Proceedings of the International conference on Computing Technology and Information Management, Dubai, UAE, 2014

A Methodology for Software Test-Case Selection in Constrained Environments Using Desirability Functions

Abrar Qureshi, Ira Weissberger
Department of Mathematics and Computer Science
University of Virginia's College at Wise
Wise, VA USA
1 College ave. Wise, VA 24293 USA
aqureshi@uvawise.edu

Abstract—**The completion of quality software products within the expected time frame represents a major problem for companies that develop software applications. As the field grows, the software industry continues to struggle with delivering quality products in a timely manner. A major cause for low quality in software products can be attributed to inadequate software testing efforts. In such cases, practical constraints, such as scheduling and cost leave decision-makers with the tough decision of partially executing the testing phase to deliver the software product on time. This creates a problem when determining the best subset of testing cases to execute. Therefore, it is important to develop systematic test cases prioritization processes that consider the complete set of functional or quality goals to be tested and provide the best fit to increase quality, reduce cost, and reduce testing time. This paper presents a methodology that considers multiple criteria to assign test cases to software projects. A sample case study is used to show the methodology's capabilities.**

Keywords: Software Engineering, Software Testing, Quality, Reliability, Decision-Making

I. INTRODUCTION

Employing an effective and efficient software test strategy is critical because it can significantly influence the quality of software products upon delivery. As documented in [1], the U.S. Department of Defense (DOD) spent nearly 8 billion dollars in 2004 to rework software. This large financial figure serves as evidence that quality-related issues continue to be a major struggle for software companies. A major reason that can be attributed to the software quality problem is tight project constraints such as schedule and cost [2]. In [2], Linberg stated that only about 16.2% of software projects are on time and within budget. When faced with such tight constraints, important decision-making that influences the resulting software quality becomes much more difficult. Previous work has tackled this problem from many different perspectives, such as optimizing the resource allocation of software engineers to tasks [3], [4], optimizing the selection process of requirements for particular projects [5], and prioritizing the selection process of test cases [6]. For the latter case, much work has been done which leads to effective but complex approaches that tend to be impractical for most real-world constrained environments. In this paper, the problem of software quality is addressed from the software testing perspective [7] . It proposes a simple but effective method for determining the desirability of test cases in projects with scheduling and cost constraints. In such cases, it is assumed that the whole set of test cases cannot be executed, thus leaving practitioners with the problem of selecting the best subset of test cases that provides the most best results given a set of project-specific criteria. The approach uses Desirability Functions [5] to provide a unified metric representative of the suitability of each test case given desired quality goals for the product. This way, decision makers can quantitatively delineate and assign test cases to projects when the complete test plan cannot be executed. The approach is extensible to consider a wide variety of product or process specific testing goals, such as security, usability, performance, availability, cost, etc. Moreover, the approach is simple and leads itself to usage through simple spreadsheets. Managers can use this methodology as a tool to increase the efficiency of test case assignment.

This paper is divided into five sections. Section II briefly describes the proposed methodology for test case selection. Section III presents a detailed coverage of desirability functions. Section IV presents results of the approach via several case studies. Finally, section V provides conclusions.

II. SOLUTION APPROACH

To properly make test-case allocation decisions in software engineering projects, decision-makers must follow a decision-making process that takes into consideration the perceived quality of test-cases present in specific projects. The creation of such process is achieved as follows. First, experienced project leads must identify the particular test cases required for a particular project. Then, from the pool of available test cases, each case is scored based on its perceived quality, which could include capabilities covered by the test case, importance of those capabilities, etc. In the proposed approach, we suggest a scale of 0 – 10, 0 being the least desirable and 10 being most desirable. Once each test case is ranked on all identified criteria, the scores are used as input to desirability functions to fuse this information and compute a unified score. Finally, the test cases with higher desirability score will get selected for the particular project.

III. Desirability Functions

Desirability functions are a popular approach for simultaneous optimization of multiple responses [8], [9], [10]. They have been used extensively in the literature for process optimization in industrial settings, where finding a set of operating conditions that optimize all responses for a particular system is desired. Through desirability functions, each system response y_i is converted into an individual function d_i that varies over the range $0 \le d_i \le 1$, where $d_i = 1$ when a goal is met, and $d_i = 0$ otherwise [10]. Once each response is transformed, the levels of each factor are typically chosen to maximize the overall desirability, which is represented as the geometric mean of all m transformed responses [9].

