
Smart Ways to Reduce Regression Testing Cycle Times

Sivamani (Shiva) Jayaraman

Feb 2011

Australia | Canada | China | France | India | New Zealand | Singapore | Switzerland | UAE | UK | USA

CONFIDENTIALITY STATEMENT

© 2009 Keane, The concepts and methodologies contained herein are proprietary to Keane Inc. Duplication, reproduction or disclosure of information in this document without the express written permission of Keane is prohibited.

Keane

Agenda

- Why Testing Takes Forever?
- Common mistakes
- Test Automation, Really?
 - Regression Test Strategy
 - Data Management Strategy
 - Environment Management Strategy
 - Automation Strategy
- Techniques
 - Automation ROI Calculator
 - Visual Test Design
 - Follow The Sun



Agenda

Why Testing Takes Forever?

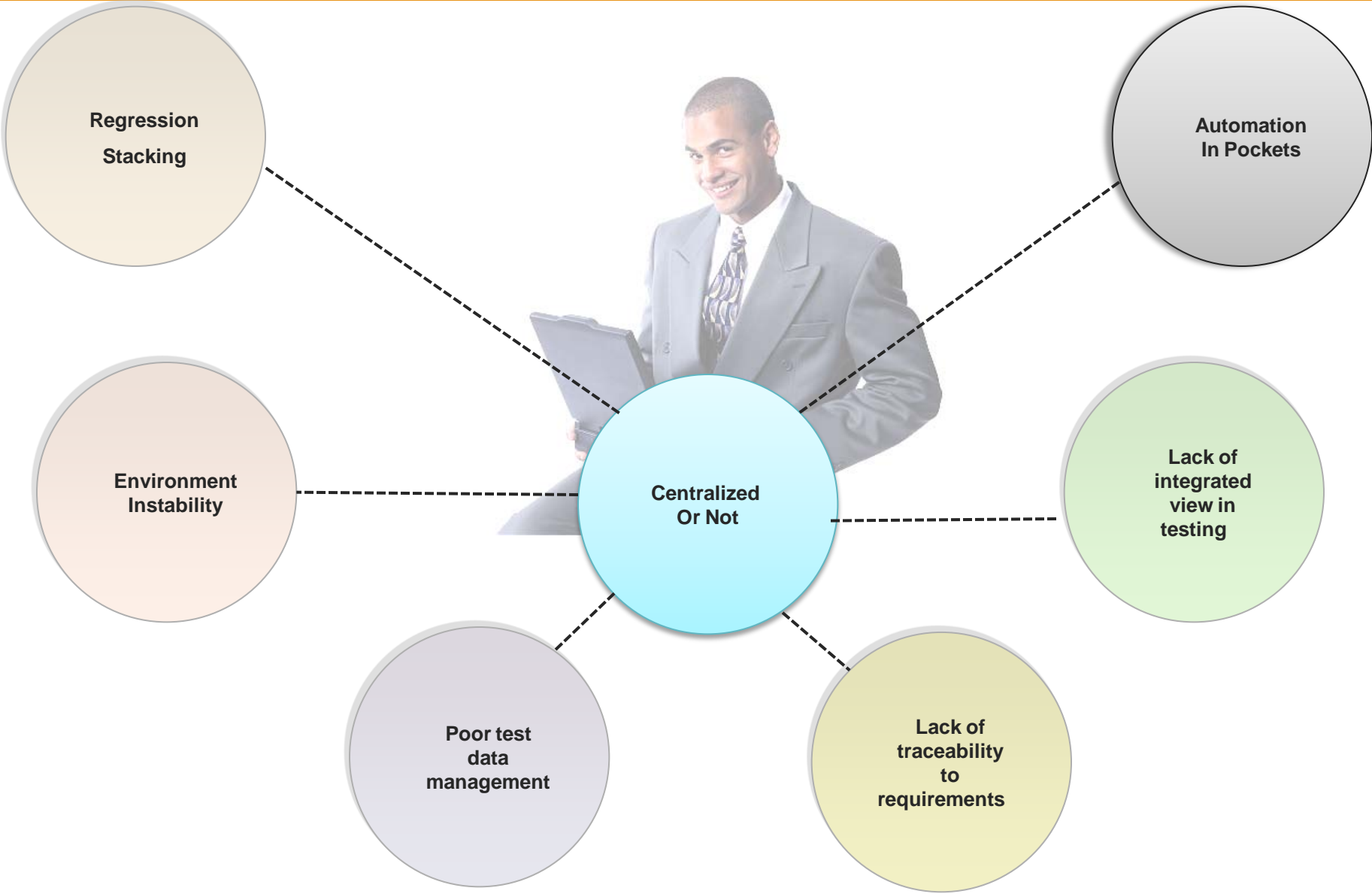
Testing is exponential to time

Lack of clarity on what to test

Enterprise testing is not metric driven

- **Too late** – Even today, testing team is not part of the estimate sent before project charter or Request For Expenditure
- **Some practitioners are just passionate** – Not all great testers become great test managers. Some are caught up into ‘testing everything’ business.
- **Everything In A Small Box** – With dysfunctional test organizations, a LOT of testing is done for a small set of applications, leaving behind a LOT untested.
- **Maintenance Costs** – Most organizations do not measure the efficiency of their testing in relation to their application / product maintenance costs in in production
- **Regression – Again and Again!**– Testing teams spend way too much time to complete regression testing due to poor software builds that keep coming

Common Mistakes



Test Automation – Really?!

Head Of Department: “Regression testing takes long time. Can we not automate something?”

QA Manager: Sure!

Head Of Department (in his mind): Lights Out Testing!

QA Manager (in his mind): Ha! Finally, something to automate!

Is automation the right solution? , Is there a clear

- Regression test strategy
- Data management strategy
- Environment management strategy
- Automation strategy

Really, is automation going to be able to reduce the regression testing time lines?



Regression Test Strategy

- Is it critical business functionality?
- Are there redundant business scenarios?
- Modularized to perform selective regression testing in future?
- Can recommendation to clients be offered with an end user perspective?
- Can risks in not selecting a candidate be clearly explained to clients?
