A Realistic Approach: RTST to Reduce Cost & Time

Neeraj Kumar[1]
Prof. (Dr.) Ajay Rana[2]

[1]Research Scholar, Singhania University and
Prof. & Head – IT, Harlal Institute of Management & Technology, Greater Noida (India)

[2]Professor, AMITY University, NOIDA(India)

Abstract:

Regression testing is the most expensive technique, but this technique gives confidence that whatever changes has been incorporated they are not making any adverse affect to the output. In Regression Test Selection Technique (RTST), we select test case from existing test set and perform test to validate modify the software. In this framework, it has been tried to reduce costs by selecting and running only a subset of the test cases in a program's existing test suite

Keywords: *Regression Test Selection Technique (RTST); Risk Based Testing (RBT); Iterative Technique (IT); Pairwise Testing (PWT); SDLC*

I. Introduction

Software development process becomes more complex as the number of requirement of different user increases. This happens as we are not following any scientific & systematic approach for the development process. Due to absence of this approach software development process become more costly and risky. A systematic & scientific approach is required for the analysis, designing, implementation, testing, and maintenance of process. This approach is defined as Software Engineering. Emphasis of Software Engineering is on quality development process. With implication of Software engineering approach software process becomes more reliable, robust & maintainable.

In Software development process we maintain the Software Development Life Cycle (SDLC). The objective of SDLC is to ensure high quality of product delivery, maximum utilization of resources and better control & monitoring over projects. The activities feasibility analysis, requirement analysis & specifications, design, coding and testing, implementation, and post implementation takes place under SDLC. In feasibility analysis objective is to find out that financially and technical viability of software or project. The objective of requirement analysis is to collect all related information from user or organization about the system that has to be developed with a clear objective to understand the complete requirement about the system without any ambiguity. To do this requirement analysis a systematic organized document is prepared which is known as Software Requirement Specification(SRS) is prepared, in which System Requirement (SR) i.e., complete system requirement is generated as per user / organization needs and in System Specifications (SS) i.e., complete system specification is generated as per the mentioned in System Requirement and further in design activity objective is transform requirement specifications as mentioned in SRS to in to some structure design. In structure design mainly two activities takes place i.e., architectural design & detailed design. In architectural design mainly we represent modules and their relationship between the modules and in detailed design data structure and working algorithms are defined. Objective of coding and testing is to transform system design into computer understandable format i.e., in to machine language and of testing objective is to find the bugs not to rectify them, implementations objective to ensure satisfactory installation of system, and finally post implementation, remedy for small problem persists.

Testing [18] is formal procedure. Testing is used to ensure bug free system. Testing is important in every stage of software development life cycle, but the testing methodology that we use at different level of software development is different in nature and has different objectives. Typically three distinct type of testing is available for software process, i.e., Unit / Component testing, Integration testing & System testing. Testing model is represented in table 1.

Table 1 TESTING MODEL [18]

Types of Testing	**Structural Testing**	**Functional Testing**
User's Perspective	User has complete view of the code and write test conditions based upon the code.	User has no idea about the code and writes test conditions based on possible Input and outputs to check functionality as mention in requirement specifications.
If Test condition Fails? Remedy?	Faulty outcome. Yes – the test condition helps to find the fault in provided code.	Output Failure No – it is very difficult to find the reason of the failure.

Regression Testing is methodology in which testing is performed on modified program to verify that the changes are correct and to ensure that the changes has not adversely affected the position of program. In this technique, already tested suites will be available for reuse and also retest all approaches, but due to this technique it consumes lot of time and available resources. To reduce the cycle time we use Regression Test Selection Technique (RTST) techniques, in this to retest modified program a subset of test suite is selected from already existed test suite. Now a day there are lots of reduction techniques available which has been successfully applied across programs. Such type of techniques has been developed to reduce the time spent in performing test execution for identified system functionalities.

Regression Test Selection Techniques with association of Risk Based Testing (RBT), Iterative Testing (IT) & Pair-wise testing (PWT) to reduce the test time. The objective of RTST with new frame work is to reduce cost & time with complete test coverage with minimum utilization of resources.

