



ENHANCING SOFTWARE QUALITY

Target Optimization – TMMi to Real Time

Author – Premalatha Shanmugham

Table of Contents

1	ABSTRACT	2
2	INTRODUCTION	2
3	THEORY OF OPTIMIZATION	3
4	OPTIMIZATION - GOALS	4
5	CHALLENGES	5
6	BALANCING SCOPE AND QUALITY	6
7	ALIGNING TOWARDS OPTIMIZATION	7
	ALIGN – DRIVE – MANAGE - IMPROVE	7
7.1	7.1 ALIGN	7
7.2	7.2 DRIVE	8
7.3	7.3 MANAGE	8
7.4	7.4 IMPROVE	9
8	OPTIMIZATION – KEY AREAS	9
8.1	8.1 SCHEDULE	9
8.2	8.2 TALENT / RESOURCE MANAGEMENT	9
8.2.1	On Demand Ramp Up / Ramp Down model	10
8.2.2	Trained Common Resource/Talent pool	10
9	OPTIMIZATION – CASE STUDY	11
9.1	9.1 CASE STUDY1	11
9.2	9.2 CASE STUDY2	13
10	HOW MUCH CAN BE OPTIMIZED	15
11	BENEFITS	ERROR! BOOKMARK NOT DEFINED.
12	REFERENCES	15
13	AUTHOR BIOGRAPHY	16
14	APPENDIX	ERROR! BOOKMARK NOT DEFINED.

1 Abstract

Go to market – Vite! This is applicable for testing as it is always towards the end of SDLC. Testing optimization is one of the best ways to achieve this goal. Considering the fact that time is always at a premium the need for having an optimized testing process is very essential. Given this challenge how does one ensure a cost and time effective testing process? How the resources in the project are utilized?

What are the parameters that affect the testing process?

Two kinds of software-testing process management problems are considered: testing-resource allocation to best use specified testing resources and a testing-resource control problem concerning how to spend the allocated amount of testing-resource expenditures during it. The parameters that affect testing are considered and analyzed to obtain the best optimization techniques.

Optimizing solutions vary for various projects – legacy applications with bulk regression suites, applications with constraints in testing schedule, high volume data applications, and multiple system interactions. Implementation of these techniques directly contributes for the project cost optimization and ensures sufficient test coverage which is the need of the hour.

This paper discusses the cost effective way to achieve test optimization without compromising the quality. The optimal allocation and control of testing resources can improve reliability and shorten the testing process. Based on this case studies of these two software testing management problems are presented.

The key takeaways of this paper are:

Utilization of resources

Best practices in scheduling

Managing the controlled environment

Maximum test coverage

Reliable techniques

Reduced IT overhead

Healthier, more reliable systems

2 Introduction

Software testing is a core activity in quality assurance. Compared to the earlier decade testing has evolved from a monotonous task to an extremely multi-layered and probably the most critical and stimulating activity in the software development life cycle. This change is due to some of the factors like

- Petite release cycles
- Complex functionality of the products being built
- Dedicated infrastructure
- Resource requirements
- Emerging software and hardware

Also the innovative trends in testing have resulted in a change in people's outlook towards it.

Recent study shows 60% of the top management expects their organization to be inundated with change largely driven by innovation and transformation. This results in the need of new evolving best practices in testing and numbers of research works published in this topic. But there always seems to be a lack of proper framework or process around refining the testing culture in the organization. What are the steps an organization can take to embark on the Testing & Optimization journey? Are there any best practices which one can learn from and potentially apply in their cases?

Due to the growing importance of testing management for business success, it is increasingly important to enable organizations to achieve sustainable competitive advantages. However, no approach has been developed yet which allows organizations to determine their current state of testing management on a process and product level, which can derive necessary steps for further development. To fill these gaps, we propose to develop new approaches. There is no universal approach that can be practiced by the universal testing teams to optimize. Some of the good approaches can be taken as a baseline and implemented.

