

Requirements-Based Test Case Prioritization

Hema Srikanth

Department of Computer Science, North Carolina State University, NC 27695

hlsrikan@ncsu.edu

ABSTRACT

Test case prioritization techniques have been shown to improve regression-testing activities by increasing the rate of fault detection, thus allowing testers to fix faults earlier. The current techniques, mostly code coverage-based, treat all faults equally. I build upon this work to develop a prioritization scheme with two main goals: identifying the severe faults earlier and minimizing the cost of test case prioritization.

Keywords: Software Testing, Reliability, and Quality

1. INTRODUCTION

Software testing is a strenuous and expensive process. Research has shown that at least 50% of the total software cost is comprised of testing activities [7]. Companies are often faced with lack of resources, which limits their ability to effectively complete testing efforts. Test case prioritization (TCP) that involves execution of test cases in a pre-defined order can be cost effective [2, 3, 6]. As a result, the rate of fault detection early in the development cycle is improved, allowing the developers to rectify the faults earlier while improving the system delivery speed [3, 6]. I build upon previous work and apply TCP at system level.

2. PROBLEM STATEMENT

The current prioritization techniques [3, 6], mostly coverage-based (statement, branch or other coverage), can be expensive to employ and assume that all faults are equally severe in nature. In reality, some faults can be critical in nature (high risk failures), such as causing the customer significant monetary loss, delaying a product shipment, or having safety or security implications. Current TCP schemes can be enhanced by incorporating additional knowledge gained through requirements engineering research: (a) requirements with high complexity tend to have a higher number of faults [1], (b) requirements volatility, which results in re-design, addition or deletion of existing requirements, tend to increase project risk [5], and fault density [4], thus often times causing project failures [5], (c) roughly 20% of the system is responsible for about 80% of the faults [8].

Therefore, there is a need to maximize the effectiveness of the testing resources via a scheme which: (1) is economical (does not add significant overhead to team);

(2) improves perceived software quality; and (3) identifies the more severe faults earlier.

3. METHODOLOGY

I build upon the current TCP techniques [3, 6] and propose a multi-faceted prioritization strategy called Prioritization of Requirements for Testing (PORT Version 1.1) by exploring three prioritization factors (PFs): (1) customer-assigned priority on requirements, (2) requirement complexity, (3) requirements volatility. Our preliminary set of research goals is listed below.

G_1 : To identify the most severe faults/failures earlier in system test.

G_2 : To improve the software field quality.

G_3 : To devise the minimal set of PORT PFs that can be used to effectively for TCP.

The PFs, customer-assigned priority (CP), requirements complexity (RC) and requirements volatility (RV) are assigned values. CP is the value (1 to 10) assigned by the customer based on the importance of the requirement. RC is the value (1 to 10) assigned by the developer based on the perceived implementation difficulties of the requirement. RV is the number of times a requirement has changed. Higher factor values indicate a need for prioritization of test case related to that requirement.

Based on the project and customer needs, the development team assigns weight to the PFs such that the assigned total weight (1.0) is divided amongst the PFs. For every requirement, Equation 1 is used to calculate a weighted prioritization (WP) factor that measures the importance of testing a requirement earlier. Test cases are then ordered such that the test cases for requirements with high WP are executed before others.

$$WP = \sum_{PF=1}^n (PFvalue * PFweight) \quad (1)$$

An example, using sample PF values and weights, is presented in Table 1 to demonstrate the PORT approach. WP represents the weighted prioritization for the four requirements computed using Equation 1. The results of the table show the prioritization of test cases for the four requirements as follows: R_1 , R_4 , R_3 , and R_2 . The WP will vary with a change in factor-weights and factor-values.