II. Test Selection Techniques

In this section we will study about the Regression Test Selection Techniques (RTST) along with other research has been done for cost effective output with minimum time. Also in the coming section we learn about the techniques like; Risk Based Testing (RBT), Iterative Testing (IT) & Pair-wise testing (PWT) and compare these techniques. We know that no single technique is capable for to reduce the test time. The objective of RTST with new frame work is to reduce cost & time with complete test coverage with minimum utilization of resources.

A. Regression Test Selection Technique (RTST)

In this approach only a subset of the test case contained in test suite is selected and retested. While selecting the optimal test case is not possible [5], is the cost benefit tradeoff of RTST is the main concern of regression testing. However empirical studies revealed several cost effective techniques, but result may vary on different parameter viz., programs, test suite [6], they affect the performance of RTST. Earlier RTST is one time activity, but now days it is continuous process. Now it is possible that RTST can perform less cost effectively. This happens due to selecting larger test suites; in returns these suites will result less cost-effective output.

Number of research has been done on RTST; however few of investigation we represent that has been done in the research techniques are briefly described in table 2 given below:

Table 2

S. No.	Technique	Description
1.	**Scaling**	In this technique, combine the effectiveness of RTST with the efficiency of a technique that work at a higher-level of abstraction, is precise but may be inefficient on large systems [3, 6, 8] and may, thus, be imprecise [19]. In this approach, an initial high-level analysis is done to identify parts of the system that has to be further analyzed and an in-depth analysis of the identified parts, which selects the test cases to retest.
2.	**Slicing**	In this technique[17], unlike other techniques, neither data flow history nor recompilation of data flow for the entire program is required to detect affected part of program, only partial program is need to be recompiled.
3.	**Safe**	In this techniques [4] select a test case under some specified conditions, in every test case that covers changed program entities. Such technique selects every test case that exercises at least one statement that was added to or modified in creating or that has been deleted.
4.	**Minimization**	In this technique [14] objective is to select a minimal set of test cases that covers all amended or affected outputs of program. In this technique randomly selects test cases every statement modified in generating output is checked at least once.
5.	**Dataflow Techniques**	In this techniques [1, 16, 21], only modified data is checked which has been affected by transformation from selected test cases.
6.	**Firewall Technique**	In this technique [10, 11, 12], a firewall is placed surrounding the modified code.

Apart from these techniques, costing [13] is one of the important factors affecting RTST. Costing involves direct and indirect overheads like test selection, execution, analysis & management. Costing plays important role in Regression Test Selection Technique. The impact of investigation [9] on test suite in which test cases are grouped into test suites on RTST is measured against metrics time, cost, and fault detection effectiveness.

It reveals from the studies that no single technique is efficient for effective results. In this research paper our emphasis is on reduced execution time, we will develop framework for RTST, how we can reduce cycle time with the help of techniques like Risk Based Testing, Iterative Testing and Pairwise Testing.

B. RISK BASED TESTING (RBT)

A risk analysis is performed to investigate the highest risk in terms of probability and cost. A risk based approach is introduced for testing. A well defined flexible approach is required to explore this.

Risk Based technique [20] is a testing techniques which is widely used in where the risk involvement is higher. The technique focuses on analyzing and understanding the business risk and measurement of critical functions in the application. With this technique we are able to identify and analyze the risk prone areas. It is more perceptive than calculative. A risk based technique is significant if it can determine and ensure the quality of product quantitatively using metrics.

The testing is generally based on the ability and judgment of tester's to find the defects rather than what technique he has applied. Objective of Risk Based Testing is to minimize the risk associated with program or project. This makes it crucial to apply risk based technique while doing testing. The main objective of Risk Analysis is to identify potential problems that could affect the cost or outcome of the project.

How to assess risk and its probability? To make assessment of risk following questions can arise:

a) Is this a risk?

b) Nature of risk?
c) What are the out comes?
d) What are after maths of risk happening?

