

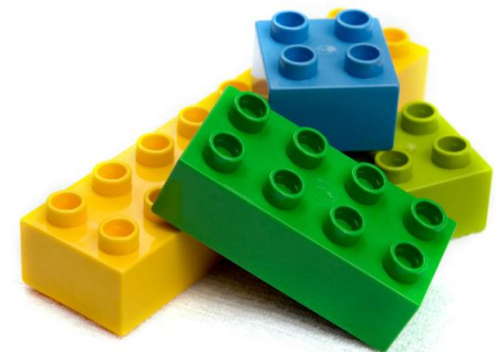
The Black Swan of Software Testing

– **An experience report on exploratory testing**

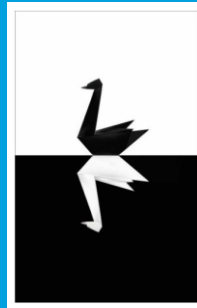
Main Author & Presenter: Shaham Yusuf

Co-Author: Venkat Moncompu

October 2012



- Till 17th century it was believed that all species of swans were white.
- Every bit of empirical evidence confirmed this as true.
- By the discovery of black Swan in Australia destroyed all previous notions that all swans are, and must be, white.



Nassim Nicholas Taleb, the author of the book “The Black Swan: Impact of the highly improbable” says, “*one single sighting of black swan can invalidate the belief that all swans are white*”.

This presentation is such demonstration of the black swan of testing which might help us to shift our mindset from the granted belief of scripted testing.

Contents

Abstract

Case Study – Traditional Approach

What is the problem in traditional approach?

What do we need to do to address problems of traditional approach?

What is Exploratory Testing and what are the challenges?

The Solution - Alternative Test Design techniques with exploratory testing

Do it yourself and conclusion

Abstract

Applications are nowadays being developed around Social Business, Enterprise Mobility, Gamification, Hyper-hybrid Clouds etc. which intrigues us to take more out of them – the world is changing. The end user renaissance of application usage is with the better User Experience (UX), consistency, and a software which work as per expectation of gen Y users.

As testers are we geared up to test sheer complex applications of this changing world, or are we still test designing and testing the old fashioned way? ?



Abstract

Case Study – Traditional Approach

What is the problem in traditional approach?

What do we need to address problems of traditional approach?

What is Exploratory Testing and what are the challenges?

The Solution - Alternative Test Design techniques with exploratory testing

Do it yourself and conclusion



Case Study – Traditional Approach

Test Design

Test Cases Created	Test Case Complexity	Average time spent <i>Creation</i> of each Test Case (hours)	Total time spent on Test Case <i>Creation</i> (hours)	Average time spent <i>Review</i> of each test case (hours)	Total time spent on Test Case <i>Review</i> (hours)	Total Time spent on Test Design (Hours)
706	Simple	1	706	0.5	353	1059
1178	Medium	2	2356	0.75	883.5	3239.5
472	Complex	3	1416	1	472	1888
Total						
2356			4478		1708.5	6186.5

Key Observation:

Significant amount of project time, 6186 hours, was spent on Test Design. With the team of 12 resources (including test lead), we spent over 3 months of time in test design phase (considering 45 hours per week per resource).

Case Study – Traditional Approach (contid.)

System Integration Testing (SIT) Defects

Test Cases Executed	Test Case Complexity	SIT Defects Detected by Test Cases	SIT Defects Detected on Ad-hoc basis (does not maps to a test case directly)	Total SIT Defects Detected
706	Simple	69	65	134
1178	Medium	104	96	200
472	Complex	24	39	63
Total				
2356		197	200	397

Key Observation:

After going through robust test design phase, about 50% of the total SIT defects were detected on Ad-Hoc basis, i.e., they were tangential to the test cases.

Case Study – Traditional Approach (contid.)

SIT v/s UAT Defects

Number of SIT Defects	397
Number of UAT Defects	50

Key Observation:

There was significant defect leakage to UAT ($(50/397)*100 = 12.5\%$).

Abstract

Case Study – Traditional Approach

What is the problem in traditional approach?

What do we need to address problems of traditional approach?

What is Exploratory Testing and what are the challenges?