3 Theory of Optimization

According to TMMi the purpose of Test Process Optimization is

- to continuously improve the existing testing processes used in the organization to identify new testing technologies (e.g., test tools or test methods) that may be appropriate and to transition them into the process
- Support the re-use of test assets

These improvements support the organization's product quality and test process performance objectives as derived from the organization's business objectives.

The test process is quantified and can be fine-tuned in order for capability growth to become an ongoing process. Establishing test process assessment and improvement procedures with responsibilities assigned from a leadership perspective :

Identification of bottlenecks

RCA for the bottlenecks

Remedial measures

Innovative improvements / Identification of reusable assets

Implementation of the innovations

Research on the new test-related tools and technologies for adaption

Supporting Knowledge transfer

Reuse of test assets

Assessment of implementations

Continual improvement

Deriving best practices

To improve the test process, we can use best practice models which describe in detail what to do in organizational test processes. The improvement activities using best practice models are performed as follows: checking the current status of test processes, suggesting and planning new actions, implementing the actions and measuring and controlling those actions.. However, it is difficult to apply all of these actions to the organization due to the limitation of resources

The objective of this paper is to provide quick descriptions and case studies on Testing and Optimization and from which the practices can be considered for approaching. . In this paper, we suggest a strategy for optimizing test process action plans. The background of this research is TMM (Testing Maturity Model), which is the most representative test process model. By applying design of experiments to the TMM assessment procedure, we can accept the actions selectively by statistical significance and find the best solution.

4 Optimization - Goals

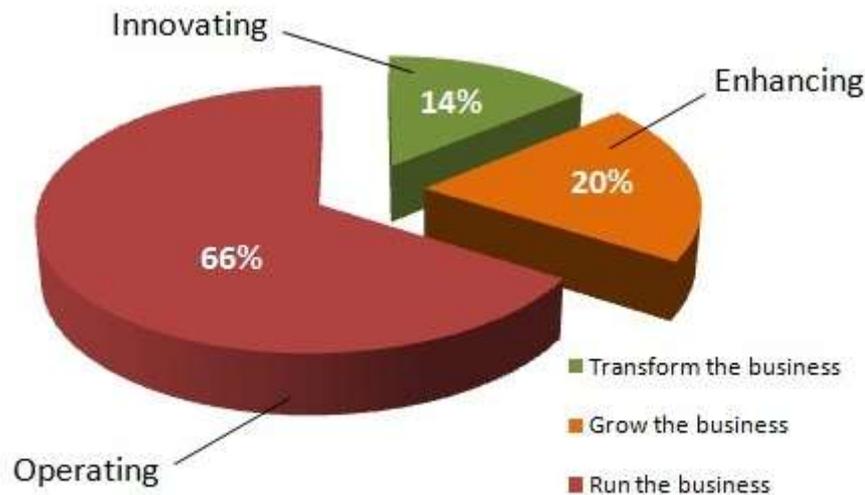
Utilizing the available resources as much as possible is called optimization. Optimization process is an incremental problem. At each and every evolution the solution leads to the target function.

Test Optimization techniques helps in adjusting a testing process so as to optimize some specific set of parameters without violating some constraint. The goal of Test Optimization techniques are:

- 1) To minimize the cost
- 2) To minimize the testing effort – this involves reduction of test design, reduction of test cases and reduction of execution time
- 3) To maximize the efficiency which means – maximum test coverage with minimal resources, maximum throughput with efficient test cases
- 4) To minimize the time to market

With minimum Test Cases and maximum coverage, overall Testing get completed in shorter span of time, thus contribute in overall goal of minimizing time to market.

5 Challenges



Gartner's report "IT Metrics: IT Spending and Staffing Report, 2011"

According to a recent survey, roughly 60% of an organization's budget is spent on operations. Of the remaining 40 percent, 70 to 80 percent is spent finding and fixing defects. Consequently resource funding for software quality delivery is fixed at best.

World Quality Report 2013 says " Although spend on quality and testing activities continues to rise as a proportion of overall IT budgets, CIOs are still under pressure to reduce the total cost of QA and Testing. The fact that most executives interviewed cite cost reduction as the main driver for outsourcing QA suggests that testing teams are striving for greater efficiency and doing "more with less".

