
HEURISTIC RISK-BASED TEST AUTOMATION

A HYBRID APPROACH

www.happiestminds.com

CONTENT

- Problem: Current Pressures of functional/feature testing 3
- Solution: Heuristic Risk based Automation 3
- What is Risk-based testing? 4
- Types of Risks: 4
 - Business or Operational 4
 - Technical 4
 - External 4
 - Non-functional web Related..... 4
- How to calculate risk? 5
- Is Risk-based testing sufficient? 5
- Need for Automation 6
- Effective Risk based CI automation runs 6
- Case Study: (Mobile – Web application feature) 7
- Return on Investment 9
- Summary: 10
- References: 10

Problem

Current Pressures of Functional/Feature Testing



For any complex system there are a vast number of tests



There is infinite cloud of possible tests



Being in Agile World, shorter release cycles are need of the hour



Selective and efficient test coverage is inherent requirement



Continuous Integration - automating the whole process



Solution

Heuristic Risk Based Automation

Risk-based testing helps us meet all the above by prioritizing and reducing scope which helps tester to find bugs early in the cycle and also manage testing in a smaller duration of time. Automating the whole process of prioritizing optimizes it further.

WHAT IS RISK-BASED TESTING?

Risk-based testing (RBT) is a type of [software testing](#) that prioritizes the tests of features/functions/test cases based on the risk of their failure, it is

- A testing strategy to determine what to test, when, and how much, based on risk
- An opportunity to negotiate what “quality” means, and the acceptable level of quality for a project
- A way to confirm that risk remaining after testing is visible and acceptable to the business

TYPES OF RISKS



BUSINESS OR OPERATIONAL

- High use of a subsystem, function or feature
- Criticality of a subsystem, function or feature, including the cost of failure



TECHNICAL

- Geographic distribution of development team
- Complexity of a subsystem or function



EXTERNAL

- Sponsor or executive preference
- Regulatory requirements

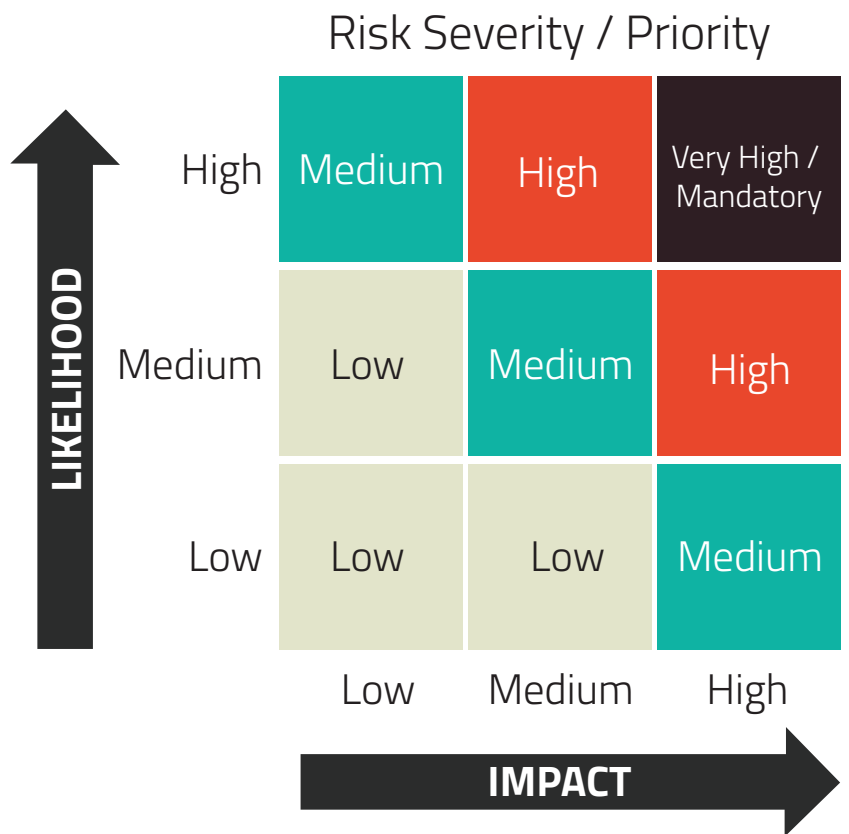


NON-FUNCTIONAL WEB RELATED

- Static content defects
- Web page integration defects
- Service (Availability and Performance) related failure
- Usability and Accessibility-related failure
- [Security vulnerability](#)
- Large Scale Integration failure

HOW TO CALCULATE RISK?