- Is there focus on the defect prone zones from previous releases?
- Can this avoid 'regression stacking'?



People



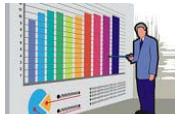
Tools



Process



Global Delivery



Metrics



Training



Centre of Excellence

Data Management Strategy

Which first - test data management strategy OR automation framework?

Data Feeder Scripts

Time and effort spent on creating dedicated data for scripts can be avoided

Data Generators

On - demand access to volumes of data capsules for various types of testing

Data Refresh Scripts

Data refresh strategies are very less expensive if there is a strong understanding of the database. Data refresh strategies also exponentially improves automation efficiency

Capsuling Best Practices

There is very minimal effort involved. This will enable the scripts self dependent



People



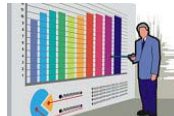
Tools



Process



Global Delivery



Metrics



Training



Centre of Excellence

Environment Management Strategy

- Is there a controlled QA environment?
- Are environment variables managed as per specifications?
- Are there audit trails for administration in environment?
- Is the environment operable and scalable to meet test automation needs?
- Is the environment flexible for ever changing delivery models?
- Is the environment conducive for parallel development?



People



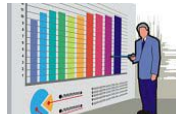
Tools



Process



Global Delivery



Metrics



Training



Centre of Excellence

Automation Strategy

- Do the regression candidates belong to repetitive business functionality?
- What is the environment stability?
- Data categorized into static and dynamic?
- Is the automation framework robust?
- Is there SDLC discipline in managing changes?
- Are the scripts modularized?
- Is analysis complete to isolate data preparation activities in order to reduce the execution timelines.?
- Is there determination on scripts that need monitoring and those that can run unattended?

Analysis & ROI Calculator - Automation

Project	Manual Hrs (per execution)	Manual Cost (per execution)	No. of execution times (per year)	Total Manual Cost	Automation Hrs (per execution)	Automation Cost (per Execution)	No. of execution times (per year)	Total Automation Cost	Gross Savings due to Automation	% of Maint	Maint Cost	Effort to Automate (hrs)	Investment Amount	Net Savings due to Automation(\$)
X	400	28,000	5	140,000	96	3,840	5	19,200	120,800	5	6,336	1584	63,360	51,104