This year's report has identified a structural shift in the QA operating models adopted by organizations to optimize their throughput, processes and resources to deliver better-quality applications.

Without this shift toward centralization and industrialization, organizations will not be able to deliver the level of quality required to sustain business growth – even with a bigger QA budget. Improved operational efficiency must remain the key goal for the Testing and QA function – through specialist skill sets, methodologies and tight organization, rather than increased budgets."

Reducing operational cost is a buzzword in the industry. We reduce waste by doing less of what we should not do. One way of optimization is to reduce the operational costs. The costs spent on software and hardware resources, salaries and incentives of the potential talents, infrastructure should be monitored continuously and right methodology to be used to arrive at those numbers.

Impromptu measures to control the costs may lead to lack of motivation and should be rightly dealt.

In the real world business expects test teams to do thorough and exhaustive testing but project cost and timeline does not make it possible. But as an optimization goal minimizing the test case design and

reduction of test execution time may end up in insufficient test coverage. This leads to more defects leaked into production which directly impacts the 40 percent and takes it to a higher level.

Impacts of a wrong optimization technique may lead to

Doubled spend on operations

Insufficient test coverage

Redundant testing

Defect leakage

Negative feedback

Finally it ends up in loss of money and the customer.

6 Balancing scope and quality

The goal of this paper is to find those time spending risk factors and reduce them, speed up software testing life cycle and reduce the operating cost.

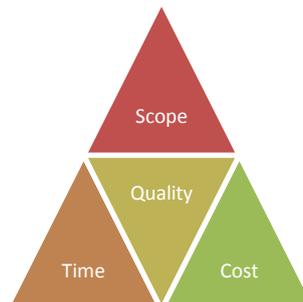


Figure (i) – Balancing Scope and Quality

This paper examines the balance between business scope and risk. It looks at the reasons for software development's changing landscape and it discussed which solutions can help your business improve quality management to overachieve in time to market, cost and enhance product quality taking the baseline from TMMi on optimization.

We increase effectiveness and efficiency by replacing waste with work that yields more value. A tester or testing practice that yields more information of greater value is more effective. An effective tester or practice that yields more per unit time or cost is more efficient.

One of the key ways to improve effectiveness is to master more of the available test techniques and understand when they are especially useful. Different techniques are optimized to expose different types of problems under different circumstances. For many type of bugs (think of race conditions, memory

corruption, or insufficiency of available resources as software usage increases) we are blind without well-tailored techniques.

One of the key ways to improve efficiency is to improve a tester’s skill in applying test techniques. This improvement is primarily fostered by personal coaching or small group training. However we need to be cautious about the training investment itself. Too much tester training is superficial and essentially skill free. It does not take much testing skill to pass a multiple choice certification exam.

7 Aligning towards optimization

Align – Drive – Manage - Improve

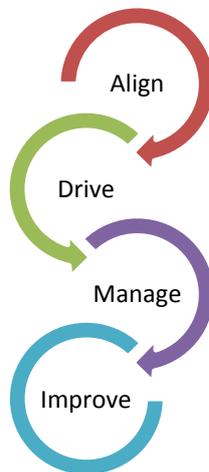


Figure (ii) – Aligning towards optimization

Many testing and optimization groups face similar challenges – prioritizing tests, identifying appropriate resources and communicating results. Appropriately prioritizing these testing ideas is incredibly important if you want to maximize the returns from your testing program, and this paper recommends a simple learning from the system which balances level of effort of testing and implementation, perceived accuracy of the hypothesis, and business impact.

7.1 Align

Improving the testing process to incorporate and accurately prioritize a broad base of ideas, iterating rapidly with experienced resources and managing expectations on test results will mitigate many of the common challenges many testing programs face.