Final decision will be based on outcome of risk assess. From the risk identification and analysis it will become more feasible to prioritize the each component of system on the basis of risk. Once the risk has been identified, risk avoidance plan is generated to minimize or to reduce the risk.
Method of assessment of Risk given in Risk Analysis Model given in fig 1.0

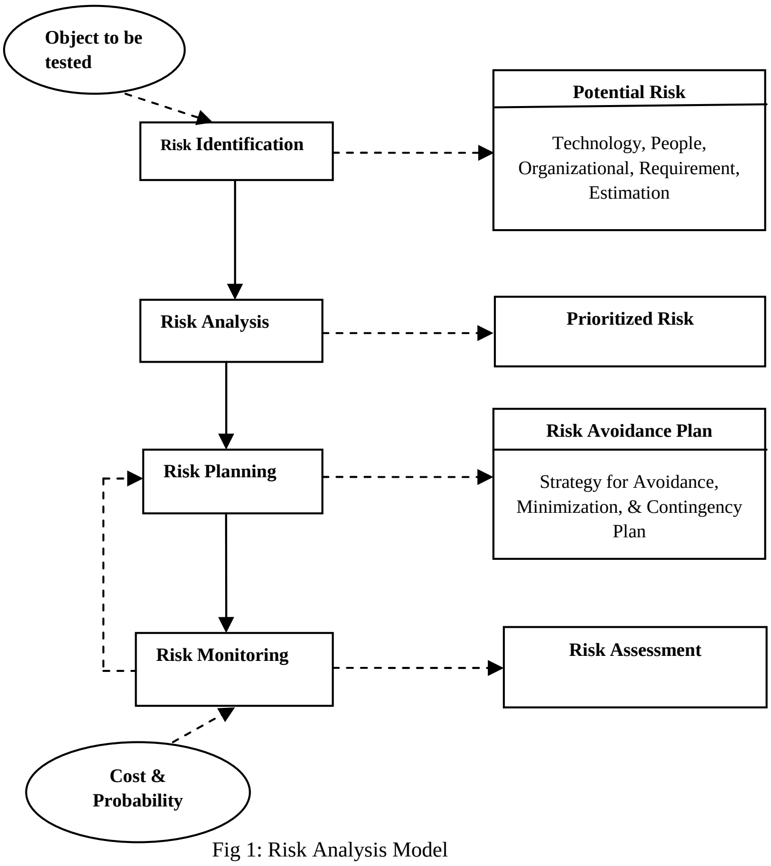

Fig 1: Risk Analysis Model

Limitations:
There are certain limitations of this technique.

- Precisely identifying the business risks is difficult, requires clear understanding of applications and detailed knowledge of requirements
- RBT focuses on the high risks and in general ignores the low impact areas.
- The technique is more practical & functionalities are identified based on tester’s judgment.
- The technique doesn’t assure that all the possible interactions between impacting and non-impacting parameters are covered.

In risk based testing, an assessment criterion is based on following parameters-

- Business Impact
- Probability of Failure
- Level of Control in restoration
- Confidence Level of tester

If tester has higher confidence than the right things will happen at the right time and risk based testing will provide fruitful results.
Finally the Risk outcome will be multiple of Business Impact, Probability or Estimated risk

C. ITERATIVE TESTING (IT)

An Iterative testing is a technique that involves planning and executing the test cases in cycles. A cycle may simply be defined as test iteration. In every iteration has a defined timeline and the test cases are distributed over iterations based on business impact, priority or test coverage.

The technique used to test the application in entirety covering all the functions, logics and interfaces. This is a time consuming affair as the testing proved to be never ending task and spanned over months to years. The new technique required to testing application to reduce time without compromising on the quality. In this techniques objective is to split different functionalities in multiple cycles. In this, a function is to be tested were selected in each cycle based on business impact. The technique ensures that the product that span over release gets adequately tested in cycles to meet the quality compliance.

The iterative testing is not so commonly known technique; however, it has advantages to use in conjunction with other techniques.

This technique is applied when there is enough timeframe for testing, e.g., in a quarterly release if testing spans for a month, then the cases can be distributed over 3 or more cycles.

Benefits of Iterative Testing

This is specifically helpful in regressions when cycles are repeated in every output keeping in view transforming the high risk function to low risk. It helps to define logical milestone for testing. It also fixes fault tested in subsequent cycles. This is helpful in Incremental Integration techniques.

Limitation of Iterative Testing

The technique is never favored when individual test cycles are too long. Many cycles delays the process.

D. PAIRWISE TESTING (PWT)

Testing exhaustive combinations is not possible always and hence PWT is used for reduction of test sets. Pairwise testing is a technique in which all possible combinations of variable are tested. In this technique set of test cases are prepared that cover-up all combinations of selected test data value for each combination. Pairwise testing basically depends on individual values selected while domain partitioning. The selected values are again permuted to get all the possible pairs for testing.