Input Fields

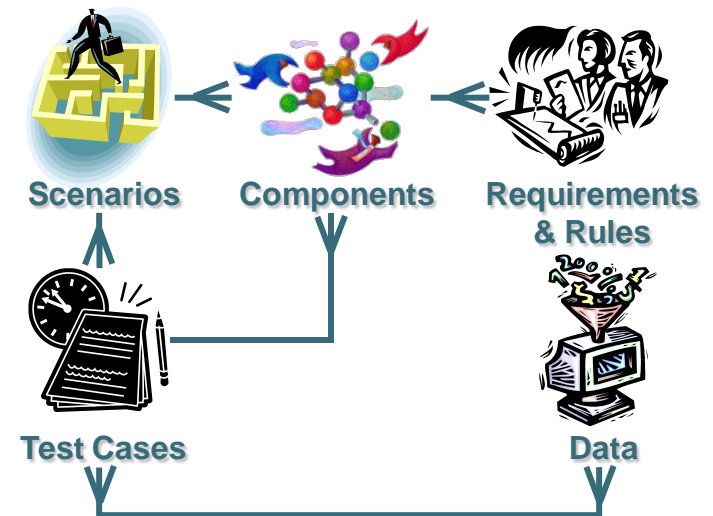
S.No.	Test Case Name / Functionality	Is it a base functionality?	Candidate for automation	Comments	Complexity for automation in a scale of 1-10	Nbr of related TC's		
						Base	To Be Automated	NA
1	Admin Properties_TC_01	Yes	Yes		6	1		1
2	Admin Properties_TC_02	Yes	Yes		8	1		1
3	Admin Properties_TC_03	Yes	Partial	Functionality related to sending mail cannot be automated	6	1		1
4	Compare_TC_01	Yes	Yes		7	1		1
5	Compare_TC_02	Yes	Partial	Functionality to verify whether the 3M Trading Partner Information is printed, cannot be automated.	6	1		1
6	Forgot Password_TC_02	Yes	Partial	Send Email Functionality cannot be automated	5	1		1
7	Page footer_TC_01	Yes	Yes		6	1		1
8	TPSearch_TC_01	Yes	Yes		6	1		1
9	TPSearch_TC_02	Yes	Yes		6	1		1
10	TPSearch_TC_03	Yes	Yes		6	1		1
11	TPSearch_TC_04	Yes	Yes		6	1		1
12	TPSearch_TC_05	Yes	Yes		6	1		1
13	TPSearch_TC_06	Yes	Yes		6	1		1
14	TPSearch_TC_07	Yes	Yes		6	1		1
15	TPSearch_TC_08	Yes	Yes		6	1		1
16	TPSearch_TC_09	Yes	Yes		7	1		1
17	TPSearch_TC_10	Yes	Yes		5	1		1
18	TPSearch_TC_11	Yes	Yes		6	1		1
19	TPSearch_TC_12	Yes	Yes		6	1		1
20	TPSearch_TC_13	Yes	Yes		6	1		1
21	TPSearch_TC_14	Yes	Yes		6	1		1

Automation Analysis Summary	
Automation : %	98
Total no of test cases :	42
Test cases to be automated:	41
Test cases to be executed manually:	1

Component Approach

The Component approach assists in reducing the Total Cost of Ownership (TCO) that is associated with developing & maintaining a manual and automated test script library

- Decomposition of a business scenarios into components
- Supports the following
 - Earlier testing (Iteration, Agile, etc.)
 - Sharing of common components
 - Unit, stringing one or units together and Business scenario testing
- Automation utilizes logic to extend the capabilities of record and playback
 - Supports conditional statements (i.e., branching)
 - Error handling
 - Supports SQL and data analysis
- Externalized data support
 - Data driven
 - Easy updates
 - Support audits based on test results
 - Positive and negative testing
- Every component has one or more measurable results



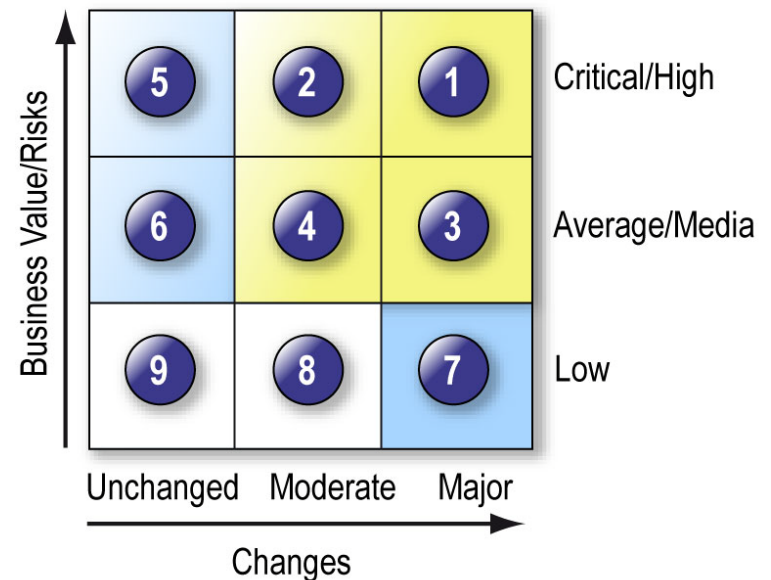
Advantages

- Reduces total # of recorded scripts
- Update one component to update many Test Cases / Scripts
- Supports all aspects of Functional, Regression, System Integration and Performance testing

