

HSTCP: A Tool for Hierarchical System Test Case Prioritization

Harish Kumar*, Vedpal**, Naresh Chauhan***

Abstract

Many constraints are imposed on the software industries that want to complete the project in time and within budget. It is a challenging task for software industries to complete the project with in time and budget due to advancement of technology. In this paper a hierarchical test case prioritisation (HSTCP) (Kumar, 2013) tool is presented which implements hierarchical system test case prioritization technique. The presented tool helps to prioritize the system test case and reduce the testing cost of the software. The HSTCP tool works at three levels. At the first level it prioritizes the requirements, at the second level the modules of highest prioritized requirements are prioritized. Finally at third level it prioritizes the test cases of the modules in prioritized order. The presented HSTCP tool is implemented in Java language.

Keywords: Test Case Prioritisation Tool, Testing Tool, Software Testing

Introduction

Test case prioritization technique is used to order the test case in such a way that the highest error prone test case should be executed first and it should detect the error in earlier phase. The literature shows that the up to 50 % cost of total budget allocated is spent on testing of the software. A fault incurs more fixing cost in the later stages as compared to when the bug is fixed in earlier stages. It may be possible that critical bugs are found at the later stages. For fixing the bug at later stage it may need design

to be changed, which results in more testing cost. There are many approaches presented by the researchers i.e. some prioritized the test cases on basis of requirements, some prioritized the test case on basis of propose factors, code coverage, fault severity, and by applying genetic algorithms. It is very challenging and time consuming task to prioritize the test cases manually.

In this paper a HSTP tool for prioritizing the test case is presented. The proposed tool work at three levels as discussed below:

- At the first level the requirements are prioritized.
- At the second level requirements modules are prioritized in order of requirements.
- Finally at the third level the test case of the module in prioritized order are prioritized.

For experimental evaluation and analysis the presented tool has been applied on the case study. The considered case study performs the various operations to calculate the tax like saving, deduction, calculate tax etc. For effectiveness of the proposed tool comparison with the existing tool is performed. The result shows the effectiveness and efficacy of the tool.

Related Work

Marchetto, Susi, Kessler and Scan niello (2012) proposed a MOTCP tool for prioritizing the test cases based on sorting genetic algorithm and latent semantic indexing. The presented tool implements the multi-based test case prioritization techniques. The tool prioritizes the test cases on the basis of the code and requirements. It also considers the execution cost of the test cases.

* YMCA University of Science & Technology, Faridabad, Haryana, India. E-mail: htanwar@gmail.com

** Department of Computer Engineering, YMCA University of Science & Technology, Faridabad, Haryana, India. E-mail: ved_ymca@yahoo.co.in

*** Department of Computer Engineering, YMCA University of Science & Technology, Faridabad, Haryana, India. E-mail: nareshchauhan19@gmail.com

Srikanth, Williams and Osborne (2005) presented a technique to prioritize the system test case based on the requirements. They considered the four factors to prioritize the requirements. These factors are the customer priority, requirement volatility, implementation complexity, and fault proneness.

Kavitha and Kumar (2011) proposed a regression test case prioritization based on the six factors. The considered factors are customer assigned priority of requirements, developer perceived code implementation complexity, change in requirements, fault impact of requirements, completeness, traceability and execution time.

Roongruangsuwan and Daengdej (2010) presented a new classification of test case prioritization techniques considering a new test case *Prioritization* method based on some along with practical weight factors. The considered weight factors are test case complexity, dependency, and test impact.

Berander and Andrews (2005) presented an approach to find an optimal subset of requirements resulting in trade of desired project scope against conflicting constraints. These constraints are schedule, budget, resources, time to market, and quality.

Herrmann and Daneva (2008) proposed a conceptual model of requirements prioritization. They used benefit and cost prediction to prioritize the requirements.

