ENHANCING SOFTWARE QUALITY

INDIUM

RCA on Residual defects – Techniques for adaptive regression testing

Prepared by: Naresh Jayabalan & Irene Preetika

www.indiumsoft.com

Table of Contents

1 Abstract

One of the important issues associated with a system lifespan view that we have ignored in past years is the effects of enduring defects – defects that persist undetected – across several releases of a system. Many studies performed to date have evaluated regression testing techniques under the limited context such as short term assessment which do not fully account for the industry based solutions.

Reports estimate that regression testing consumes as much as 80% of the overall testing budget and can consume up to 50% of the cost of software maintenance.

Researchers have developed techniques to address the number of issues related to regression testing. In this paper we have concentrated on the three areas. First, techniques that can help to reuse the test suites created for one build and categorizing those test cases that need to be repeated for testing subsequent releases and the strategy for effective ordering for running the test cases.

Secondly techniques that will reduce the regression testing time by creating effective regression test suites that test the altered modules, by identifying test cases in the regression suite that are not needed to rerun on the new build and removing obsolete test cases.

Finally, techniques to recover test cases by identifying, manipulating, and transforming obsolete test cases, by writing new test cases from old ones, and by repairing/rewriting test cases when the software evolves.

2 Overview

Software products undergo several processes like design, development, testing etc. before they get introduced into the market. Even after all the phases, there are chances where one or more defects persist undetected or missed during the testing phase for a long time and gets leaked out to the customers. When these defects are encountered by the customers this ultimately becomes a failure. Such defects are termed Residual defect as one might know.

With the success of testing, measured by the number of such Residual defects, the only solution being regression, how does one go about doing it in the most effective way?

Root Cause Analysis is performed on those Residual Defects to find the actual cause of the failure, the area that was missed out, the area that requires more importance, reason for failure etc. After this investigation, the result / inference can feed in as the scope for regression testing. Adaptive regression techniques can then be applied to make testing and in turn the product successful.

3 Residual Defect

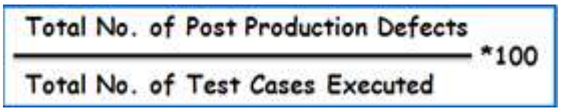
Residual defect is an important factor that decides the quality of a released product. In common, it is impossible to find all the defects within a period of time. Residual defects might affect the success of the software products based on each organization's business goals and the type of system. This will leave behind a very dissatisfied customer.

Post-release defect density of a software product cannot be measured until it has been put into production and used by the customers. This information becomes available too late to take corrective actions to software quality. Software defects correction is significantly more expensive when the defects are identified by the customer compared to defects correction in the earlier development process.

3.1 Residual Defect Density (RDD)

RDD is used to measure the Percentage of application defects found in production. It influences the quality of the end product to the customers. RDD gives the percentage of defects leaked to the end user.

Formula:



$$\frac{\text{Total No. of Post Production Defects}}{\text{Total No. of Test Cases Executed}} * 100$$

Total No of Post Production Defects	= 20
Total No of Test Cases Executed	= 250
As per the formula	= 20/250*100 = 8% (Ratio = 0.08)

Example:

In the above formula, "Total No of Post Production Defects" refers to all the defects found after deployment but missed during testing. Residual Defect Density is a "Lower the better" metric. It evidences the Effectiveness of Testing.

4 The need for RCA

Basic RCA (Root Cause Analysis) can often be extremely illuminating. For example, one might be able to arrive at reasons for defects and will be able to categorize them as Environment Issue, Insufficient Test Coverage, Data Issue, Oversight of Requirements, Inadequate Instruction, etc. It would also be advisable to map defects with their respective areas.

RCA, though a generic term may not have hard and fast rules of a specific template or standards. This can be customised in a way that will help the tester arrive at solid inferences /

solutions that will help in the next stages of defining your regression tests. This will in turn help reduce Residual defects that are leaked to customers.

Effective RCA documentation can also serve as one of the valuable test artefacts and can be bundled together as part of the test summary / results report.