Table 1: Sample Prioritization

Factors	R ₁	R ₂	R ₃	R ₄	Weights
Customer Priority	7	4	10	8	0.30
Req Complexity	10	5	4	9	0.40
Req. Volatility	10	8	5	5	0.30
WP	9.1	5.6	6.1	7.5	1.00

4. CASE STUDY

A postmortem analysis was conducted on an industrial project that comprised of 152 KLOC. The project planning started in February 2003, progressed into beta test in August, and delivered in October 2003. A total of 1030 faults were identified in system test: 16 of them were classified as Severity 1, 259 as Severity 2, 608 Severity 3, and 157 Severity 4 faults. I analyzed the fault detection rates of the Severity 1 and 2 faults. My goal was to determine the characteristics of the severe faults and their actual detection rates.

Figure 1 shows the rate of detection of Severity 1 and 2 faults. Thirteen percent of Severity 1 and 23% of the Severity 2 faults were found during beta testing. The company's goal for future releases is to reduce the detection of severe faults during beta testing to less than 5%. Figure 2 depicts fault density in the program modules. Seventy eight percent of the Severity 1 and 2 faults were found in Modules 1 and 2. The test team indicated that Modules 1 and 2 were the two most volatile and complex modules. Of the total number of Severity 1 and 2 faults detected, 12% of the Module 1 and 21% of Module 2 faults were found in beta test. Identification of severe faults later in the process cycle could result in a lack of time to thoroughly fix severe bugs, thus compromising software quality. These results motivate the development of the PORT TCP scheme.

5. SUMMARY

The preliminary results show that requirements complexity and volatility impact fault density. These factors were identified as initial set of PFs for PORT. I plan to further analyze the industrial data to determine the effectiveness of customer priority in our prioritization scheme. I believe that PORT could improve the effectiveness of testing activities as it (1) reduces the effort utilized for TCP in comparison to coverage-based techniques that prioritize based on the number of statements or branches covered, and (2) focuses on functionalities that are of highest value to the customer, and (3) improves the rate of detection of severe faults. Rectifying severe faults earlier is believed to improve perceived software quality. I will validate PORT with industrial partners. IBM and Tekelec have expressed interest in participating in this research.

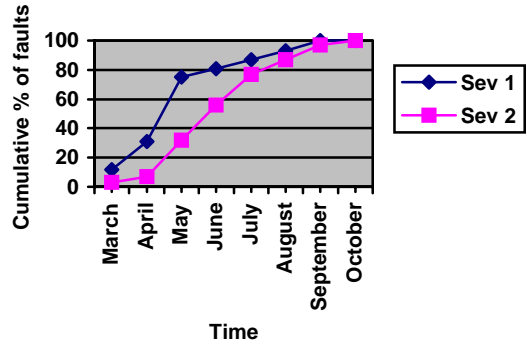


Figure 1: Fault Detection Rate

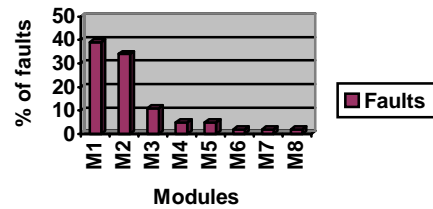


Figure 2: Defect Density in Modules

ACKNOWLEDGEMENTS

This work has been funded by the North Carolina State University Center for Advanced Computing and Communication (CACC).

REFERENCES

- [1] S. Amland, Risk Based Testing and Metrics, 5th Intl. Conf. EuroSTAR '99, Barcelona, Spain, 1999, pp. 1-20.
- [2] F. Basanieri, A. Betolino, and E. Marchetti. CoWTeSt: A Cost Weighed Test Strategy, Escom-Scope 2001, London.
- [3] S. Elbaum, A. Malishevsky, and G. Rothermel, "Test Case Prioritization: A Family of Empirical Studies," *IEEE Trans. on Software Engineering*, vol. 28, February 2002.
- [4] Y. K. Malaiya and J. Denton, "Requirements volatility and defect density," Proc. of 10th Intl' Symposium on Software Reliability Engineering, 1999.
- [5] J. O'Neal and D. Carver, "Analyzing the Impact of Changing Requirements," Proc. IEEE Int'l Conference on Software Maintenance, Florence, Italy, 2001.
- [6] G. Rothermel, R. Untch, C. Chu, and M. Harrold, "Test Case Prioritization," *IEEE Transactions on Software Engineering*, vol. 27, pp. 929-948, October 2001.
- [7] L. Tahat, B. Vaysburg, B. Korel, and A. Bader, "Requirement-Based Automated Black-Box Test Generation," 25th Annual Int'l Computer Software and Applications Conference, Chicago, Illinois, 2001.
- [8] E. Wong, J. Horgan, M. Syring, W. Zage, and D. Zage, Applying design metrics to predict fault-proneness: a case study on a large-scale software system, *Software Practice and Experience*, 2000, 30, pp 1587-1608.