The Solution - Alternative Test Design techniques with exploratory testing

Do it yourself and conclusion



What is the problem in traditional approach?

Test Design:

1. Typically 40% of the Testing Phase time is spent on 'Test Design' which is a major chunk.
2. Any large and complex application typically has thousands of test cases which makes it cumbersome to maintain them; and create them in first place.
3. It is not uncommon to see requirement gaps and new 'discover-as-we-build' items coming up during actual testing. This makes it practically difficult to update test cases during that time as we are time pressed during test execution.
4. Sometime testers are too caught up in the perfection of test design / scripts rather than focusing on execution/testing.

Business:

1. We do not get much time to gain business knowledge of the application.
2. Go-to time (time to begin test execution) is more, as much time is spent on planning and test designing.

Execution:

1. Defect leakage in UAT (User Acceptance Testing) is high.
2. Less defect count in SIT (Systems Integration Testing).
3. Focus on predefined test scripts is more during execution no matter if the application is changing dynamically.

Abstract

Case Study – Traditional Approach

What is the problem in traditional approach?

What do we need to do to address problems of traditional approach?

What is Exploratory Testing and what are the challenges?

The Solution - Alternative Test Design techniques with exploratory testing

Do it yourself and conclusion



What do we need to do to address problems of traditional approach?

Test Design:

1. We need a light weight test designing techniques which can significantly reduce test designing time.
2. Our scripts should be easy enough to make quick updates on the fly.
3. Our scripts should be on a bit high level so that we can accommodate changes, CRs, design gaps etc.

Business:

1. We need to gain business knowledge of the application.
2. Testing team should be in a position to begin test execution as early as possible.

Execution:

1. Defect leakage in UAT (User Acceptance Testing) should be minimal.
2. Detect exponentially high number of defects, compared to traditional approach.
3. Focus more on the behavior of the application rather than the nitty-gritty of test scripts

Abstract

Case Study – Traditional Approach

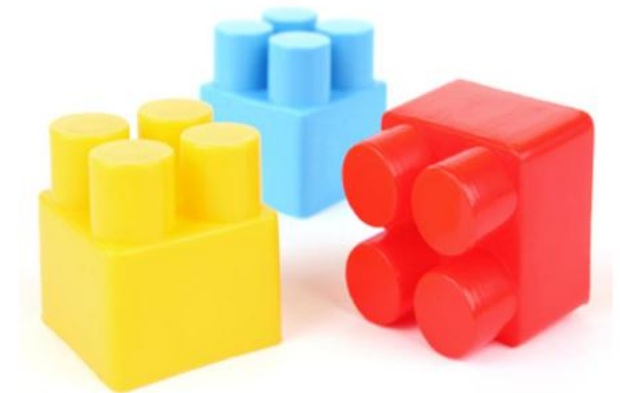
What is the problem in traditional approach?

What do we need to address problems of traditional approach?

What is Exploratory Testing and what are the challenges?

The Solution - Alternative Test Design techniques with exploratory testing

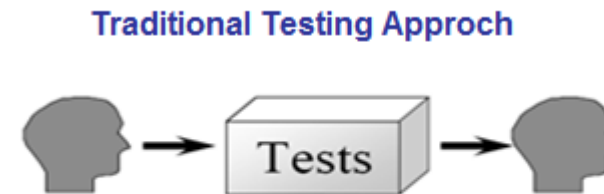
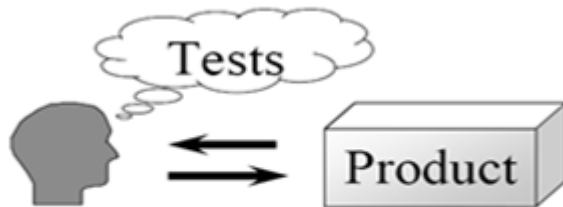
Do it yourself and conclusion



What is Exploratory Testing and what are the challenges?