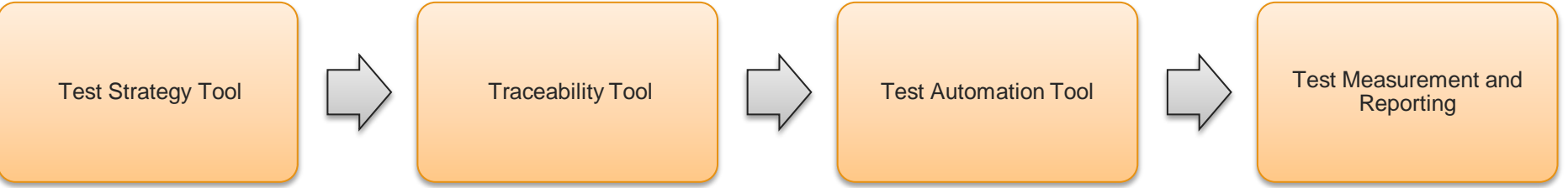
Risk-Based Testing

Because optimal testing is based on a balanced assessment of risk against time, scope, and cost, a risk-based approach that emphasizes awareness of the strategic business relevance of testing in addition to the tactical is fairly important

- Improves the quality of applications while optimizing resource utilization and supporting on-time deployment
- Allows management to balance the cost of resources to eliminate business risk (defects) and the costs to the business of delivering those defects in production applications
- Provides an objective method of prioritizing what to test based on risk
- Delivers disciplined repeatable processes and reusable assets shortening future release cycles
- Defines best practices for every step in the SDLC