Similar to industrial processes, the software testing prioritization problem can be optimized by finding the set of criteria that provide optimal test cases for a particular software project. When formulated this way, desirability functions can be used to provide a unified metric that characterizes the suitability of test cases based on a set of predefined criteria. Once the desirability of all available test cases is computed, decision-makers can use this information to prioritize test cases by simply choosing the most desirable ones.

A. Computing Desirability

The first step in the desirability functions approach involves the selection of test cases for a particular task. Ideally, each available test case would be executed. However, in some practical scenarios this is not the case, where scheduling, cost, and other constraints placed on the project required that only a subset of the most appropriate test cases can be executed. The test case vector is presented in (1).

$$X = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} \tag{1}$$

Once the test case vector is identified, each test case can be evaluated against a set of product (e.g., functional or quality requirement) or process (e.g., cost) criteria $s_1, s_2, \ldots, s_m$ on a scale of 0 – 10, where 0 represents the lowest score and 10 the highest. This assessment is carried out using the test case assessment matrix S presented in (2). As seen, each y_{ij} value of the matrix represents the score of j^{th} test case based on each individual i^{th} criterion.

$$S = \begin{matrix} s_1 & s_2 & \cdots & s_m \end{matrix} \\ \begin{bmatrix} y_{11} & y_{21} & \cdots & y_{m1} \\ y_{12} & y_{22} & \cdots & y_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{1n} & y_{2n} & \cdots & y_{mn} \end{bmatrix} \tag{2}$$

Finally, to assess the importance of each test case criterion, a weight vector W is created where r_i represents the importance of the s_i quality criterion using the previously identified 0 – 10 scale. The weight vector W is presented in (3).

$$W = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix} \tag{3}$$

Once the information of X, S, and W is collected, desirability values can be computed using the desirability matrix d presented in (4). As seen, each d_{ij} value of the matrix represents the desirability of the j^{th} test case based on each individual i^{th} criterion.

$$d = \begin{bmatrix} d_{11} & d_{21} & \cdots & d_{m1} \\ d_{12} & d_{22} & \cdots & d_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ d_{1n} & d_{2n} & \cdots & d_{mn} \end{bmatrix} \tag{4}$$

Each individual desirability value d_{ij} is computed according to decision-makers' goals. For maximization (e.g., quality requirement) and minimization (e.g., scheduling, cost requirement), the desirability values are computed using (5) and (6) respectively [10],

$$d_{ij} = \begin{cases} 0 & y_{ij} \le L \\ \left(\dfrac{y_{ij} - L}{T - L} \right)^{r_i} & L \le y_{ij} \le T \\ 1 & y_{ij} > T \end{cases} \tag{5}$$

$$d_{ij} = \begin{cases} 1 & y_{ij} < T \\ \left(\dfrac{U - y_{ij}}{U - T} \right)^{r_i} & T \le y_{ij} \le U \\ 0 & y_{ij} > U \end{cases} \tag{6}$$

where L and U are the lower and upper limits (i.e., 0 and 10 respectively), T is the target objective (e.g., 10 for maximization, 0 for minimization), and r_i is the desirability weight for the i^{th} criterion. A desirability weight of $r = 1$ results in a linear desirability function; however, when $r > 1$, curvature is exposed by the desirability function to emphasize on being close to the target objective (T). When $0 < r < 1$, being close to the target objective is less important. Desirability functions can also be created to find responses on a given range. In this case, the response y_i is transformed using (7) [10].

$$d_{ij} = \begin{cases} 0 & y < L \\ \left(\dfrac{y-L}{T-L}\right)^{r1} & L \le y \le T \\ \left(\dfrac{U-y}{U-T}\right)^{r2} & T \le y \le U \\ 0 & y > U \end{cases} \tag{7}$$

Once individual desirability values for each skill are computed, the overall test case desirability value can be computed using (8). As seen, each overall desirability value is computed as the geometric mean of all *m* individual desirability values for test cases *1*, *2*, …, *n*.