What is Exploratory Testing:

- Not 'Ad-hoc testing' or 'Random' testing',
 - but a "context driven" approach
- Not an extension of test case execution,
 - but an execution with investigation and contextual test design
- Not undefined / unstructured testing,
 - but a structured and accountable approach
- Not a technique,
 - but an approach to test
- Not quick testing,
 - but dedicated work with learning
- Skills of Exploratory Testing can be acquired by practice



What is Exploratory Testing and what are the challenges?

Challenges:

Primarily there are 4 typical challenges which make the traditionalists reluctant to adopt Exploratory Testing formally in engagements.

Test Design: Usually people think that test designing is poor as it is an exploratory/ad-hoc approach hence may affect the project deliverable.

Accountability: The word exploratory gives an impression to layman that accountability cannot be established hence they are reluctant to adopt it.

Learning styles: It is a usual notion that heavy documentation of pre-defined test cases will help the team members to learn the application quickly but exploratory approach can come in the way of knowledge transition.

Negotiated Contract: Negotiated contract with Client comes with many pages of requirement document for which we need to showcase the coverage.

Abstract

Case Study – Traditional Approach

What is the problem in traditional approach?

What do we need to address problems of traditional approach?

What is Exploratory Testing and what are the challenges?

The Solution - Alternative Test Design techniques with exploratory testing

Do it yourself and conclusion



The Solution - Alternative Test Design techniques with exploratory testing

The Solution:

Leverage Exploratory Testing approach with Alternative Test Design Techniques.

Alternative Test Design techniques :

Scenarios – *scenario based testing complemented by test execution narration*

- We used high level test scenarios for designing complex business rule type of tests

Flow Diagrams

- We used flow charts for designing end-to-end type of tests

Mind Maps

- Mind Maps were used to jot down spontaneous ideas which flow during JAD (Joint Analysis & Design) sessions and requirement analysis phase.

Cheat Sheets

- We used cheat sheets for existing/old modules, primarily to regress the system.

Checklists

- Checklists were extensively used to design nitty-gritty types of test e.g., Date should be in GMT, Text should be Ariel Black, size 14 etc.

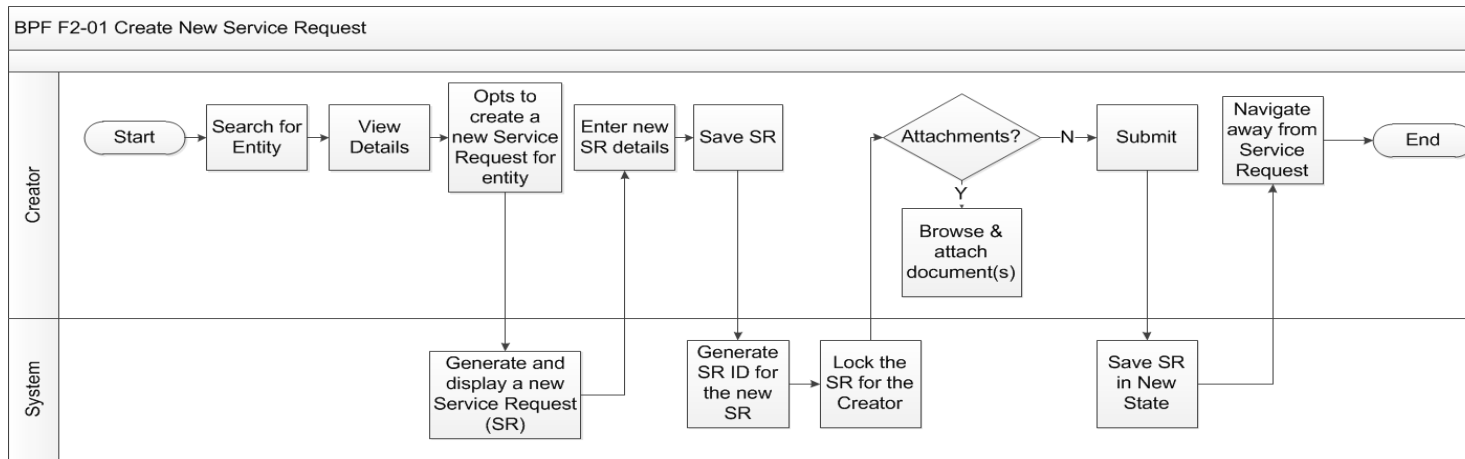
The Solution - Alternative Test Design techniques with exploratory testing