5 Regression Testing

As discussed in earlier sections, the outcome of the RCA will serve as the basis for your regression suite. For example, if you find a specific area to be most bugs prone, you can pick the relevant test cases for regression when you form your suite.

Basically, there are two software regression testing strategies followed by organisations. The first strategy is complete regression where all previous test cases are executed again. The second strategy is partial regression where software changes are analyzed qualitatively and affected parts are retested. The merit of using the first strategy is that the affected parts are found as much as possible, and the demerit is that the organization cannot meet the expense spent on time and resources needed as it is very huge. The second strategy has the merit of less cost and the demerit is poor accuracy of qualitative analysis and the inadequacy of the test.

6 Proposed Process

Traditional methods may have its own advantages and disadvantages. A general approach towards Regression testing was only to focus on test cases. Normal testers may think that results will always be positive and less number of defects captured during this phase. Another perspective is that repeated test steps are executed and testers would even skip on such test steps.

In our exclusive 3R approach - Reuse, Reduce and Recover with its own techniques – we have tried to tweak the traditional methods to suit our practical needs.

- Assimilation of SME
- Mechanized Checks
- Innovative Tests

By following these approaches and techniques we were able to deliver a quality product with more focus on the application as a whole. Tester's perspective towards Regression testing has widely changed. In this paper, our results show that this is the best practice to reduce the residual defect rate.

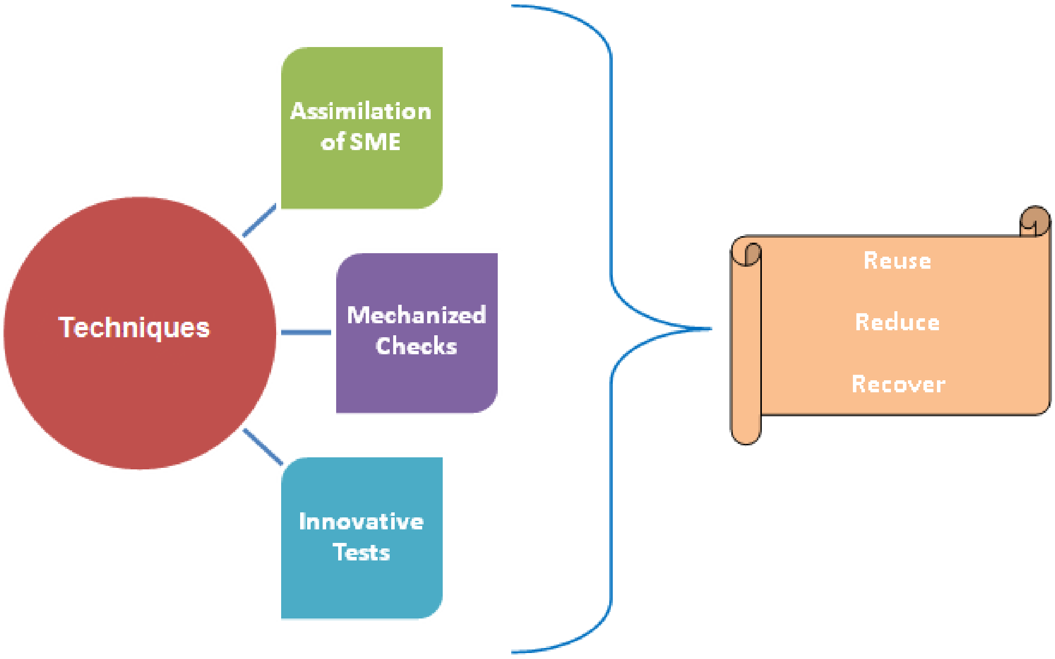


Fig 1: Proposed Techniques

6.1 Assimilation of SME

Performing Regression Testing on all modules of an application is a tedious process and lots of hours get wasted when many resources involve in the same modules. Thus we incorporated and assigned SME to each module of the application who will take good control of it. Each of them will be indulged deeply in monitoring the application.