Aligning the test ideas to a common objective is inherently analytical. Getting completely absorbed in the ideas however a common trap is. It is often the qualitative insights that can either help you come up with great testing ideas or discover “why” the observed tests results occurred. The process of aligning towards optimization goals should have the following :

- Outline test Goals > Identify Bottleneck > Brainstorm Test Hypotheses

The following are the 5 tactical suggestions to align testing:

- Find out why people are not testing to the objective. – Ask the right questions and try to get feedback at the point of non-alignment or deviation.
- Use the exact words that the customers use
- Start with a hypothesis and then prioritize – Have key questions for each experiment and a framework for prioritization
- Create experiments that are small – Rather expending huge amount of testing resources into a single test, see if you can run smaller experiments to incrementally learn.
- Continuously experiment – keep iterating in small steps and “pass the baton”

Key takeaways: Coming up with good tests is essential but hard. Combine numerous sources of evidence (qualitative and quantitative) to design your next test. Test incrementally and continuously to minimize wasted effort. Before you start to conduct split tests, prioritize the tests from the brainstorming phase. Consider the overall impact of the test as well as how long the test will take.

7.2 Drive

Once the first test is outlined, with the prioritized tests created and double checked goal tracking was in place, we are ready to start the first test!

Once the first test has run with recording of the time taken, review the results. Identify the key takeaways and implement the necessary changes. Now restart the testing process to test the learnings again. Apply this information to upcoming tests to continuously improve.

7.3 Manage

Now that the process is driven, the ground work for optimization is done and it is time to get it measured to manage it further.

Identify the metrics to be measured viz effort, schedule variance, resource. Arrive at the numbers to truly know what is working and what is not when it comes to the optimization strategy.

7.4 Improve

The metrics shows the deviation and where the control has to be exercised, concentrate on those bottlenecks to improvise. This is a continual process until we reach the set optimization goal of 10% - 40%

8 Optimization – Key areas

8.1 Schedule

While preparing a schedule for a streamlined execution, it is very important to have certain points called milestones. To keep in progress and on track of the projects schedule it is very important to reach these milestones.

These milestones are usually agreed with the client but rarely all of those are met in time. Failing to meet them in time, means extra cost for both sides; to the client for company's efforts and to the company for its employees work.

As the testing contains several activities, it is extremely important to bring in those attributes into the schedule. The first estimation would always be quite optimistic assuming all test cases pass without any show-stopper defects. Of course this is not realistic. Some of the test cases will have various defects that require time in order to get fixed and that will definitely affect the schedule.

For these, consider the planning and analyzing outcomes of the tests which are both added to the time spent on those.

8.2 Talent / Resource management

Cost is directly linked with number of resources/talents working in a project. According to Gartner's survey 85% of the project cost or budget is consumed by the human resources. It is extremely important that how we are engaging the talents. Are they are enabled with proper work opportunities on a day to day basis. In software testing tester's also has many dependencies like waiting for deployment, defect fixtures, test environment instability and network downtimes.

As we talk about optimization the following are some of the techniques to achieve the objective :

- On Demand Ramp Up / Ramp Down model

- Trained Common Resource pool

8.2.1 On Demand Ramp Up / Ramp Down model

As we know in the different phases of STLC the people engaged may not have the same amount of work. It is imperative to perform due diligence while assigning the tasks to the team. It is the basic principle of project management that the team should be aligned to work and not work being aligned to the team. Effective resource deployment is the key to reduce QA cost down.

During the planning and design phase, senior level talents are engaged and in the execution phase the team size can be ramped up to speed up the process.

It is good practice to have some low priority tasks identified and readily available so that based on the phase the allocation of multiple tasks, 100% capacity utilization. Regression suite creations, creation of re-usable assets are good examples of tasks that can be assigned and also help in process improvement.

8.2.2 Trained Common Resource/Talent pool

Generally, the project team hiring gets started on the basis of the phase in the STLC and resources are deployed on the need basis. But when we try to ramp up the team during the execution phase there might be problems like lack of domain knowledge which impacts the productivity.

A detailed effective KT plan should be in phase for speedy transition into the billable mode. Domain expertise is the key factor in enhancing the quality of a product or project, and hence reducing the QA cost. A tester who has very good domain knowledge can test the application better than others.