Scenarios:

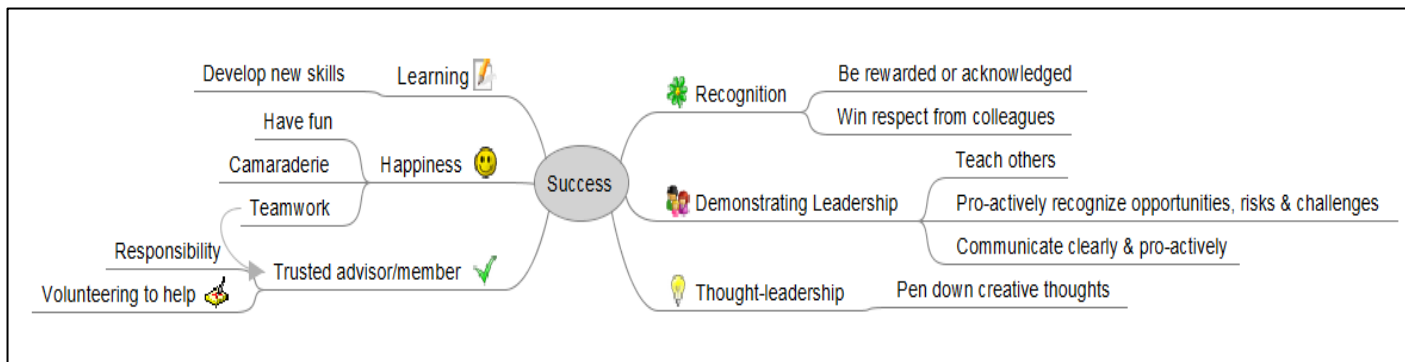
Scenario 01				
Objective:	To verify the "Show Changes Only" checkbox is functional when user navigates to ABC XYZ Tab of an Add XYZ Change request			
Requirement Details				
Related Defects	123456			
Input	The ABC summary page with add sample page is displayed			
Pre condition:	<ol style="list-style-type: none"> 1. User should have ABC rights. 2. System has ABC state 			
Process	Perform actios on "Show Changes Only" checkbox			
Output	The checkbox functions and shows the changes			
Secondary Expected Results	Other functions are intact			
Additional information (for testing)	There was a related defect in previous release, ID 123456			
Checklist	<ol style="list-style-type: none"> 1. Verify in view mode also 2. Verify in all states 			
Matrix				
Initial State	Data Input	Condition	Process	End State
ABC State	Add Sample Page	With abc data	Add	Gets added
XYZ State	Add Sample Page	With xyz data	Remove	Gets removed

The Solution - Alternative Test Design techniques with exploratory testing

Flow Diagrams



Mind Maps



The Solution - Alternative Test Design techniques with exploratory testing

Cheat Sheet

Cheat sheets are useful to test a known functionality or module. We basically used them to regression test the existing modules.

Checklists

Check List		Description	Iteration1	Iteration2	Iteration3	Iteration4	Iteration5	Iteration6
	#							
Text	1	Headings 24, Arial Blue	✓	✓				
	2	Sub headings 16, Arial, Black	✓	✓				
	3	Text 14, Arial black	✓	✓				
	4	All text are localizable	✓	✓				
	5	Roman, Hebrew, Chinese are accepted	✓	✓				
Date	6	Date in GMT	✓	✓				
	7	Start date is never greater than end date	x	✓				
	8	End date is never more than current date	✓	✓				
General	9	Bread crumb is always clickable, expect the current page (last one)		✓				
	10	Bread crumb is Home > module > sub module > page	✓	✓				
	11	Enter key begins search	x	✓				
	12	Sequence of Tab key is first field to last field	✓	✓				
	13	Escape works as cancel	✓	✓				
	14	Help menu is present on each page	x	✓				
	15	Buttons have key commands	✓	✓				
	16	All buttons are all of similar size and shape, and same font & font size	✓	✓				
	17	Mouse and keyboard action are consistent in all pages	✓	✓				
	18	Session extension appears after 20 minutes	✓	✓				
	19	Session extension appears twice and then system logs out	✓	✓				
	20	Checking the check box refreshes the screen	✓	✓				
	21	Lists are scrollable	✓	✓				
22	Arrow key works in list	✓	✓					
23	All windows have a consistent look and feel.	✓	✓					
24	Numeric field does not accepts other characters	✓	✓					
25	All numeric fields have field length of 20	x	✓					