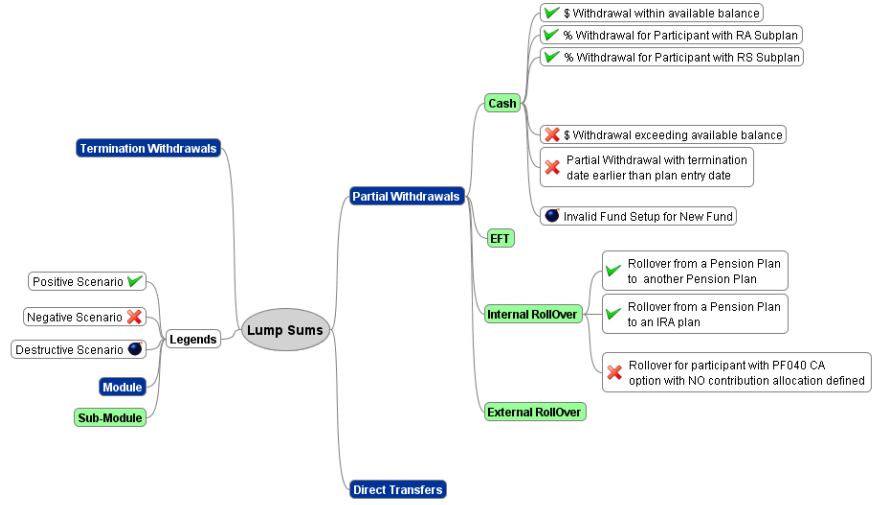


Visual Test Design & Risk Based Testing



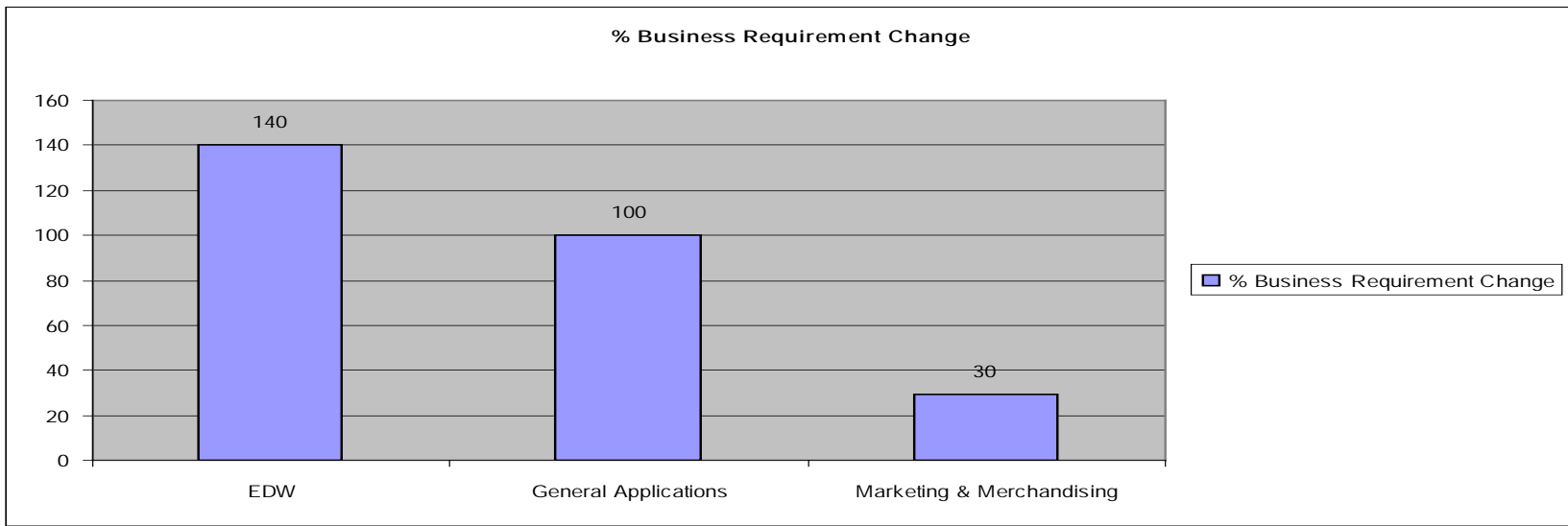
- ### Traceability Tool - Visual Test Design
- Enables better requirement coverage
 - Reduces on - boarding time by 30%
 - Reduces test scenario verification time by 20%
 - Easy to identify re - usable test scenarios, improves productivity

- ### Test Strategy Tool - Risk - based Testing
- Improves the quality of applications while optimizing resource utilization and supporting on - time deployment
 - Provides an objective method of prioritizing what to test based on risk
 - Delivers disciplined repeatable processes and reusable assets shortening future release cycles



Business Value/Risks ↑	Critical/High	5	2	1
	Average/Media	6	4	3
	Low	9	8	7
		Unchanged	Moderate	Major
		Changes →		

Early Defect Detection – Reduces Regression Time!



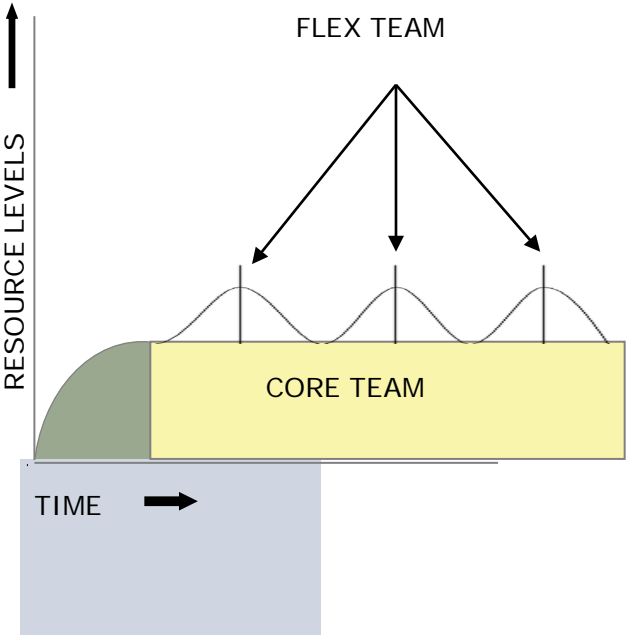
Domain	Business Requirements	Resulted in Business Requirement Change	% Business Requirement Change
EDW	57	80	140
General Applications	4	4	100
Marketing & Merchandising	169	50	30

Follow The Sun

Tenets Of Follow The Sun model

- Dev. team turns QA builds during end of day
- QA team completes 1st round of testing before next morning
- Dev. Team fixes / Core testing team re - tests
- Dev. Team turns the QA build by end of day
- Test team completes 2nd round of testing before next morning

Reduces Regression Cycle Time By 30 - 50%



Core Team

- People knowledgeable about client’s business processes, culture, technical environments, and applications
- Enables quick ramp - up for new projects
- Provides continuity of key resources
- Maintains client’s knowledge base
- Provide value - add services during “down time” thorough test automation, strengthening regression suites

Flex Team

- Responds quickly to constantly changing resource requirements
- Optimizes resource utilization across projects and over time
- Effective on - boarding, orientation, and training programs
- Skilled quality engineers that are knowledgeable in testing tools and techniques
- Consistent quality of service ensured by repeatable processes, reusable artifacts, and process controls