again
- His manager Stopped him after spending additional $90,000
- Total spending – $290,000 in 6 months for nothing

So Remember

- Look for projects to automate
- Choose carefully
- Look for hidden costs
- Calculate ROI
- USE Frameworks
- Remember – An application does what's its been told - don't imagine a magic button that will create/run automated scripts
- Get your team involved

THANKS

Robby Green

robby_green@infosys.com

Phone: +44-7796268342

Appendix

Tools and ROI

Tool Selection for Automation

PARAMETERS

- Business Criticality
- Operating System
- Language Platforms
- Transaction Volume
- GUI Complexity
- Transaction Complexity
- GUI Standard Requirement
- Technology Uniqueness

Area	Description	Weight %	Average(Tool 1) (on scale of 1 to 3)	Average(Tool 2) (on scale of 1 to 3)
Funtional Evaluatio	Test Requirement Management	10	2.833	2.833
	Test Script planning	30	2.675	2.3
	Advanced planning	5	2	0.667
	Test script execution	25	2.818	2.272
	Test script defect Management	15	3	2.5
	Test execution reporting	15	2.9167	2.667
		Average score	**2.778**	**2.35**
Technical Evaluation	Application architecture	25	2.875	1.9375
	Technical architecture	20	3	2.267
	Deployment	15	2.6	2.4
	Security	20	2.833	2.333
	Development and Maintainability	20	2.53	2.235
		Average score	**2.78**	**2.21**
Vendor Evaluation	Business direction	20	2.667	2.667
	Technical support	25	3	2
	Customer Service	30	3	2.667
	Financial Viability	25	3	3
		Average score	**2.93**	**2.58**

ROI Calculation for Automation (Sample Study)

These figures have been arrived at, based on our past experiences with automation projects

Assumptions:

- Initial cycle includes costs incurred in Test Requirement analysis, Test case design and execution
- Subsequent cycles only deal with execution costs
- 70% automation can be achieved
- Tool costs haven't been taken into account in the calculations. (Assumption : Licenses for QTP are already available)

Type of testing	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5
Manual	70000	119000	168000	217000	266000
With Automation	83720	104370	122570	140770	158970
% Expenditure	119.60	87.71	72.96	64.87	59.76

**Expenditure for each cycle is in dollars, figures are indicative and should not be taken as absolute values.

ROI Calculations for Automation (method 1)

ROI= (Benefits – Cost) / Cost. This is expressed as a percentage.

For a typical iterative testing environment,

No. of Cycles (Re-tests+Regression Tests) = N

Manual Testing Cost/Cycle = M

Fixed Cost of Automation = F

% of Test Time reduction due to automation = S

Then to break-even:

N = F / (M * S) (Here M*S is the benefit due to automated testing)

For Example :

Lets say there is 50% reduction in execution time by automation (S=0.5)

Manual execution cost → $100/cycle

For test suite automation, development cost → $150/cycle

To Break even :

Number of cycles = 150/100*0.5 = 3 cycles

This means that to take up the test case for automation, it should be used for at least three cycles of testing.

ROI Calculations for Automation (method 2)

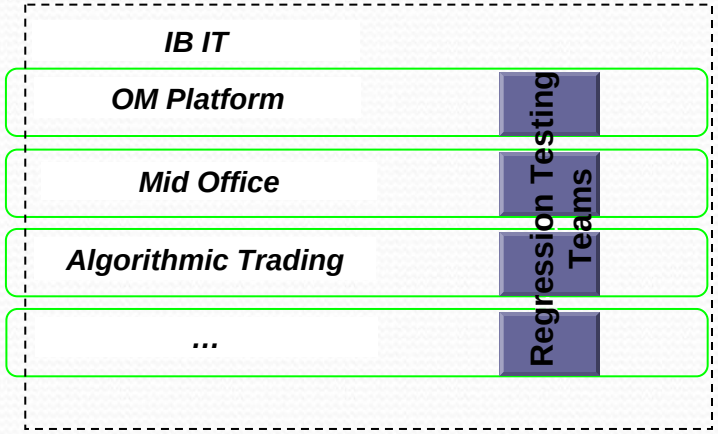
- List down costs for manual testing for each cycle in:
  - Test specification (Vm)
  - Single manual test execution (Dm)
- List down costs for automated testing for each cycle in:
  - Test specification & Implementation (Va)
  - Test interpretation after automated testing (Da)
- Calculate break even of Test automation:
  - N = (Va – Vm) \ (Dm - Da)
  - N = Number of cycles needs to run to achieve break even
- Calculate % increase or decrease in expenditure on each cycle after automation
  - En = (Va + n*Da)/ (Vm + n*Dm)