The domain knowledge plays an important role in software testing as one of the software testing principles says “Testing is context driven”, which means the testing performed for a Financial application is different than testing done for Insurance/health care application. A tester with domain knowledge will prove to be productive to the project immediately as he or she will be able to understand the problem in depth and come up with solutions. This will in turn add value to the project and the product by ensuring that it meets the quality standards.

To enhance domain knowledge, one needs to establish the baseline or create the current skill matrix and shared. Once the talents know their current level of domain knowledge on a particular application, the improvement goals can be set to track it on a regular basis.

Here are a few advantages of Domain Knowledge:

1. Reduces the Training time: A tester can be productive quicker than a person that has little or no domain/industry knowledge.
2. Good idea on UI features: It will help the person for improving the look & feel as we'll catch more bugs at initial stage.
3. Good idea on Back end Processing: how effectively/efficiently is the data/code being handled
4. Domain knowledge is also important to defect triage: Knowing how the application will likely be used, and how it is expected to perform, will tell QA whether a given defect is trivial, significant, or critical.

9 Optimization – Case Study

9.1 Case Study 1

Project

Product is a scientific research based application which works on the agile model

Release happens every month. Has minimum 4 cycles to test for each release.

Objective

To perform Functional Compatibility testing across various OS / Browsers

Approach

Browsers selected on the Top market trends. Iterations planned by selection of test cases from the regression suite. New features and their impact areas is the target.

IT impacts

A few hours of deviation in the schedule in once cycle impacts the following cycle and thereby the release

Business impacts

Postponed release increases the cost and delay in the time to market

9.1.1 Schedule Optimization

Let us take a look at a real world scenario, where a deviation in the schedule increases the Project cost exponentially and how this can be minimized.

For the sake of simplification, we have taken 1600 hours of testing effort, in which 10 team members are required for 4 weeks. There are 4 cycles of testing to be carried out.

There are 2 options to manage this effectively.

In each week 25% of progress has to be made. A planned schedule which includes all the risk factors is worked out. The schedule should accommodate any changing business requirements or ambiguous business requirements. Further the following should also be considered : Not detected start-up failures, Critical defects, defect aging and test efficiency (number of defects found/number of tests).

We have jump start kits with known failure patterns across browsers. This serves as baseline to start with the tests. Automation is used for regression and new features with the developed new scripts. Also test optimization techniques like risk based testing helps to achieve most of the coverage.