Case Study – Alternative Test Design Approach

Test Design

Test Design - Artifacts	Number of Test Artifacts	Average time spent <i>Creation</i> of each Test Artifact (hours)	Total time spent on Test Artifact <i>Creation</i> (hours)	Average time spent <i>Review</i> of each test case (hours)	Total time spent on Test Case <i>Review</i> (hours)	Total Time spent on Test Design (Hours)
Scenarios	481	1.5	721.5	1	481	1202.5
Flow Diagrams	55	1.5	82.5	0.5	27.5	110
Mind Maps	20	2	40	1	20	60
Cheat Sheets	15	1	15	0.5	7.5	22.5
Checklists	27	1	27	0.5	13.5	40.5
Total	598		886		549.5	1435.5

Key Observation:

Considerable amount of Test Design time is saved due to light weight test designing.

Case Study – Alternative Test Design Approach (contid.)

System Integration Testing (SIT) Defects

Test Artifacts Executed	Test Artifacts Executed	Defects Detected by Test Artifacts	Defects Detected on Ad-hoc basis (does not maps to any of the test artifact directly)	Total Defects Detected
Scenarios	481	289	11	300
Flow Diagrams	55	23	1	24
Mind Maps	20	34	6	40
Cheat Sheets	15	3	8	11
Checklists	27	52	0	52
Total	598	401	26	427

Key Observation:

Most of the detected defects were the outcome of the execution of the high level test artifacts. Only 6% of the total defects were detected on Ah-Hoc basis; thus the designed artifacts were leveraged completely and effectively.

Case Study – Alternative Test Design Approach (contid.)

SIT v/s UAT Defects

Number of SIT Defects	427
Number of UAT Defects	10

Key Observation:

Defect leakage was on 2.3% as the high level test artifacts allowed testers to explore like end users.

Note: Client used pre-defined test scripts for UAT prior to implementation of our alternative Test Design Techniques as well as after the implementation.

Abstract

Case Study – Traditional Approach

What is the problem in traditional approach?

What do we need to address problems of traditional approach?

What is Exploratory Testing and what are the challenges?