Automation Projects

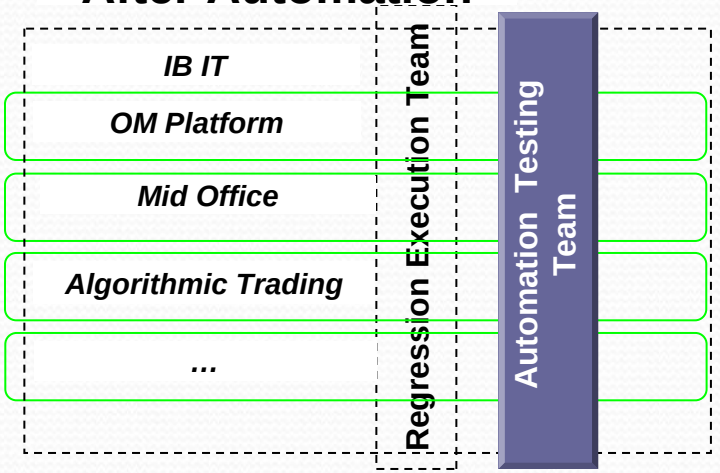
	Detail	Relevance
1	For a leading US based Investment Bank, developed and implemented an Automated Regression Tool for Order Management system in FI and Equities	• Implementation of middleware automated solution in Front Office & Middle Office • Establishing a unified Automation team for the application group
2	For a leading global Investment Bank, delivered automation as a focused track within the QA programme	• Built an automation team to create and maintain automated test scripts across the Investment Bank – Factory Model
3	For an Investment Management leader, delivered one of the largest automation implementations using a structured approach	• High volume of test scripts (90% of 16000 test scripts) • Evaluation of tools and Proof of Concept implementation
4	For a large US Retail and Internet bank, implemented a Data Driven framework for the automation of large set of applications	• Implementation of Data driven Framework • Automation suit used accelerated automation solution

The client has employed automation across various applications and returns are being delivered since last 5 years

Before Automation



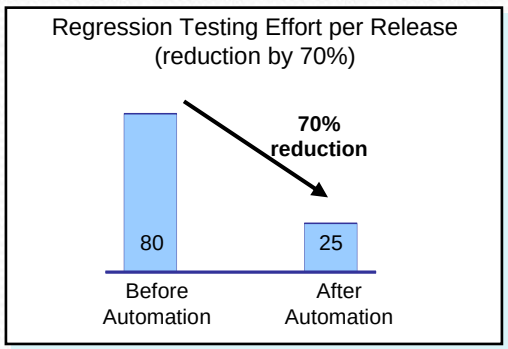
After Automation

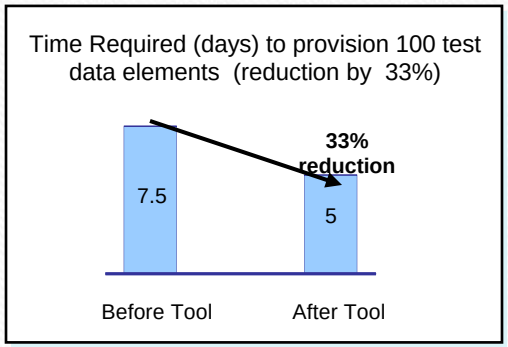


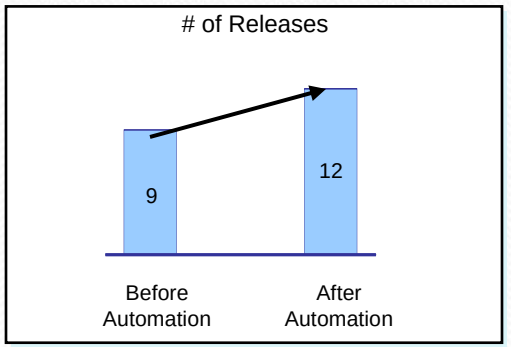
Automated helped reduce effort for every release and hence reducing the release cycle timeline by 2 weeks …

