6 Hats Thinking for Comprehensive Test Case

M.Rabindranath
Sr. Software Test Professional
Worked for: Microsoft India, Wipro Technologies and Ivy Comptech

Preface:
Designing test cases is both “Art” and “Science”, it is Art because it requires lot of creativity, intuition and contextual understanding and it is science because to create good test cases it is important to understand the facts and introspect on the findings and realities of implementation.

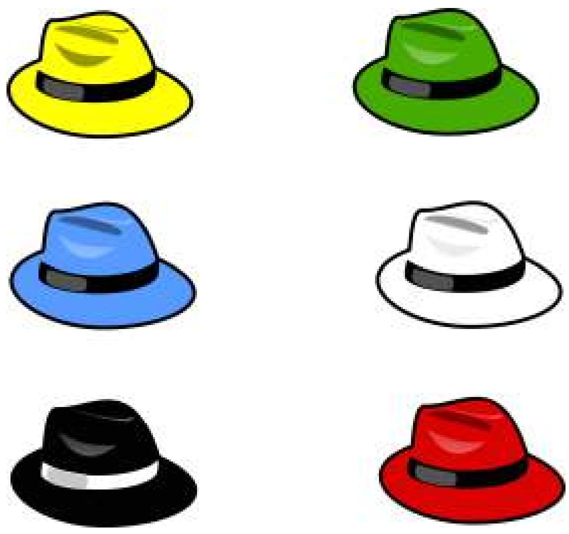
Some of the data shows that at an average 8-10% of the bugs slip into UAT or Production across the software industry from test phase, and the obvious impact because of this is dent on the credibility, high cost incurred to fix bugs in UAT or Production.

Though the goal and ambition is to reach the Nirvana state which is “Zero” bugs in UAT or Production is it really attainable, and the answer is “yes” (at least most of the times) please look at what I just said closely, I did not say Zero bugs in Software code, I just said Zero bugs in UAT or Production, there is a very delicate difference, lot of logic and combination of conditions in the Software code might never be used in Production or UAT by the Users or UAT testers (as UAT testing more concentrates on user scenarios).

Bug removal efficiency which is directly proportional to the comprehensiveness of the test case design is always challenged with 3 factors:
(1) Trade off that tester has to take between cost/time Vs quality of testing
(2) What is and How much is comprehensive testing is always subjective and varies from situation to situation
(3) Limitations in metrics/mechanisms to track comprehensiveness of test cases.

All this is driving to a point of building a mechanism for comprehensive test cases both qualitatively & quantitatively and tracking to help decide what to test and how much to test with minimal efforts.

Overview of the Concept:
“6 Thinking Hats for comprehensive test cases” mechanism is devised to alleviate and address the above points, the concept is evolved out of the famous Problem solving technique called de Bono Hats System or Six Thinking Hats, in which strategies for issues can be evolved in a structured way by group of individuals or a single individual by disguising in Six different characters in different hats, and each hat has a colour which symbolizes each character.



White Hat is for Facts & Information character.

Red Hat is for feelings and Emotions character.

Black Hat is for Critical Judgment character.

Yellow Hat is for Positive character.

Green Hat is for new ideas and

Blue Hat is for the big picture.

Team or Individual will analyze problems donning these hats/characters and come up with strategy/solution to the problem.

Mechanics of the Concept:

Taking the concept further to Test case design, team or individual will don each hat of the six hats at any point of time and build test scenarios from the requirements/functional Specs/User Scenarios. If it is team each member of the team will play the role of a character, in Six Thinking Hats, at a given point of time to create test scenarios before creating detail test cases.

Member with Yellow Hat in Six Hats thinking will create Positive flow test scenarios, Black Hat will create pessimistic test scenarios, Red Hat will create Usability, User based test scenarios, Green Hat will create innovative test scenarios, and White Hat will create technical / architectural test scenarios and Blue Hat for Integration test cases. Test cases would be designed from the above created scenarios the test cases will evolve or will be designed.

If it is individual at any point of time the individual will don one character and create test Scenarios, from which subsequently test cases will be designed. The test scenarios thus created are mapped in a Traceability Matrix which is designed to address the "6 Thinking Hats for comprehensive test cases" concept. Sample shared for reference is created manually by SQA team and would be reviewed by Pgm/Dev.

It is very apt for agile methodologies because teams from Dev, PM and Test in these methodologies work closely together and as each role comes with strong attributes of characters in 6 Thinking Hats, this can very well contribute to the design of test scenarios.

Dev being pessimistic can don Yellow Hat thinking, PM being more user conscious can wear Red Hat for creating usability and user based test scenarios and test(er) team being pessimistic can wear black hat thinking creating negative test scenarios, other hats White, Blue and Green for technical, Integration and innovative test Scenarios can be donned by any one from Dev, PM or Test. This is just indicative roles only, it can be interchanged as required as the bigger goal here is to bring systematic and structural way of building test scenarios.

Advantages of the concept in Project life:

- Test functional coverage is no more subjective, for each requirement/functionality you exactly know how much coverage is obtained.
- As the test scenarios are fragmented based on the user, business, technical aspects, it is lot more easy to make decisions on when and how much to test.
- This is Agonistic of the project execution model.
- The advantage of this is that very comprehensive and traceable test scenarios are created.

Next Steps to the concept:

Taking this concept further to make it simpler to use by the teams, tools should be built around the concept to track the test coverage based on the test scenarios built under each characteristic of 6-Thinking Hats and build reports on the metrics of the test coverage.

Closing words:

"6 Thinking Hats for Comprehensive test cases" is an effective tool which can be adopted by any team for any type of project. Expected additional efforts over the existing efforts will be 8-10% till the teams get accustomed to the method.

The effectiveness of testing is directly proportional to the efficiency of test case design, and this can be achieved by reducing ambiguity on comprehensiveness, clarity on decision to take appropriate tradeoffs between time Vs Quality of testing, mechanism to track the comprehensiveness, which can be systematically achieved by "6-Thinking Hats for comprehensive test cases" .