The Solution - Alternative Test Design techniques with exploratory testing

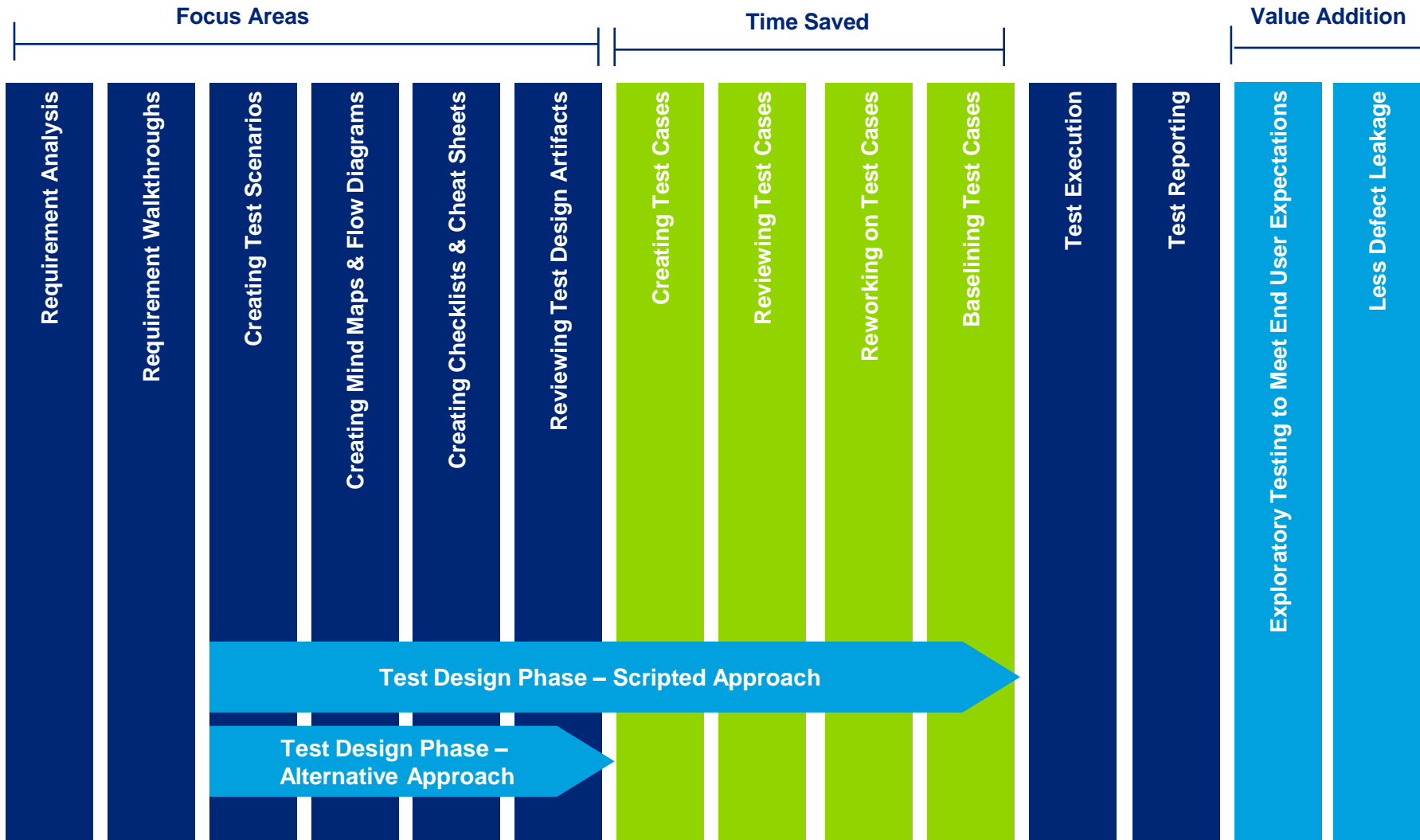
Do it yourself and conclusion



Do it yourself and conclusion

Conclusion:

Lightweight test design saves considerable test design time as well as reduces defect leakage.



Do it yourself and conclusion

Some Challenges during implementation of Exploratory testing using Alternative test design techniques:

Some of the challenges during implementation were related to management, team members and efficiency of execution. We have described below those challenges and their possible solutions..

Leadership & Management	<ul style="list-style-type: none">■ You need to convince your project stakeholders by showcasing the benefits of this approach■ A pilot should be conducted prior to implementation in large scale projects■ Design Test Strategy and Test Plan to suite the context of Scenario Based approach
Test Team	<ul style="list-style-type: none">■ Change in mindset of the resource is required as they would be transmitting to high level execution from an approach which was having detailed steps■ Team members need to sharpen exploratory testing skills by regular practice
Effectiveness & Efficiency	<ul style="list-style-type: none">■ During initial phase effectiveness may be bit less but it would increase drastically once the process is matured■ The lessons learnt and good practices can be leveraged to other projects of the organization for efficient execution of the later projects

Do it yourself:

We thank you for spending your time in reading the benefits of exploratory testing. We highly encourage you to pilot it in your organization / project and see the benefits for yourself. A few things to keep in mind which will help your organization become more successful.

Ensure that the stakeholders are well informed and know the pros and cons of the effort:



References:

References:

The Black Swan by [Nassim Nicholas Taleb](#), Random House, 2007

An Introduction to Scenario Testing by [Cem Kaner](#)

Scenarios: The Art of Strategic Conversation by Kees van der Heijden, Royal Dutch/Shell's former head of scenario planning, Wiley, 1996

Value Of Checklists by [Cem Kaner](#), CAST 2008 presentation

A tutorial in Exploratory Testing by Cem Kaner

The Nature of Exploratory Testing by Cem Kaner