The SME will be recording all the changes of the modules with time and maintain a report. This can be **recovered** on need basis. Any new feature introduced or changes made in the module is indentified and recorded instantly. The data collected is updated in a well-timed manner.

Each record collected from the SME is shared within the Team. As a result, each one of them is aware of the latest updates and changes in the modules of the application. Clarifications regarding the modules in specific or knowledge transfer can be obtained from them. These consolidated SME reports are valuable test evidences when converted into checklists.

By allotting each SME for a module, we **reduce** the time spent on regression testing and RCA of any new issue is identified easily by the SME. The data collected by the SME can be **reused** in future as well.

6.2 Mechanized Checks

Mechanized Checks is another approach that we adopted to save time by using the **reuse**, **reduce** and **recover** approaches. Rather than writing the same scripts again, it is always better to automate all the test cases so that it can be **reused** later from the maintained records. It is also easy to make changes when necessary by altering just the required section of the script instead of writing the entire scripts which saves valuable time and manpower by just **recovering** new scripts from the old ones.

Mechanized Checks approach when used effectively results in a better quality by using the 3R approach. It also **reduces** the cost spent on resources eventually.

Mechanized Checks has it big advantage when its reports are designed with care. The initial effort put into designing these reports will really pay off as it is going to be consistent and precise. What more test evidence than an execution report! These are the reports that the customer is finally going to be interested in.

6.3 Innovative Tests

It is virtually impossible to write test cases that cover every scenario that can occur. To uncover hidden risks, you need to go off script, and that is where using Innovative Tests can help.

Let us say a client wants his legacy application revisited but have no formal documentation, testers could employ exploratory testing to learn how the system currently works, and simultaneously gather information for next steps.

In this phase a suggestion would be conducting your Innovative Tests not completely adhoc. Follow a scenario based checklist that can be easily reviewed by the client. Once these are finalized, you can convert them to formal test cases. This will **reduce** back and forth review communication drastically. Since this was not done adhoc and a documentation process was followed, this can be **reused** in the design phase. While information is collected, it is also wise to capture as many pictorial references that can be **recovered** from a common storage point. This will help when you want to go back versions for cross verification. Not only do these scenario based checklist and pictorial references going to serve as check points but are also going to help build both the test planning for the next stages and the final reports of summary. These reports help the customer have a safe and convinced feeling about what is covered and will give them an opportunity to widen the scope if they see a need.

Traditionally test executions were focused only on scripts that limited the scope of testers. When Innovative Tests is followed it will let testers to think out of box. Innovative Tests results in an almost whole test coverage and the count of residual defects go down.

We used Innovative Tests for the current year and compared the residual defect rate with last year. Data collected were analyzed and results were obtained. Our results clearly showed that residual defect rate has decreased rapidly due to the implementation of Innovative Tests.

7 Case Study

Project Type: Java based web application

Resources: A total of 15 resources employed for testing

Test Type: Test scripts are automated and also run manually

Release Frequency: Monthly

Number of Test Modules: 250 (For each cycle)

Resources are primarily categorized into Manual testers and Automation testers. Manual testers are further grouped into testers using Innovative Tests & SMEs. Build frequency is 5 to 6 per month. Nearly around 2500 test cases are present in the library for the application.

General Challenges:

- Minimize re-testing efforts and achieving 100% testing coverage
- Identification of software changes and its impacts
- Identification of affected software test cases
- Reducing the re-test suites
- Wise selection of test cases from test suite for regression testing

Process Flow:

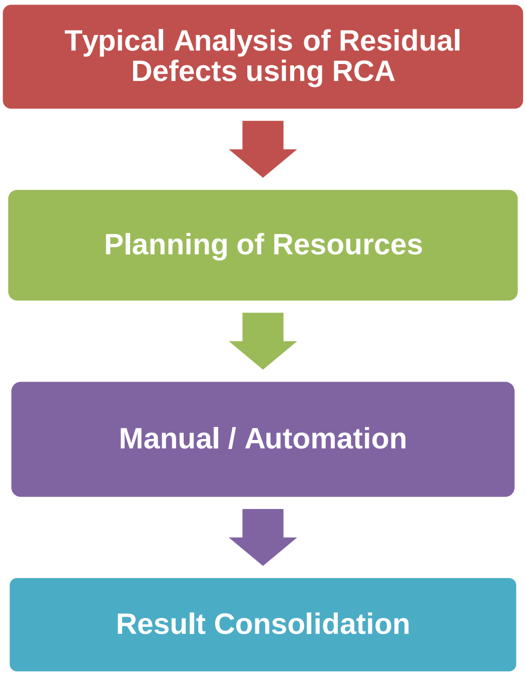


Fig 2: Work Flow

7.1 Metrics

Defect Comparison with customer:

Figs. 3, 4 and 5 show, how the defects leaked to Customers have gradually decreased after implementing the 3R approach compared to defects leaked to Customers before implementing the 3R approach. Also Fig. 4 shows how the curve has gradually gone down for the residual defects which is a good sign resulting in a satisfied customer.