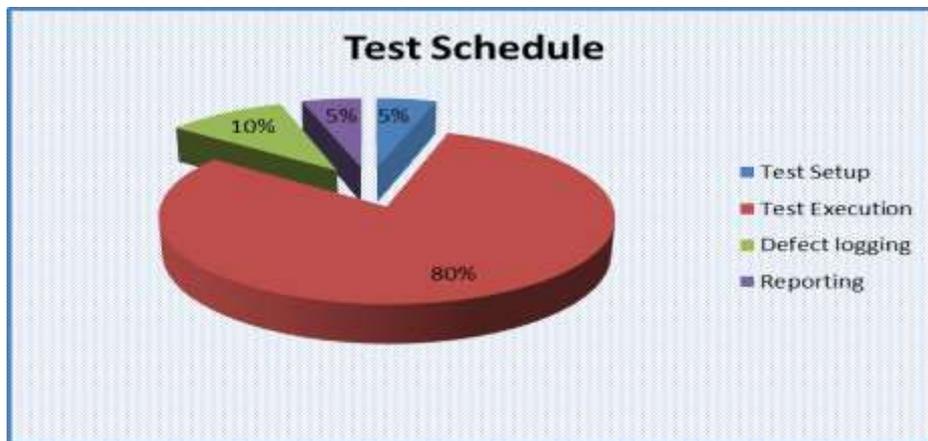


Figure (i) – Test Schedule - General

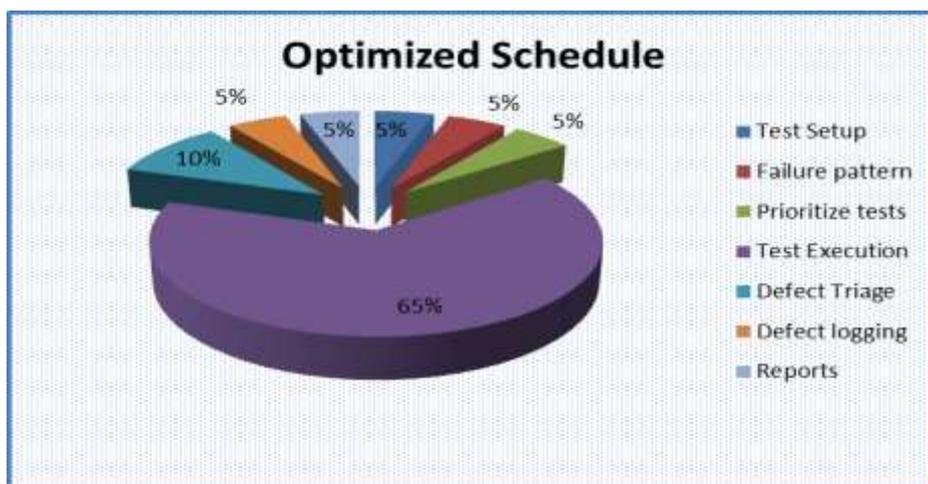


Figure (ii) – Test Schedule - Optimized

9.1.2 Illustration

Team size = 10	Planned	Actual	Optimized
Cycle 1	400	417	405
Cycle 2	400	414	405
Cycle 3	400	410	402
Cycle 4	400	410	395
Total	1600	1651	1607

Table 1: Total Effort

9.1.3 Benefits

- 10% More defects captured
- No schedule overrun as risk factors are included
- On Time every time
- Cost saved on delivering on time

9.2 Case Study2

Project

A retail based application which follows waterfall model.

Planned project period is 4 months.

Objective

To perform Functional testing

Approach

A comprehensive approach to test for the functional stability without any risk

IT impacts

Unfound bug found after release impacts the whole testing team

Business impacts

High cost for the defects leaked in production

Postponed release shortens the time to market

9.2.1 Resource Optimization

Let us take a closer look at this, 3200 hours of testing effort is needed with 5 resources. In the normal scenario we can engage a 5 member team with 3 senior test engineers and 2 test engineers.

Let us see how can this be optimized.

There are 2 options to manage this effectively. During the initial weeks and in the closure we need the effort of senior level resources and in the middle of the cycle we can engage test engineers.

The best option is to deploy more people from the trained common resource pool; test engineers in the execution phase to speed up the process. 2 Senior resources be involved in the initial phases and towards the end of the cycle; whereas from the trained resource pool we can pull in 2 more for execution.

