A Survey on Risk-Based or Design-Based Test Cases Prioritization

Sidra Laeeq, Rushda Shahid

Abstract— Software testing is an essential part of software engineering. During the software development life cycle, testing is highly required to assure the quality of the software process and product. The software development industry spends more than half of its budget on maintenance interrelated activities. Software testing offers a means to decrease errors, cut maintenance and overall software costs. These thesis discuses a new approach to software testing that enhances testing process using a mix approaches of exploratory testing and risk-based testing. Early in the history of software development, testing was confined to testing the ended code, but testing is further of a quality control mechanism. If software cannot be tested prudently, it must be tested selectively. A risk-based testing approach provides proactive opportunities to reduce the levels of product risk, opening in the initial phases of a project. The other is design-based test cases, where the focus is on ensuring that the software is capable of completing the core operations it was intended to do. These results can then be charity to improve testing organizations and to identify better practices for test case selection

Index Terms— Test Case Priority (TCP), Risk-Based Testing, Design Based Testing, Requirement Analysis, Software Testing, Comparisons, Verification And Validation.

1 Introduction

TEST case prioritization organizes test cases in a way to accomplish some performance goals efficiently. Rate of fault detection is one of most important performance objective. The test cases must run in an order that raises the opportunity of fault detection in the earlier life cycle of testing. Test case prioritization techniques have proven to be beneficial in improving regression testing activities. In this paper, we have proposed an algorithm, which prioritizes the system test cases based on the six factors: client priority, variations in requirement, implementation difficulty, requirement touch ability, and execution time and fault impact of requirement. We conducted a controlled experiment on two industrial data sets to compare the proposed value based test case Prioritization algorithm with random prioritization for early rate of fault detection. Average percentage of fault detection metrics has been used to measure the efficiency of proposed and random prioritization and it shows that the proposed value based algorithm is more efficient than random prioritization to generate sequence of test cases for early rate of fault detection.
Now a days software expansion environment testing has come under pressure due to shorter product
Time to market, shrinking budgets and higher quality demands. Software testing is an expensive time
The skill level of test personnel and size of the project impact consuming process, which is often restricted by cost and time constraints, defect detection rate. As does testability of the system. This is due to poorly specified requirements with inadequate description of user feedback. This is particularly apparent in test automation where test cases rely on user feedback in order to execute as expected. Software testing drives are commonly beleaguered by constraints such as time, cost, and insufficient Skills. These constraints impose risk on the realization of software test effectiveness with respect to software testing aims. Sympathetic how to moderate this risk is a key-factor in achieving successful Software testing. Our exploration aims to detect software test tools.

Numerous software development and testing methodologies, tools, and techniques have emerged over the last few decades promising to enhance software quality. Software testing is a trade-of between budget, time and quality. However, as the practice of software development has evolved, there has been increasing interest in expanding the role of testing upwards in the SDLC stages, embedding testing throughout his systems development process. The rapid change in the software
Development process brings many challenges to this field. In instruction to come across these challenges, the companies search for more agile and cost effective methods. This attitude is visible in all phases of software development process. The methods, approaches and techniques of software testing have developed to adapt his change.

2 Background Study

Testing is a field, which is well researched and surveyed, relatively, within software engineering. We study 12 research papers for our research and surveyed 8 organizations in their review. We identified empirical studies, evaluating from 8 organizations 5 organizations favored risk-based test cases selection rest of 3 organizations favored design-based test cases selection. Less than a third of the studies comprise Industry scale contexts. Since the area is well reviewed recently, we

- *Sidra Laeeq is currently pursuing bachelor's degree program in Information Technology in Jinnah University for Women, Pakistan. E-mail: sidralaeeq.bsit@gmail.com*

- *Rushda Shahid is currently pursuing bachelor's degree program in Information Technology in Jinnah University for Women, Pakistan. E-mail: rushda.shahid@yahoo.com*

here only focus on the work closely related to the topic under study, namely, empirical evaluations in industry on regression test prioritization and selection.