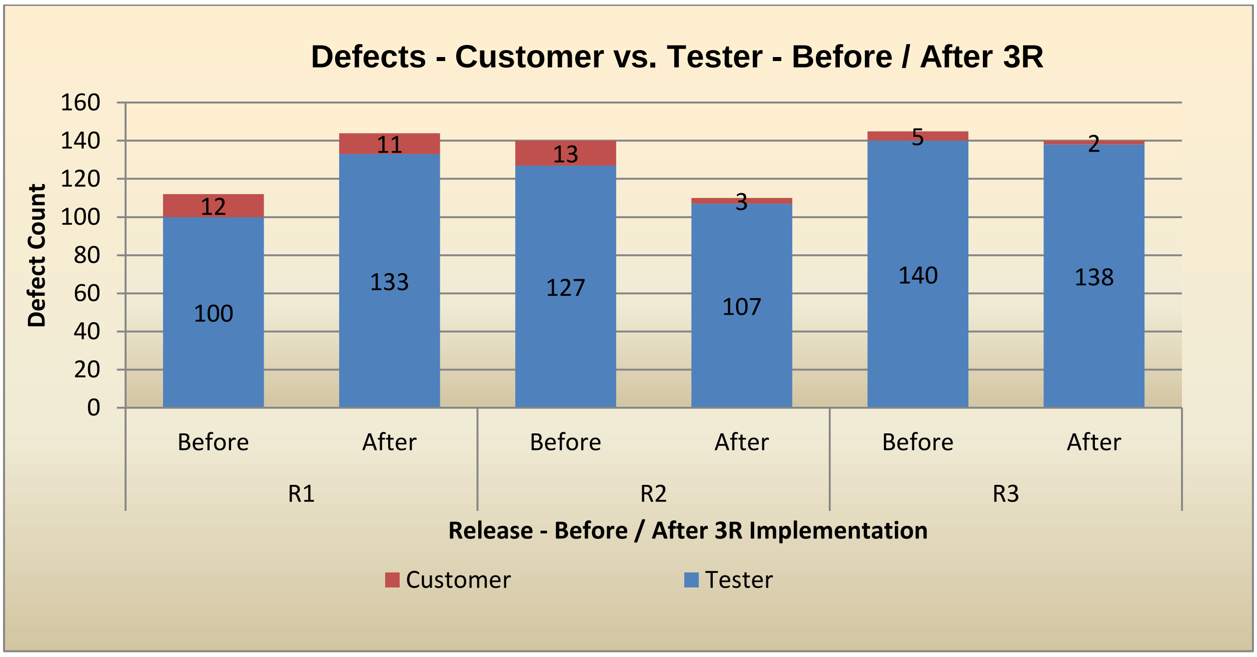


Fig 3: Defects - Customer vs. Tester - Before / After 3R

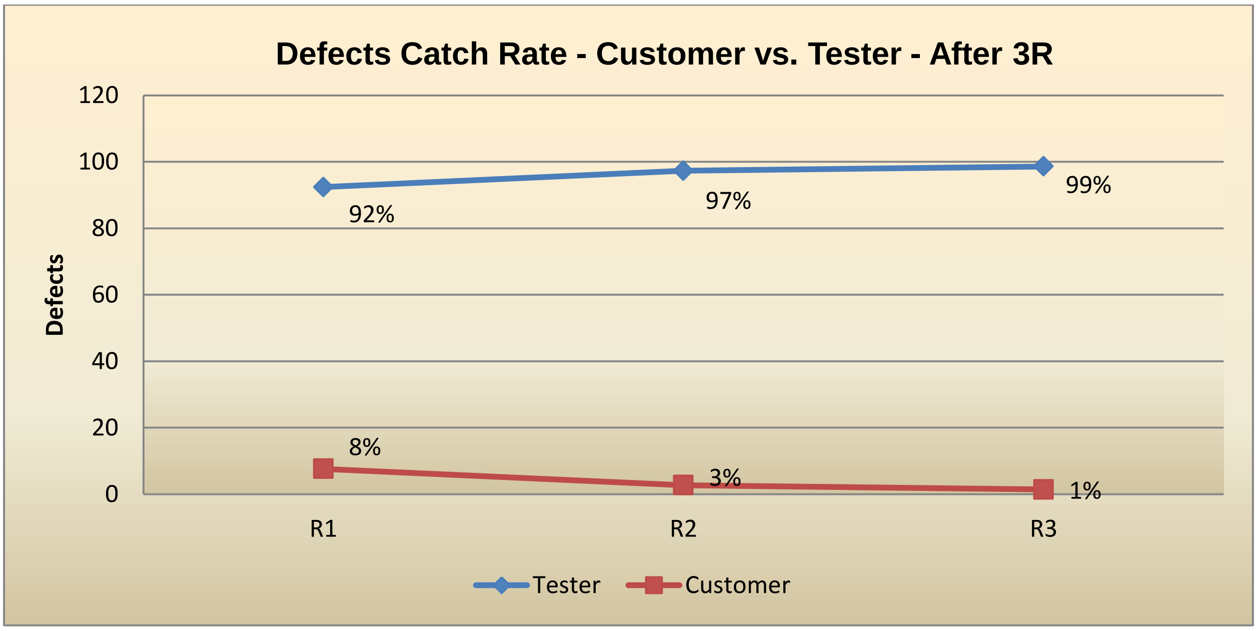


Fig 4: Defects Catch Rate - Customer vs. Tester - After 3R

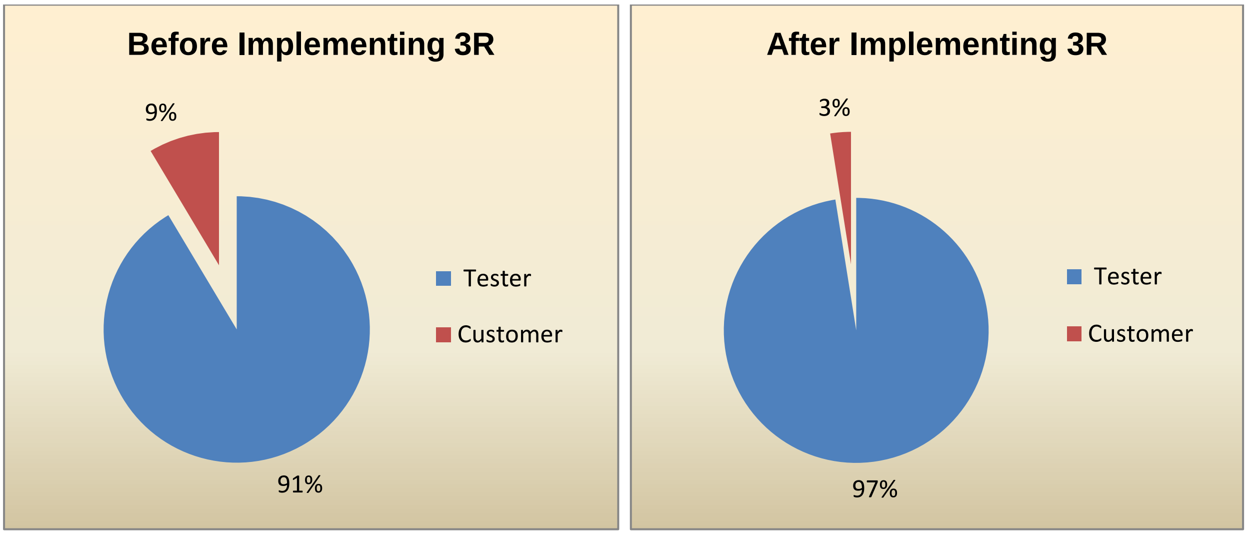


Fig 5: Before/After Implementing 3R

Defect Distribution using Innovative Tests:

Fig. 6 shows, how the defect rate has increased by performing Innovative Tests than the usual Test Case based Testing. There is ideally about 8% more defects detected in the Innovative Test approach.