- Identify heuristics for exploring software risk.
- Risk analysis using inside-out approach.
- Inside-out is a direct form of risk analysis. It asks "What risks are associated with this thing?"
- Identify cost and probability factors based on this approach.
- Use heuristics to assign weights [cost (weight) and probability (weight)] to the factors.
- Calculate risk using these factors using a generic formula.



Use Case Risk

= Impact of Occurrence * Likelihood of Occurrence

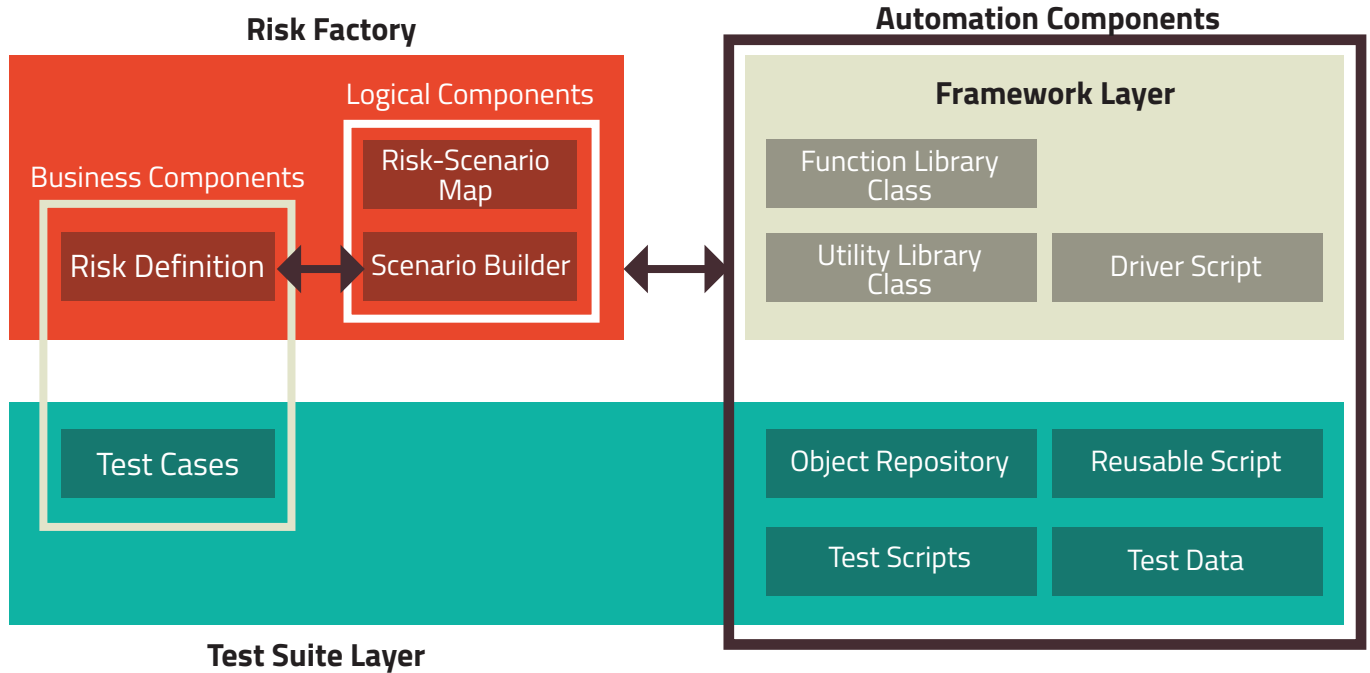
= Average cost ($\sum (\text{cost (weight)} * \text{cost (value)}) / \text{Number of cost factors}$) * Probability ($\sum (\text{defect generator (weight)} * \text{defect generator (value)}) / \sum (\text{defect generator (weight)} * \text{defect generator (max value)})$)

Test Case Risk = Use Case Risk * Test case Importance

- **Risk Calculation can be tedious and time consuming.**
- **Also with continuous Integration being a mandate process in this agile world, regular runs are need of the time.**
- **This brings the need to automate the Risk calculation process.**

IS RISK-BASED
TESTING
SUFFICIENT?

NEED FOR AUTOMATION



EFFECTIVE RISK BASED CI AUTOMATION RUNS

	Priority	Execution Frequency	Criteria	Automation Plans	
RISK ↑	P0	With every deployment	Acceptance criteria for the build	Smoke	FREQUENCY ↑
	P1	With every deployment	Only when P0 cases pass	High Sanity	
	P2	Alternate deployment	Deployment with major changes	Sanity	
	P3	Start and end of sprint	Sign off criteria for the Testing Cycle	Regression	

CASE STUDY

(MOBILE – WEB APPLICATION FEATURE)

Steps to calculate Risk Exposure for Use Case:

- Identify cost factors and assign weights to each of them. (Range 1-3) For e.g. Visibility - 2
- Add weights for each use case aligning to each cost factor.
- Calculate Average Cost for each Use case.