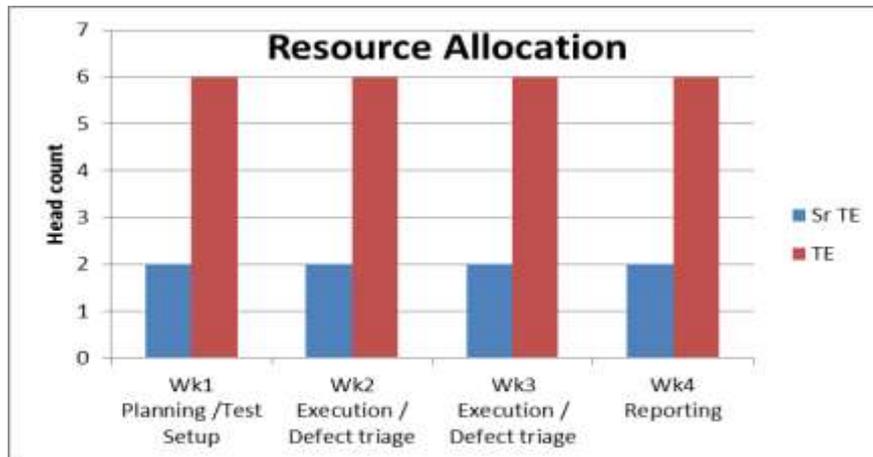


Figure (iii) – Resource Allocation – Usual practice

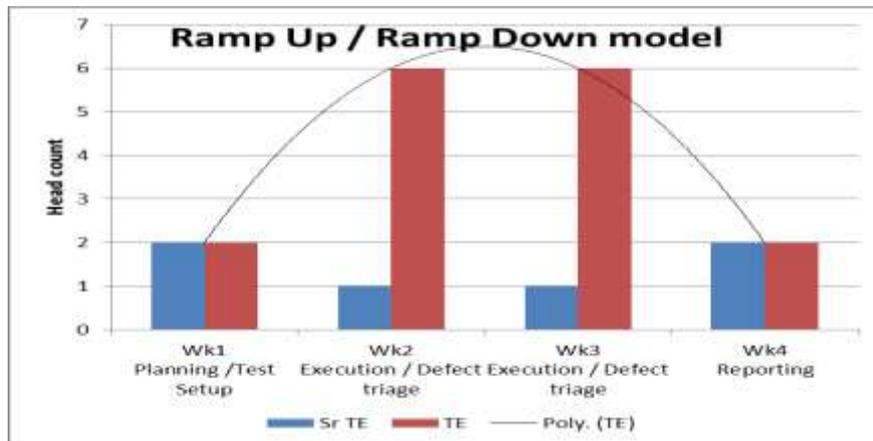


Figure (iv) – Resource Allocation – Ramp Up / Ramp Down model

9.2.2 Benefits

- 100% Capacity utilization
- Skill matched to Requirement
- Streamlined KT
- On Demand On Load / Off Load

10 How much can be optimized

How are best-of-breed firms improving software quality?

An IBM Global Services review of 846 projects across multiple clients revealed significant differentiated value with collaborative and automated approaches.

Comprehensive testing processes, integrated end-to-end lifecycle technologies, industry-based test-case and script reuse, and advanced defect analysis and quality management processes have shown the following improvement ranges:

- ✓ *Quality improvements from 30 to 70 percent*
- ✓ *Cycle-time improvements of 20 to 50 percent*
- ✓ *Cost reductions of 15 to 60 percent*

11 Key Takeaways

Saving of time and money can be achieved by paying more attention to optimization across the entire software development and delivery lifecycle.

How are the optimized solutions achieved? The following are the means to achieve an optimized test process.

Develop repeatable industry test solutions. It seems reasonable that within any given industry, test cases and manual scripts become fairly similar. That means copying and reusing those common test cases and scripts could net a substantial savings.

Perform advanced defect analysis. It's not easy to detect and prevent duplicate defects, but it's very important to detect them early. Otherwise multiple members will end up working on the same defect without knowing it, often producing even more defects. Automated detection of duplicated defects not only improve quality, but also reduces risk and cost.

Develop repeatable test procedures that are applicable to future projects. Developing new test artifacts for every project, rather than reusing available artifacts, is time consuming and unnecessary. Standardizing on repeatable testing procedures and techniques can save a considerable amount of time.

Integrating end-to-end processes. Integrating requirements with test cases and test results provides full bidirectional traceability from requirements to test cases and test results. These are key to leveraging competitive advantages by increasing quality and lowering cost.

In summary, a process-led, collaborative and automated approach to optimization not only enabled risk mitigation strategies and improved quality, but also helps to save cost.

12 References

<https://abckpn.googlecode.com/.../3.%20Software%20Testing%20using...>
<http://www.caam.rice.edu/~zhang/caam454/nls/MGH.pdf>

<http://www.funnelenvy.com/blog/2014/04/25/optimization-testing-takeaways-opticon-2014/>
<http://www.tmmi.org/pdf/TMMi.Framework.pdf>

13 Author Biography

Premalatha is working as a Project Manager in Indium Software and is responsible for Project deliveries. She comes with extensive experience in Compatibility Testing, CMS Testing, Regression Optimization, Test Optimization techniques implementation. Her initiatives around reducing QA cost, Improving Productivity, Resource Optimization, and Test Optimization techniques implementations has helped a lot for Quality improvements in Indium deliveries.