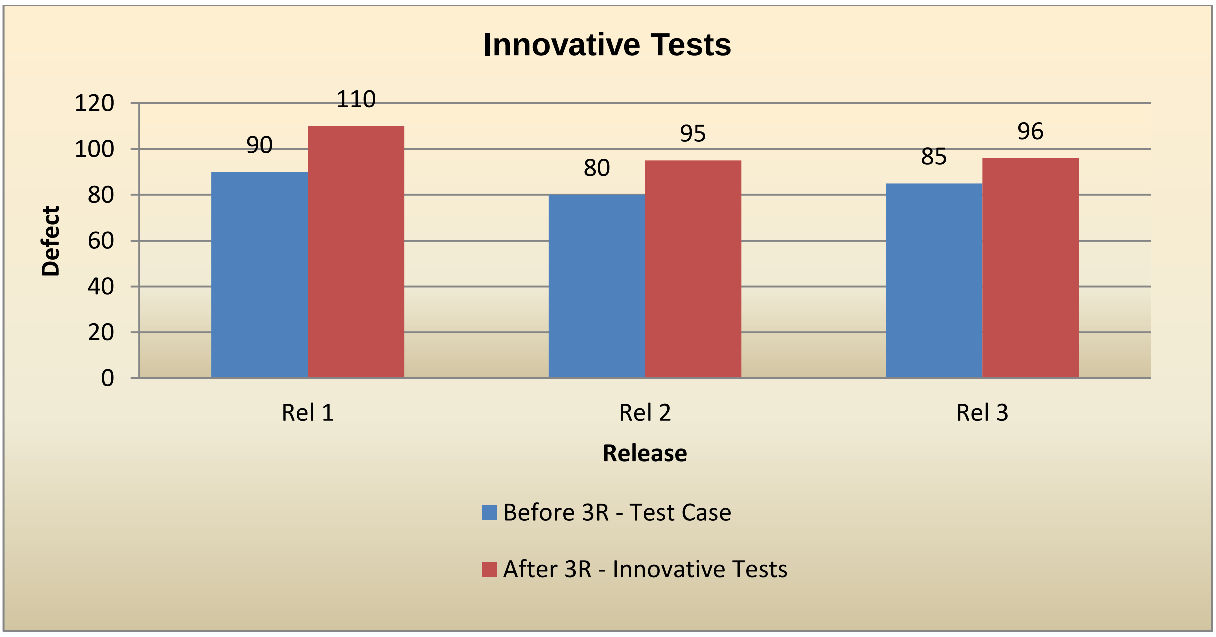


Fig 6: Innovative Tests

Defect Distribution Tester vs. SME:

Fig. 7 shows, defects identified by the SMEs are higher by about 7% compared to Testers. Three Vital concepts of SME in our process are Formal scenario documentation as a Value addition, Wider Scope and Suggestions on enhancements. All these, most of the time result in happy customers as these show that we have understood their product well.