105 BANYON TREE LANE
CARY, NC 27513

PHONE 919-466-7719
E-MAIL
hlsrikan@eos.ncsu.edu

Hema Srikanth

Objective

To obtain a research position in software testing.

Summary of qualifications

- Design and development of traceability tool to map requirements and test cases.
- Design and development of pair programming tool.
- Developed an agile approach to software development, ADaPT (Adaptive Development and Prototyping Technique) that employs scenario and goal-based analysis techniques to elicit requirements.
- Developed metrics to improve software process and software quality for ABB's Software Engineering Department.
- Tailored CMM for E-Commerce Companies to meet the Repeatable level.
- Proficient in C, C++, and Java.
- Courses in:
 - Software Reliability
 - Software Engineering and Risk Management
 - Parallel Architecture
 - Design and Analysis of Algorithms
 - Operating Systems
 - Compiler Construction
 - Software Process Improvement, Computer Networks, E-Commerce Technology, Online Systems, Database Management Systems, Managing Digital Enterprise, E-commerce Practicum, Graph Theory, Computational Methods to Molecular Biology.

Education

Fall 2002 – Present North Carolina State University Raleigh, NC

Ph D Candidate (Computer Science)

Fall 2000 – Summer '02 North Carolina State University Raleigh, NC

MS (Computer Science)

- Overall GPA 3.85 / 4.0

1999 - 2000 Clemson University Clemson, SC

Post Graduate in Computer Science

- GPA 4.0/4.0

1995 – 1996 Indiana University of PA Indiana, PA

Master of Business Administration

- Overall GPA 3.7 / 4.0

Publications

- Requirements-Based Test Case Prioritization- A Feasibility Assessment, Submitted to ISSTA Posters, February 2004.
- On Pair Rotation in the Computer Science Course, IEEE Conference on Software Engineering Education and Training, March 2004.
- A Model to Evaluate the Economic Benefits of Software Component Development, Accepted for IEEE CSMC, 2003.
- A GQM Approach to Measure the Benefits of CBSE, Asea Brown Boveri Internal White Paper, July 2001
- Tailored CMM for a Small e-Commerce Company, Level 2: Repeatable, TR, NCSU, 2001
- Towards Improved Requirements Practices in Agile Software Development, TR, NCSU, February 2002
- Towards Achieving a Balance between Planning and Agility in Software Development, TR, NCSU, September 2002.

Awards received

Presidents List Spring 2000 - Clemson University

Work experience

Fall 2000 - Present NC State University Raleigh, NC

Research Assistant

- Design and development of requirements traceability tool.
- Design and development of pair programming tool.
- Goal based analysis of electronic commerce privacy policies.
- Designed UI for SMART (Scenario Management and Requirements Analysis tool), which is a tool used to elicit requirements.
- Assisting Management and Computer Science students in using various software process models for their e-commerce projects.

Summer 2001/2002 ABB Raleigh, NC

Summer Intern

- Created Metrics to measure the benefits of Component based software engineering (CBSE) over traditional approaches.
- Introduced a process model that complies with ABB's GATE process model.
- Developed metrics to improve software quality.

1999 - 2000 Clemson University Clemson, SC

Web Designer

- Clemson University Bio-Engineering & Materials Department

Other Activities

Member of planning and outreach committee for Women in Computer Science [WICS] group.