Average cost (Σ (cost (weight)*cost (value)) / Number of cost factors)						
Use Case ID	Cost factors	Maintenance	Visibility	Reputation	Total Cost	Average Cost
	Weight	2	2	3		
Web_TS_01	Validate introduction menu on promotion top page	3	3	4	24 (2*3+2*3+3*4)	8 (24/3)
Web_TS_02	Validate downloading of application from promotion top page	3	5	5	31 (2*3+2*5+3*5)	10.33 (31/3)

- Identify probability factors and assign weights to each of them. (Range 1-3) For e.g. Complexity - 3
- Add weights for each use case aligning to each probability factor.
- Calculate Probability for each Use case.

Use Case ID	Defect Generators	Complexity	Changes	New Technology	Total Sum	Average Cost
	Weight	3	1	2		
Web_TS_01	Validate introduction menu on promotion top page	3	3	4	20 (3*3+1*3+2*4)	0.83 (20/24)
Web_TS_02	Validate downloading of application from promotion top page	4	4	4	24 (3*4+1*4+2*4)	1 (24/24)

- Calculate Risk exposure

Risk Exposure Average cost * Probability				
Use Case ID	Use Case	Average Cost	Probability	Risk Exposure
Web_TS_01	Validate introduction menu on promotion top page	8 (24/3)	0.83 (20/24)	6.64
Web_TS_02	Validate downloading of application from promotion top page	10.33 (31/3)	1 (24/24)	10.33

CASE STUDY

(MOBILE – WEB APPLICATION FEATURE)

Steps to calculate Risk Exposure for Test Case:

To calculate test case risk:

- Define Importance and assign weight factor to each of them.
- High – 1.2
- Medium – 1.0
- Low – 0.8
- Assign Importance (High, Medium, and Low) to each test case.
- Calculate Test Case Risk.

Test case Risk =

Risk Exposure of the use case to which the test case belongs * Importance weight of the test case.

Test case id (Use case id)	Risk Exposure	Importance	Test Case Risk
TC1 (Web_TS_01)	6.64	High (1.2)	7.97
TC2 (Web_TS_01)	6.64	Medium (1.0)	6.64
TC3(Web_TS_02)	10.33	Medium (1.0)	10.33
TC4 (Web_TS_02)	10.33	Low (0.8)	8.26

Observation

Hence TC1 has high risk than TC4 even if the use case it belongs to has lesser risk exposure.

Note: The values given for the importance depend on what we want to prioritize; the use case risk or the importance for the test case. Currently low is given 0.8, medium is given 1.0 and high is given 1.2, this will adjust the risk exposure for the use case with 20 %

Automate the Test Case

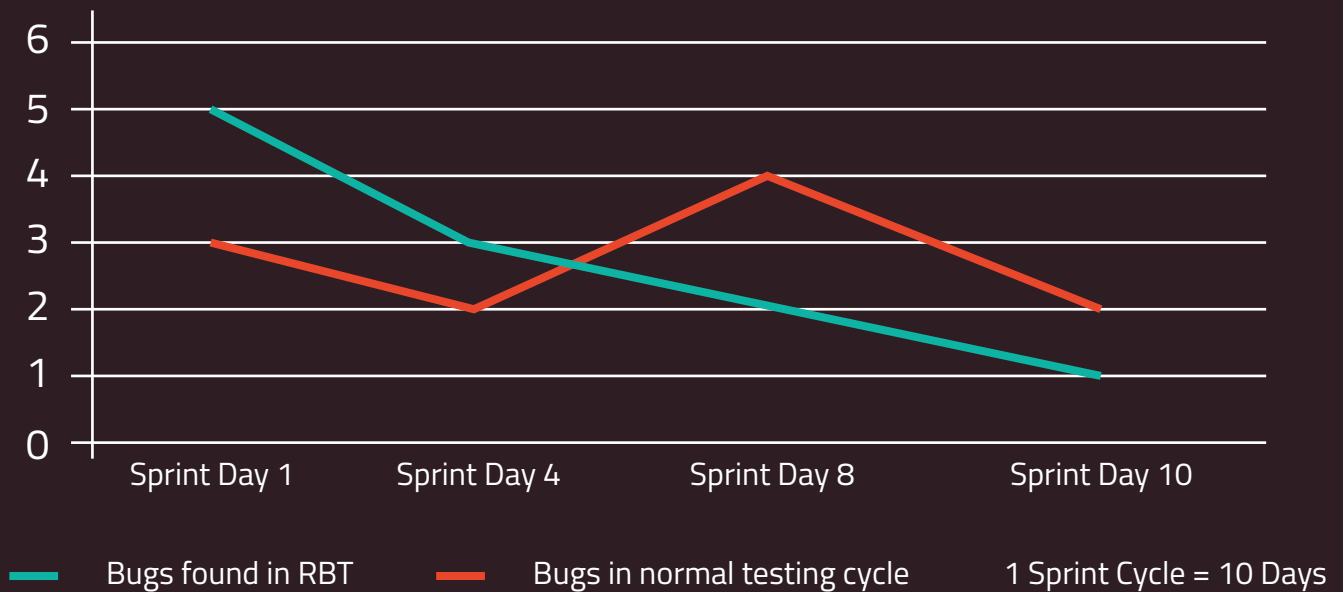
Use Case Id (Total Test cases)	Smoke (5%)*	High-Sanity (15%)*	Sanity (30%)*	Regression (100%)*
TC1 (Web_TS_01)	1	2	5	12
TC2 (Web_TS_01)	0	1	3	9