After reviewing the presented work by the various researchers in the area of test case prioritization, there is a strong need to add some more factors to make the prioritization effective. The researchers did not consider some factors that affect the test case prioritization. In this paper some new factors are considered for prioritizing the system test cases. These factors are developer assigned priority, show stopper requirements frequency of requirements, expected fault, cyclomatic complexity non-Dc path.

Proposed HSTCP Tool

The proposed tool works at three levels. For prioritizing the test case customers, testers, developers, and business analysts participate for prioritizing the test cases. The process of prioritization of test case is as shown below:

- Customers, developers, and testers assign the weight and value to the considered factors. They assign

the value within the range from 1 to 10 and factor weights are assigned in the range between 0 and 1. 0 mean less critical and 1 means most critical factors. The weights are assigned at one time for the projects and remain unchanged during the *Prioritization* of requirements of the projects.

- After prioritizing the requirements a mapping between the prioritizing requirements and the module is performed. After that the modules are prioritized in prioritized order of the requirements.
- At the third level of prioritizing the requirements, four factors are considered. The weights of the factors are assigned within range from 0 to 1 and values are assigned to the factors within the range from 1 to 10. Weight assigned to the factors shows the criticality of the considers factors

Prioritization of Requirements

For prioritizing the requirements the proposed tool accepts the weight assigned to the 12 factors. The range of assigning the weights to factors existed between 0 and 1. The considered factors are shown in the Table 1.

Table 1: Requirements Prioritization Factors

S.No.	Factors Name
1	Customer Priority
2	Developer priority
3	Requirement volatility
4	Fault Proneness
5	Expected Fault
6	Implementation Complexity
7	Execution Frequency
8	Traceability
9	Show Stopper Requirements
10	Penalty
11	Cost
12	Time

Figure 1 shows the screenshot of proposed tool by taking factors' weight in the range. The tool takes the name of projects, number of requirements in project to be prioritized, and the considered factors.

Figure 1: Assignment of Weight to Factors of Requirements

After assigning the factors' weight, the factors' values of all requirements are assigned. The values are assigned in the range from 1 to 10. This window also takes the name of the requirements whose value to be assigned. The screenshot of the window as shown in the Figure 2.

Figure 2: Assignment of Value to Factors of Requirements

After taking the factors values, the requirements are prioritized on the basis of the requirements prioritization factor value (RPFV). Value of the RPFV is calculated by using Formula 1. Highest value of RPFV indicates the highest priority of the requirements. The screenshot showing the order of prioritized requirements is shown in Figure 3.

$$RPFV_i = \sum_{j=1}^n (rfvalue_{ij} * rfweight_j) \quad (1)$$

Figure 3: Prioritization of Requirements and Assignment of Value to Factors of Module

Prioritization of Module

In this phase the requirements modules are prioritized in the prioritized order. The modules are prioritized on the basis of two factors. These factors are cyclomatic complexity and number of non-dc path in module. The test cases are prioritized on the basis of the module prioritization value (MPV). Higher the value of MPV, higher the priority of module. MPV can be calculated by using Formula 2. Figure 3 and Figure 4 show the screenshots of taking the factors value of module and prioritization order of modules.

$$MPV = (\text{Cyclomatic Complexity} + \text{Number of Non-DC Path}) \quad (2)$$

Figure 4: Prioritized Modules and Assignment of Weight to the Factors of Test Cases

Prioritization of Test Cases

In this phase test cases of the module are prioritized. The test cases of the modules are prioritized in the prioritized orders of the module i.e. the test cases of the highest priority module of highest priority requirements are prioritized first. The screen shot is shown in Figure 4. The weights assigned to the four factors show the capability of test cases to detect the error. The considered factors are shown in Table 2. After assigning the factors' weight, the value of each factor corresponding to all test cases is assigned as shown in Figure 5. The factor value is assigned in the range of 0 to 10.