$$D = \begin{bmatrix} \left(\prod_{i=1}^{m} d_{i1}\right)^{1/m} \\ \left(\prod_{i=1}^{m} d_{i2}\right)^{1/m} \\ \vdots \\ \left(\prod_{i=1}^{m} d_{in}\right)^{1/m} \end{bmatrix} \tag{8}$$

IV. CASE STUDY

This section presents results of a test case prioritization case study using the proposed approach. The case study assumes a scenario where 10 test cases are available. The identified quality criteria are: C_1 = security, *C2* = availability, *C3* = usability, *C4* = reliability and *C5* = cost. As seen, C1-C4 are criteria that decision-makers seek to maximize, whereas C5 represents criterion that needs to be minimized. Using synthetic data, the test case assessment matrix is presented in tabular form using Table I.

TABLE I. TEST CASE ASSESSMENT MATRIX FOR CASE STUDY 1

Test Case	**Test Case Quality Criteria**				
	C_1	C_2	C_3	C_4	C_5
1	6	9	10	5	9
2	3	3	3	9	2
3	5	7	4	8	7
4	8	9	8	9	9
5	7	2	5	4	8
6	9	2	3	7	5
7	10	2	4	9	3
8	6	1	10	1	9
9	5	1	8	7	9
10	3	6	5	10	2

As seen, for this project, testing the system's usability is of most importance, followed by availability and cost, security, and lastly reliability. Using the test cases from Table I and the parameters presented in Table II, the individual desirability values and overall desirability are presented in Table III. As seen, for this particular scenario, test cases 10, 4, 7, and 3 present the best overall fit for the required test-case objectives for the project. These test cases have desirability values of 59%, 55%, 55%, and 50% for the particular project. These results are representative of real world scenarios where none of the test cases are strongly desirable for testing every function of the system.

TABLE II. DESIRABILITY FUNCTION PARAMETERS FOR CASE STUDY 1

Parameters	**Test Case Goals**				
	C1	*C2*	*C3*	*C4*	*C5*
Target (T)	10	10	10	10	0
Weight (r)	1	1	1	1	1
Upper (U)	10				
Lower (L)	0				

TABLE III. SOLUTION SPACE FOR CASE STUDY 1

Test Case	**Desirability**					
	d_1	d_2	d_3	d_4	d_5	D
1	0.600	0.900	1.000	0.500	0.100	0.486
2	0.300	0.300	0.300	0.900	0.800	0.455
3	0.500	0.700	0.400	0.800	0.300	0.507
4	0.800	0.900	0.800	0.900	0.100	0.553
5	0.700	0.200	0.500	0.400	0.200	0.355
6	0.900	0.200	0.300	0.700	0.500	0.452
7	1.000	0.200	0.400	0.900	0.700	0.550
8	0.600	0.100	1.000	0.100	0.100	0.227
9	0.500	0.100	0.800	0.700	0.100	0.309
10	0.300	0.600	0.500	1.000	0.800	0.591

To show the effects of desirability weights on the overall selection process, the case study is modified to presents results of the allocation process when reliability (C_4) and cost (C_5) are identified as the most important criteria. The available test cases are presented in Table IV; the desirability functions' parameters in Table V; and the desirability results in Table VI. As seen, when reliability and cost are most important, the best identified test cases are 6, 9, and 7. In this case, it is evident that some test cases (e.g., 10) are completely undesirable for the project

TABLE IV. TEST CASE ASSESSMENT MATRIX FOR CASE STUDY 1A

Test Case	**Test Case Quality Criteria**				
	C_1	C_2	C_3	C_4	C_5
1	10	8	5	4	4
2	1	8	6	7	3
3	2	6	5	8	8
4	9	7	3	2	6

5	9	5	5	7	6
6	4	9	8	10	5
7	8	4	7	8	1
8	9	1	4	8	2
9	10	6	5	9	3
10	3	4	7	6	10

TABLE V. DESIRABILITY FUNCTION PARAMETERS FOR CASE STUDY 1A

Parameters	**Skill Set**				
	C_1	C_2	C_3	C_4	C_5
Target (T)	10	10	10	10	0
Weight (r)	1	1	1	10	2
Upper (U)	10				
Lower (L)	0				