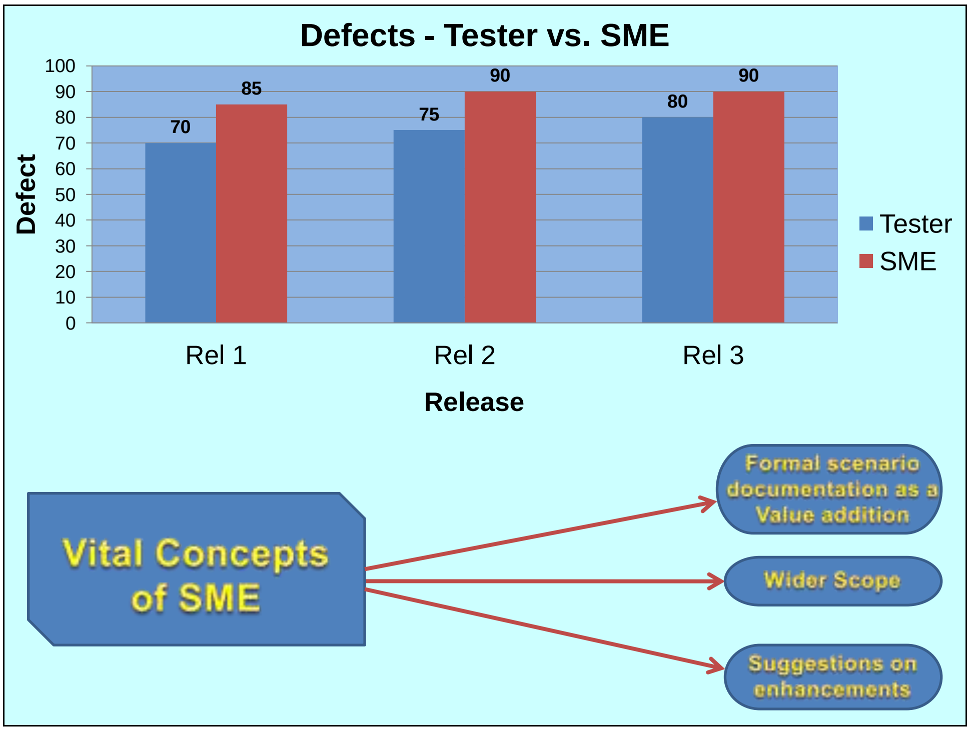


Fig 7: Defects - Tester vs. SME

8 Benefits

- Achieve risk-free regression testing at an optimal cost
- Provide precision in regression testing to facilitate maximum coverage through minimal number of test cases
- Increase the productivity and efficiency of quality assurance applications
- Efficient results in a short span cycle
- Reduce time to market significantly

9 Key Takeaways

- Re Usable & reliable techniques
- Best practices in regression
- Managing the changing environment

- Maximum test coverage
- Reduction in overlap testing
- Healthier, more reliable systems

10 Conclusion

On using the 3R approach for the regression testing, we observe that there is an effective changeover in the results. It also solves the challenges that are imposed by continual regression cycles. Due to the inclusion of SME concept, the end result not only eliminates the residual defect but also provide a value addition in the form of finding out of scope issues, suggestions to improve the product, etc.. It also proves to be a best practise for regression testing enabling maximum test coverage and reducing timeline issues. Therefore aim for delighted customers with all the possible test evidences with our 3R approach!

11 Author Biography

Naresh Jayabalan comes with 3+ years of experience in software testing and he is into compatibility testing project for more than 3 years. He holds a B.Tech in Information Technology from S.A. Engineering College, Chennai and a MBA in Human Resource Management from University of Madras. He strives himself into analyzing and experimenting defects which has lead the way to this paper. He spends his idle time in listening music & freaks out to adventure places.

Irene Preetika comes with 2+ years of experience in software testing. She is into compatibility testing project and was part of functional testing project. She holds a B.Tech in Information Technology from Dr MGR University, Chennai and pursuing MBA in Technology Management from Anna University, Chennai. She often loves to conduct brainstorming contests. She is passionate towards music, dance & indulges herself in art work during her idle time. She has contributed many presentations internally which lead the way to this paper.