3 Research Methodology

Testing of software is difficult theory, it has many related model and different programming testing is a complex theory, which has a few related ideas and distinctive methodologies even with apparently comparable associations. Recognizing this, we chose to seek after exact subjective examination by applying the grounded hypothesis strategy. Grounded hypothesis was viewed as suitable for finding the premise of testing exercises, as it watches and describes genuine phenomena inside their social and organizational context. A grounded methodology empowers the distinguishing proof of new speculations and ideas, settling on it a substantial decision for programming building exploration, and thusly, fitting to our survey.

3.2 Risk-Based Testing (RBT)

- To mark risk-free project by means of best practices in risk management to achieve a project outcome that balances risks with quality, features, budget and schedule.
- It is basically for those project and application that is created on risk. Using risk, Risk based testing prioritize and emphasize the suitable tests at the time of test execution.
- Risk-based testing take starts early in the development, identifying risks to system quality and using that knowledge of risk to guide testing planning, specification, preparation and execution.
- Risk-based testing includes both modification – testing to provide chances to reduce the likelihood of faults, especially high-impact faults – and possibility – testing to identify work-around to make the faults that do get past us less harmful.
- Risk-based testing also contains determining how well we are doing at discovery and eliminating faults in critical phases.
- Risk-based testing can also include using risk examination to identify proactive chances to remove or avoid faults through non-testing events and to help us select which test events to accomplish.

3.2 Design-Based Testing (DBT

- Design -based testing can be used for generating GUI test scripts, but also for producing scripts for embedded systems or even manual test instructions.
- The approach is especially useful for projects with many test repetitions, but will not pay off if there are only a few. In addition, its efficiency depends on the re-usability of the design components.
- Where the design-based approach was applied, the organizations emphasized Management and policies.

3.3 Conduct RBT Working Periods And Evaluate Risk:

Risk based working periods are programmed for respectively risk item once risk limits are recognized and sharp. These risk exploration-working sessions help stakeholders to rank risk based on the defined limitations.

The following steps explain RBT period pre-requisites and method.

- Make arranged list of risks.
- Collect Quantitative inputs for the period– Transaction dimensions, Occurrence of use, Number of members.
- Collect Qualitative inputs for the session– SME Knowledge, Member, user impact, Degree of change, Criticality of a process, System complexity, Regulatory requirements
- Participants will evaluate the risk objects, work as a team to rate likelihood and impact of each item based on the defined criteria
- Risk category will be defined for each risk items as Must have, Important or Nice to have
- Once risks are identified and assessed, then define Mitigation strategy to reduce the impact of risk across unit test, system test, etc.
- Assign ownership across business, IT, and test domains to implement and monitor strategies

TEST DESIGNERS	DEVELOPERS, PROGRAMMERS AND TESTERS	MANAGERS; TEST AND PROJECT MANAGERS
Development approach	Leans towards agile method	Lean toward plan-driven methods
Testing resources	Limited	Sufficient
Explorative testing	Applied commonly	Applied rarely
Effect of policies in decisions on testing	Small, most decision are done in project level	Large; most decisions are based on company policies or customer requirements
Customer influence	In the testing process	In the design process
Limitations of the model	Test cases coverage become limited	Test process may become laborious to manage
Design concept	"What should be tested to ensure smaller losses if the product is faulty?"	"What should be tested to ensure that the product does what it is intended to do?"

4 Our Approach

This study identified several components, which effect the decision making process and resulted to two stereo typical approaches on test case selection and prioritization method, named risk based and designed based selection method was favored in organizations, in which the test resources were limited or competed, and decisions on test cases were made by the testers themselves or designers in the lower level of organization in design based approach, the selection and prioritization process was done by the project level management or dedicated expert in the risk based approach the focus of testing was on verification, "what should be tested to minimize possible losses from faulty product", whereas the design based approach focused on validation, "what should be done to ensure that the product does what it is supposed to do".