*rounded % of automatable test cases

RETURN ON INVESTMENT

A properly implemented RBT methodology helps assure that high-risk areas are tested first, then medium-risk and, finally, low-risk areas.

Bugs found Trend



Dividing the test cycle to 4 quarters, the % of bugs found in each priority is as follows.

- In a sprint, the % of bugs found in each priority is as follows
- 100% coverage of high risk features in the 1st sprint, and daily testing of the same further using automation with minimal effort, to keep an eye on the breakage.
- In case of patch release as there is no time for exhaustive testing, Prioritize Execution of Test Scenarios based on Risk like Smoke, High-Sanity, Sanity and Regression if time permits. This diminishes the probability of immediate production issues by 80% (as per Pareto's 80-20 rule, 80% of usage on 20% of the app)

As Risk-based testing is

- A Systematic approach
- Accommodates time constraints

Hereby ROI is

- High priority bugs can be found early in the cycle.
- Reduced effort, cost, duration
- Enables sizing remaining risk at any point

To be noted: It may leave undetected defects; hence acceptable risk related criteria need to be agreed upon from stake holders.

About the Author



Shabnam Choudhury

Test Architect
Product and Engineering Services
Happiest Minds

She is an expert in Test Automation of GUI and Web Services, along with Performance and Security Testing with robust Test Management skills. She has Extensive Exposure in domains such as Retailing and Auction, SPM, ERP, Healthcare, Logistics and Technology, Small Business and Advertisement Platform domains.

She started her career in software testing in the year 2000. She believes, "to be a tester is easy, but to be a good tester can be very challenging". Hence in her strive to be a good tester she keeps exploring different ways to make testing efficient while delivering quality products. This has led her to discover the risk-based testing strategy which she has studied in-depth and has implemented a complete solution around it.

She has a variant industry experience working for multinational service providers like Mphasis-HP and Mindtree, and fortune 100 MNC like Yahoo! Currently while pursuing her EGMP program at IIM-B, realizes that the macro organizational behavior of IT services industry has turned the corner and needs to focus primarily on customer business problems and holistic, end-to-end IT solutions addressing them rather than products, feature sets, individual technology or expertise based offerings.

Summary

There are several ways to do Risk-Based Testing, from using risk to find the test, to using the risks as a way to prioritize what areas to focus during testing. The choice was a method to prioritize use cases and eventually in a risk analysis. Test cases are designed from the most risky use cases and resources for testing are estimated. This method was chosen because it seemed to be a good way to prioritize the tests

References

- http://en.wikipedia.org/wiki/Talk:Risk-based_testing
- www.stickyminds.com/sitewide.asp?Function=edetail&Object
- <http://www.methodsandtools.com/archive/archive.php?id=31>

www.happiestminds.com

About Happiest Minds Technologies:

Happiest Minds, the Mindful IT Company, applies agile methodologies to enable digital transformation for enterprises and technology providers by delivering seamless customer experience, business efficiency and actionable insights. We leverage a spectrum of disruptive technologies such as: Big Data Analytics, AI & Cognitive Computing, Internet of Things, Cloud, Security, SDN-NFV, RPA, Blockchain, etc. Positioned as "Born Digital . Born Agile", our capabilities spans across product engineering, digital business solutions, infrastructure management and security services. We deliver these services across industry sectors such as retail, consumer packaged goods, edutech, e-commerce, banking, insurance, hi-tech, engineering R&D, manufacturing, automotive and travel/transportation/hospitality.

Headquartered in Bangalore, India; Happiest Minds has operations in USA, UK, The Netherlands, Australia and Middle East.

To know more about our offerings. Please write to us at business@happiestminds.com