…Automation tool helped reduce data provisioning effort for every release…

…Thus allowing more releases to be delivered within the same time window…







For a leading global Investment Bank, delivered automation as a focused track within QA programme delivering savings of 40%

Key Objective

Establish an Automation track for the QA programme in the Bank to implement automation of test packs across functions

Key Issues facing Client

- The client is a leading European bank offering Global Wealth Management and Investment Banking services
- We had setup a QA function across functions.
- As part of QA Programme, We proactively identified areas / applications where automation could maximize the savings and returns
- For some critical applications, testing was critical as any defects could lead to heavy monetary and business losses

Approach

- Identified areas / applications for automation based on analysis of the recent releases and the effort spent on testing
- Once an application approved by the client for automation, knowledge management team initiates acquisition
- Identify prioritized automation cases and implement the automation solution
- Implement guidelines and repository for Knowledge Management
- Move to other applications / areas based on the priorities agreed with the client
- Ensuring team is updated by conducting domain knowledge and awareness sessions on regular basis

Key Benefits Delivered

- Cost savings - reduction in cost of regression testing by 40%
- Increase in productivity by 30% due to test automation.
- Reduction of time to test by 50%

For an Investment Management leader, delivered one of the largest automation implementations using Implementation Framework

Background

A global leader company with a global presence in 37 countries in the Americas and Asia Pacific. An international provider of financial, retirement and brokerage services

Key Objective

Implement Automation for a group of large applications requiring frequent releases

Key Issues facing Client

- Unstable application environments leading to re-run of testing multiple times
- Complex technical architecture of the applications required judicious selection of automation framework
- Very large number of business requirements making it difficult to monitor and execute all the needed test scripts each cycle

Solution

Implementation Approach

- Automation feasibility study was conducted to validate the automatable test cases as part of the Proof of Concept phase. This also included validating support for third party components
- PoC also included evaluation of tools using a scorecard
- Designed keyword driven framework for automation of the application suite
- Automation implemented for 7 applications and 90% of 16000 test scripts
- Identified common functionalities across all applications and developed reusable components
- Created specific suites for Smoke, Regression and Production validation cycles

Maintenance Approach

- Designed a robust Script Maintenance methodology to handle script maintenance for applications enhancements

Key Benefits Delivered

- 90% of the regression test suite was automated.
- 100% schedule adherence across all releases

For a large US Retail and Internet bank, implemented a Data Driven framework for the automation of large set of applications

Background

A diversified financial services company that provides a broad range of retail banking, brokerage, asset and wealth management and corporate and investment banking services in the United States

Key Objective

Prepare for shorter release cycles to ensure inclusion of all strategic quality assurance activities

Key Issues facing Client

- Automation in the existing applications was limited with no structured framework
- Minimal benefits being achieved with potential to accelerate savings

Solution

Implementation Approach

- Analysis on the previous framework was done to identify the feasibility of re-use
- Functional Decomposition framework was fine-tuned to Data Driven framework with consideration of performance enhancement
- A phase wise approach was followed for automation with high priority applications in initial phases
- Acceleration tool used for test case modeling resulting quicker implementation timelines

Maintenance Approach

- Necessary checklist for execution of the scripts were created and are being maintained
- New enhancements, bug fixes are analyzed and necessary changes made to the scripts as part of standard review process

Key Benefits Delivered

- Reduction in execution time by 80% as compared to manual execution efforts.
- Highly structured, detailed, maintainable automated test script repository was created
- Developed an automation framework that could be applied to high number of applications