The orthogonal array [15] is used along with greedy algorithm [2]. It's a powerful technique to meet up all the requirement of Pairwise Testing. For e.g. consider a function having two input variable I_1 and I_2, let's assume F as a set of test data and say for each input variable $F(I_1) = \{1,2,3\}$; $F(I_2) = \{A, B,C\}$, then the possible number of test solution can be 3 x 3 = 6 (six) possible test solutions. With help of greedy algorithm we select the best optimal solutions, in this way we left with minimal set of solutions. From table 3 it is clear that half of the solution is left for further testing and also every variable is represented at least once.

Table 3

Test Case	**I_1**	**I_2**
T1	1	A
T2	2	B
T3	1	C

The PWT is popular as it produces small test sets that are relatively easier to manage. In general with the application of Pairwise Testing test suite become smaller and efficiency will improve.

Limitation:

It fails when we don't select right values for test case.
It fails when we don't have knowledge how variables are interacting each other.
It fails when we don't have good perspective.

III. Comparison of RBT, IT and PWT

Table 4

S. No.	Technique	Benefits	Limitations
1.	**RBT**	RBT is well suitable for small and medium sized projects where all the exact impact can be easily calculated.	In complex situations, where there are many logical or conditional statements are there, the amount of time to calculate the risk and cost associated with it. it is not feasible to calculate the exact impact and costing involve. In such type of situation RBT is costlier.
2.	**IT**	IT is suitable in regressions when cycles are repeated in every release. Objective is reducing the high-risk functions to low-risk. IT is suitable in incremental Integration techniques.	Not suitable when test cycle are long. Number of repeated cycles delays the process.
3.	**PWT**	Complex situation, where test situation is complex, i.e., interaction level > 4, in such situation, the exhaustive test case count will turn out to be very high and it will take lot of time to calculate.	PWT applicable only when level is upto 2 or 3. When exact level of interaction is not defined PWT fails.

Each technique has its merits & demerit. Comparing all three testing techniques, it is very difficult to analyze the performance of test suit. Now the framework we design in a way that the technique applied in such test situations to control the number of cycles with ensuring high quality. By using, these techniques we will ensure that most of the defect will be detected and removed at earlier stage if used with iterative testing.

This technique will help us to reduce the limitation of Pairwise test technique. The PWT ensures the effective test reduction at 2 and 3-level interactions. However, there is always a probability of defects at higher order interactions. These techniques can make effective use of other techniques to identify the critical parameters or combinations that have high possibility of defects or impose high risk to the business. By ensuring that all multi-level combinations are adequately covered, eliminate the risk of low coverage or not testing unidentified combinations.

In our framework we will combine and customize individual techniques to draw advantages of each of the individual techniques in RTST. Possible combination can be represented in table 5:

Table 5

S. No.	Test Case	Combination
1	1	RBT & IT
2	2	RBT & PWT
3	3	PWT & IT

Which combination will be suitable to which problem, it depends on the nature of the problem. Before implementation we should identify the facts related to problem – complexity, priority, business impact, business risk, and goal. After combining these facts deciding the case scenario which combination of technique is applicable. The following steps will be required to implement these techniques in RTST:

a) Identify impact of cost, time & quality.

b) Identify cycle level
c) Identify parameter
d) Identify priority with help of RBT
e) Identify exhaustive parameter set and segregate impacting and non-impacting parameter
f) Re – arrange impacting parameter
g) Apply PWT to non impacting parameter, resulting in reduced set.
h) Assign priority to reduced set accordingly.
i) Then apply IT on the basis of priority.

First, we will identify the parameters & cycle level, and then use RBT to identify impacting and non-impacting parameter, while using PWT resulting in reduced test cases. Reduced test cases will be distributed on multiple cycles to use Iterative Technique, resulting output will be exhaustive and almost risk free. Finally, the test set available for Regression Testing to validate the modified software is very concise and hence take less time & cost.

IV. Conclusion and Future Work

We have presented a framework for RTST for reduction in time & cost with respect to software or project. We have represented that with application of our framework is more efficient and effective to evaluate RTST. From our framework it reflects that the techniques that we have used having different objective, but the collective efforts are comparable. Our framework can be use to compare other techniques available for RTST. In our framework technologies that we have used are having objective different, but collectively output is fruitful. From the perspective of testing we know that cost and time is critical factor, our framework is more concern over these two factors.