5 Selection Of Test Cases

The selection of test cases based on costs or related risk is not a novel concept. We argue cost evaluation methods for testing. By ranking the test cases based on their value, i.e. the amount of money lost if the test fails, a 20% speculation in testing is sufficient to achieve 80% of the software value. Similar results of testing cost-effectiveness that testing can be organized effectively even with as low as 15% Permission to make digital or hard copies of all or part of this work for personal or classroom use is decided without fee provided that copies are not made or distributed for profit or profitable advantage and that copies bear this notice, or republish, to post on servers, or to redistribute to lists, requires preceding specific permission and/or a fee. As for the test case selection approach, there are different methods, which vary in applicability or results, but in general, the testers seem to agree on applying risk-based selection. However, the criterion on which the selection is based on is usually incomplete or undefined. This often leads to a solution where risk analysis is based on individual experience and can be based. For example, for developers the priorities for technical risks may be well adjusted. However, risks linked to other stakeholders, concepts like legal costs and compensations, loss of reputation for the company or maintainability by third party associates, are probably beyond the scope of a single software developer the developers have to plan the development ahead to make sure that every needed case can be tested. Within software projects, this would require rigid plan-driven development or continuous testability analysis for verification purposes, which would obviously generate other expenses. In contrast, in some cases like in software product line development, the testability requirements and possibility for conformance testing are emphasized. Software development methods are geared towards producing quality in software products. For example, international standards like ISO 25010 define quality as an amalgam of eight attributes like reliability, operability or security. In addition to these definitions, real-life measurements like the mean time between failures or number of errors found in testing versus errors found after release may also be used as indicators for software development quality.

Organizational testing practices may also vary because of otheraspects, Even if the purpose of testing is to verify functionality and to increase product quality, practical applications do vary, as different approaches to software development allow different types of tests in different phases. For example, developing software with agile development methods differs from the traditional plan-driven approach to the degree that they can be seen as exclusionary to each other. On the other hand, several techniques like pair programming, code reviews, think-aloud testing or explorative testing have been developed to enhance product quality and ultimately make the testing process easier. Even the task of generating test cases from which the selection is made varies; for example, black box testing and white box testing define two approaches to case generation based on knowledge regarding the structure of the object being tested. However, these types of approaches focus on the generation process itself, not actually on defining how the test cases are selected, and in case of resource shortages, on the decision of which cases to include and exclude. Overall, there seems to be an abundance of information and studies regarding test case selection in regression testing with several different models for cost/benefit-calculations and usability assessment methods. However, there seems to be lack of studies in software development, where regression and conformance testing models are not applicable.

6 Our Survey:

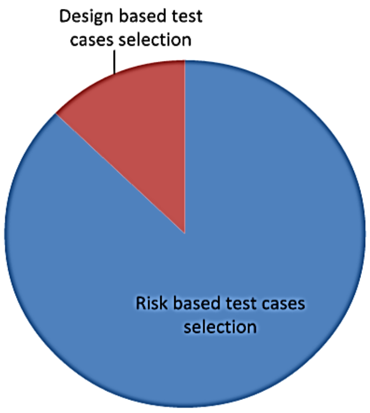
We make questionnaire related to our research paper test cases selection and prioritization and filled that form from 8 organizations

https://docs.google.com/forms/d/1GfVMVsnZrf6M9W8IvFrTShPI1PYkxKnTvoIPmHuuYDY/viewform

For our paper we used survey methodology

First we make questionnaire related to our paper which helpful for us

- 8 organizations filled our questionnaire
  → 5 organizations uses risk based selection
  → 3 organizations uses design based selection



6 Conclusion

Software testing is process of executing software in a controlled manner and where testing is an expensive but necessary maintenance activity performed on modified software to provide confidence that changes are correct and do not adversely affects other portions of the software. In this paper, we have proposed an approach to extract data from risk based and design based techniques and use for test case plans.

Following this risk-based testing approach, a test manager can ensure that the most effective testing is performed within the limited time and using limited number of resources available for testing.

In our observed organizations, test cases were selected using: (1) The risk-based approach (2) The design-based approach.