Table 2: Test Case Prioritization

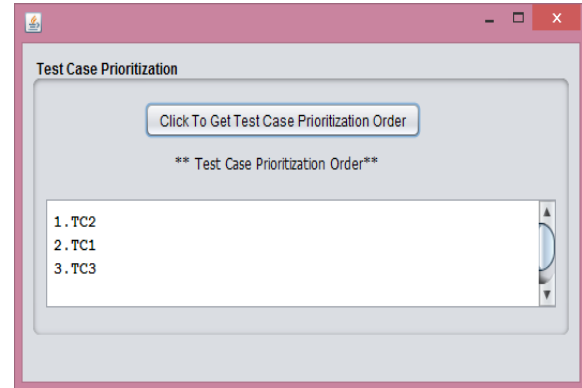
S.No	Test Case Factors
1	Test impact
2	Test Case complexity
3	Requirement Coverage
4	Dependency

Figure 5: Assignment of the Value to the Factors of Test Cases

After assigning the factors' weight and factors value test case prioritization value (TCPV) of each test case is determined by using Formula 3. The test cases are prioritized on the basis of value of TCPV. Highest the value of TCPV highest the priority of the test case. The screens hot showing the prioritized test cases is shown in Figure 6.

$$TCPV_i = \sum_{i=1}^n (tfvalue_{ij} * tfweight_j) \quad (3)$$

Figure 6: Prioritized Test Cases



Conclusion

The presented hierarchical test case prioritization (HSTCP) tool works at three levels. At the first level requirements are prioritized, at the second phase modules are prioritized, and at third phase the test cases are prioritized. The proposed tool has been implemented in Java. The graphical user interface of the presented tool is user friendly. The proposed tool saves lots of time and efforts to prioritize the test cases of the requirements. In the back end of tool a file is also created in which prioritized order of requirements, modules and test cases are stored. The results show the effectiveness and efficacy of the tool.

References

- Berander, P., & Andrews, A. (2005). Requirements prioritization. In Aurum, Aybüke (Hrsg.), Wohlin, Claes (Hrsg.), *Engineering and Managing Software Requirements*. Berlin, Deutschland SpringerVerlag, pp. 69-94.
- Chauhan, N. (2010). *Software Testing- Principle and Practices*. Oxford University Press.
- Herrmann, A., & Daneva, M. (2008). *Requirement prioritization based on benefit and cost prediction: An agenda of future research*. 16th IEEE International Requirement Engineering Conference, 125-134.
- Kavitha, R., & Kumar, N. S. (2011). Factors oriented test case prioritization technique in regression testing. *European Journal of Scientific Research*, 55 (2), 261-274.
- Kavitha, R., & Kumar, N. S. (2010). Test case prioritization for regression testing based on severity of fault. *International Journal on Computer Science & Engineering*, 2 (5), 1462-1466.

- Kumar, H., & Chauhan, N. (2012). Identifying and analyzing the research challenges in test case prioritization. *Journal of Intelligent Computing and Application*, Serial Publication.
- Kumar, H., Vedpal., & Chauhan, N. (2013). *A hierarchical system test case prioritization technique based on requirements*. 13th Annual International Software Testing Conference in India. December, 04-05, Bangalore, India.
- Kumar, V., Sujata., & Kumar, M. (2010). Test case prioritization using fault severity. *International Journal of Computer Science and Technology*, 1(1), 67-71.
- Marchetto, A., Susi, A., Kessler, F. B., & Scanniello, G. (2012). *MOTCP: A tool for the prioritization of test cases based on a sorting genetic algorithm and Latent Semantic Indexing Software Maintenance (ICSM)*. 28th IEEE International Conference.
- Roongruangsuwan, S., & Daengdej, J. (2010). A test case prioritization method with practical weight factors. *Journal of Software Engineering*, 4(3), 193-214.
- Srikanth, H., Williams, L., & Osborne, J. (2005). *Towards the prioritization of system test cases*. North Carolina State University TR-2005-44.
- Srivastava, P. R. (2008). Test case prioritization. *Journal of Theoretical and Applied Information Technology*, 4(3), 178-181.