Further in future work we represent the practical application of our frame work. We represent practically how our frame work is more effective and efficient and how it helps to reduce cost & time for any software or project.

References:

[1] A.B. Taha, S. M. Thebaut, and S. S. Liu.” An approach to software fault localization and revalidation based on incremental data flow analysis”. In Proceedings of the 13th Annual International Conference on Computer Software and Applications, pages 527–534, Sept. 1989.

[2] D. M. Cohen, S. R. Dalal, M. L. Fredman, and G. C. Patton, "The AETG System: An Approach to Testing Based on Combinatorial Design," *IEEE Transactions on Software Engineering,* vol. 23, no. 7, pp. 437-444, 1997.

[3] G. Rothermel and M. J. Harrold. A safe, efficient regression test selection technique. ACM TOSEM, 6(2):173–210, Apr. 1997.

[4] G. Rothermel and M. J. Harrold. A safe, efficient regression test selection technique. ACM Transactions on Software Engineering and Methodology, 6(2):173-210, Apr. 1997.

[5] G. Rothermel and M. J. Harrold. Analyzing regression test selection techniques. IEEE Transactions on Software Engineering, 22(8):529-551, Aug. 1996.

[6] G. Rothermel and M. J. Harrold. Empirical studies of a safe regression test selection technique. IEEE Transactions on Software Engineering, 25(6), pages 401-419, June 1998.

[7] G. Rothermel and M.J.Harrold. Empirical studies of a safe regression test selection technique. IEEE TSE, pages 401–419, Jun. 1998.

[8] G. Rothermel, M. J. Harrold, and J. Dedhia. Regression test selection for C++ software. Journal of Software Testing, Verification, and Reliability, pages 77–109, Jun. 2000.

[9] G. Rothermel, S. Elbaum, A. Malishevsky, P. Kallakuri, and B. Davia, The impact of test suite granularity on the cost-effectiveness of regression testing. In Proceedings of the in international Conference Software Engineering, May, 2002, pages 230 – 240.

[10] H. K. N. Leung and L. White. A study of Integration Testing and Software Regression at integration Level. In Proceedings of the Conference of Software Maintenance, pages 290–300, Nov. 1990.

[11] H. K. N. Leung and L. White. Insight into Regression testing. In Proceedings of the Conference of Software Maintenance, pages 60–69, Oct. 1989.

[12] H. K. N. Leung and L. White. Insight into testing and Regression testing global variables. Journals of Software Maintenance, 2: 209–222, Dec. 1990.

[13] H.K.N. Leung and L.J. White. A cost model to compare regression test strategies. In Proceedings of International Conference on Software Maintenance, pages 201-208. Oct. 1991.

[14] J. Hartmann and D. Robson. Techniques for selective revalidation. IEEE Software, 16(1):31-38, Jan. 1990.

[15] J. M. Harrel, "Orthogonal Array Testing Strategy (OATS) Technique," http://www.seilevel.com/OATS.html, 05/28/2004.

[16] M. J. Harrold and M. L. Soffa,. “An incremental approach to unit testing during Maintenance”. In Proceedings of the Conference on Software Maintenance, pages 362–367, Oct. 1988.

[17] M. Weiser, ‘‘Program slicing,’’ *IEEE Transactions on Software Engineering,* vol. SE-10, no. 4, pp. 352-357, July 1984.

[18] N. Kumar and A. Rana. A Road Map to Testing Techniques – Historical Perspect. GLOBUS: An International Journal of Management & IT, pages 25 – 30, Volume 2 No 1 Jul – Dec 2010.

[19] P. Hsia, X. Li, D. Kung, C.-T. Hsu, L. Li, Y. Toyoshima, and C. Chen. A technique for the selective revalidation of OO software. Software Maintenance: Research and Practice, 9:217–233, 1997.

[20] Stale Amland, Risk Based Testing and Metrics, 5th International Conference EuroSTAR '99, November 8 - 12, 1999, Barcelona, Spain.

[21] T. Ostrand and E. Weyuker. “Using dataflow analysis for regression testing”. In Proceedings of the Sixth Annual Pacific Northwest Conference on Software Quality, pages 233–247, Sept. 1988.