Organizations where testing resources were limited and the product design was allowed to adapt or change during the process; the risk-based approach became increasingly favored.

The limited time and resources available made it very important to have a well-defined test process. This included:

- *Interfaces to design, construction and unit test stages.*
- *Deliverables with defined quality standards.*
- *Test preparation and execution procedures stating entry and exit criteria for each test phase.*
- *Control procedures to handle scope changes and issues.*
- *Well defined organisation and responsibility including training of the testers.*
- *Progress Tracing.*

Organizations where more testing resources and the software design was made in a plan-driven fashion, the objective for the test process shifted towards test case coverage, and subsequently, towards the design-based approach in test case selection.

Intelligent effort allocation within constraints Priority-order bug discovery, optimizing bug-fix time windows , Flexible handling of reduced time and resources Optimization of quality within constraints by focused testing Key points: Include business users and potentially even customers in analysis Start risk analysis early in the life cycle Bottom line: Risk-based testing allows teams to deliver increasingly sophisticated, complex products, within tightening constraints, prioritizing testing and balancing quality with other priorities

Acknowledgement

At the start we would like to express our greatefulness to our instructor Miss Ramsha Ali to made this chore possible although we cannot pay accurate Research Paper to her devotion and keenness to ourselves and to our paper but we deliberated her right that her effort should go in history with our paper.

References

[1] http://dl.acm.org/citation.cfm?id=1852800

[2] https://archive.org/stream/arxiv-1202.6118/1202.6118_djvu.txt

[3] http://www0.cs.ucl.ac.uk/staff/mharman/stvr-shin-survey.pdf

[4] http://www.wseas.us/e-library/conferences/2013/Rhodes/CIRCOM/CIRCOM-08.pdf

[5] http://www.ijcsit.com/docs/Volume%202/vol2issue3/ijcsit2011020321.pdf

[6] http://www.iaeng.org/publication/WCECS2012/WCECS2012_pp156-160.pdf

[7] http://www.researchgate.net/publication/221553669_An_Automated_Technique_for_Risk based_Test_Case_Generation_and_Prioritization

[8] http://www.researchgate.net/publication/235751299_Risk based_Testing_Techniques_A_Perspective_Study

[9] http://www.researchgate.net/publication/252090399_RiskBased_Adaptive_Group_Testing_of_Web_Services

[10] http://collaboration.csc.ncsu.edu/laurie/Papers/PORT_ISESE.pdf

[11] http://www.scirp.org/journal/PaperInformation.aspx?PaperID=23971#.VHC-fDSUdAo

[12] http://www.rbcs-us.com/documents/Risk-Based-Test-Results-Reporting.pdf

[13] file:///E:/8th%20semester/software%20quality%20assurance/research%20paper%20(sqa)/ejise-volume14-issue2-article776.pdf

[14] file:///E:/8th%20semester/software%20quality%20assurance/research%20paper%20(sqa)/EJSR_74_3_07.pdf

[15] http://www.ijarcsse.com/docs/papers/Special_Issue/icadet2014/Lord_38.pdf

[16] http://crpit.com/confpapers/CRPITV135Garg.pdf

[17] http://www.cs.ndsu.nodak.edu/~hdo/pdf_files/icsm11.pdf

[18] http://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=1747450&fileOId=3738235

[19] http://dennisjeffrey.com/student_site/COMPSAC06.pdf

[20] http://cse.unl.edu/~myra/papers/quICSM.pdf

[21] http://www.satisfice.com/articles/hrbt.pdf

[22] http://books.google.com.pk/books?id=2WtlAnauLuEC&pg=PA40&lpg=PA40&dq=test+cases+selection+and+prioritization+risk+based+or+designed+based&source=bl&ots=9g5Pczy7Wk&sig=ZbwaRvtCcisSBnk1o4uxKeyBJAQ&hl=en&sa=X&ei=M-9xVPzBOeroywPm44C4Cg&ved=0CDkQ6AEwBA#v=onepage&q=test%20cases%20selection%20and%20prioritization%20risk%20based%20or%20designed%20based&f=false