TABLE VI. SOLUTION SPACE FOR CASE STUDY 1A

Test Case	**Desirability**					
	d_1	d_2	d_3	d_4	d_5	D
1	1.000	0.800	0.500	0.000	0.360	0.109
2	0.100	0.800	0.600	0.028	0.490	0.231
3	0.200	0.600	0.500	0.107	0.040	0.192
4	0.900	0.700	0.300	0.000	0.160	0.020
5	0.900	0.500	0.500	0.028	0.160	0.252
6	0.400	0.900	0.800	1.000	0.250	0.591
7	0.800	0.400	0.700	0.107	0.810	0.455
8	0.900	0.100	0.400	0.107	0.640	0.301
9	1.000	0.600	0.500	0.349	0.490	0.552
10	0.300	0.400	0.700	0.006	0.000	0.000

V. CONCLUSION

The research presented in this paper develops a systematic approach for selecting the best test cases for software projects based on multiple criteria. Specifically, it presented a methodology that uses Desirability Functions to create a unified metric that represents the suitability of test cases for particular software projects. This metric can be used as basis for test case prioritization. Through several case studies, the approach is shown successful in providing a way for analyzing multiple criteria to support decision-making using project-specific goals.

There are several important contributions from this research. First, the approach is simple and readily available for implementation using a simple spreadsheet. This can promote usage in practical scenarios, where highly complex methodologies for test case prioritization are impractical due to schedule and budget constraints. Another important contribution from the approach presented in this research is the ability to consider numerous decision-making factors in the decision-making process. For example, besides the criteria presented in the case studies, the approach can be easily extended to incorporate numerous other project-specific factors. Finally, the results provided by using this approach can be used by program managers to tailor scheduling goals to make them more realistic. For example, using the presented case studies, in some cases, it was shown that only a maximum of 59% test case desirability was possible. In other cases, test case desirability was as low as 0%. Depending on the overall desirability of all test cases, program managers can determine if either more resources need to be allocated for the task, or scheduling requirements need to be relaxed to deliver a high-quality product as result of the project. Overall, the approach presented in this research proved to be a feasible technique for efficiently managing and planning the test selection process in constrained software engineering projects.

VI. REFERENCES

[1] U. G. A. O. (GAO), "Defense Acquisitions: Stronger Management Practices Are Needed to Improve DOD’s Software Intensive Weapon Acquisitions," 2004.

[2] K. R. Lindberg, "Software developer perceptions about software project failure: a case study," *The Journal of Systems and Software,* vol. 49, pp. 177-192, 1999.

[3] S. J. T. M. A. Acuña, "Emphasizing Human Capabilities in Software Development," *IEEE Software,* pp. 94-101, 2006.

[4] L. D. Otero and C. E. Otero, "A fuzzy expert system architecture for capability assessments in skill-based environments," *Expert Systems with Applications,* vol. 39, no. 1, pp. 654-662, 2012.

[5] C. E. Otero, E. Dell, A. Qureshi and L. D. Otero, "A Quality-Based Requirements Prioritization Framework using Binary Inputs," in *4th Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation (AMS)*, pp. 187-192, 2010.

[6] S. Khan, I. Rehman and S. Malik, "The Impact of Test Case Reduction and Prioritization on Software Testing Effectiveness," in *International Conference on Emerging Technologies*, pp. 416-421, 2009.

[7] D. Drusinsky, J. B. Michael and M. Shing, "A Visual Tradeoff Space for Formal Verification and Validation Techniques," *IEEE Systems Journal,* vol. 2, no. 4, pp. 513-519, 2008.

[8] I. Weissberger, I. Kostanic and C. E. Otero, "Background Service QoS in a UMTS Network," in *Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon)*, pp. 230-233, 2010.

[9] G. Derringer and R. Suich, "Simultaneous Optimization of Several Response Variables," *Journal of Quality Technology,* vol. 12, pp. 214-219, 1980.

[10] D. Montgomery, Design and Analysis of Experiments, 7th ed., Wiley, 2008.