Productive Mechanism to Validate Program Segment by Using Test Case Reduction and Test Suite Prioritization

Pradeep Udupa, A. Rijuvana Begum

Abstract: Software Testing is investigation of software, it is used with goal of discovering defects in a system. It evaluates the system against faults, errors, bugs, failures and it is used to verify and validate application and improve efficiency of software. Hence testing system plays foremost role in developing product, but when test cases increases execution time, expenditure, effort, bugs, complication surpass, therefore proposed method is implemented by performing test case diminution, apfd, test case ordering to attain maximum efficiency.

Keywords: diminution of Test Suit, Test suit ordering, Parallelization, Fault detection, reduction, priority, testing.

I. INTRODUCTION

1.1 SOFTWARE TESTING
Software Testing is the method of verifying, validating and resolving bugs, which ensures bug free software with productivity and lessen overall executions [1], when test suit exceeds immense cycle time required, therefore here we first attempt to curtail test cases then order them and lastly increased efficiency.

1.2 System Testing
Used to process entire system and which is used to test entire software by integrating all the modules and to check developed product meets precise requirement given by client as in [7].

1.3 Alpha testing
in this testing performed to identify all possible errors before delivering the software and deploying the software to the users. here testing is performed to automate testing by actual users by means of black box and white box methods. This involves performing testing that a distinctive customer may execute.

1.4 Beta testing
This is a user testing executed at the end users situate by the end users to evaluate the functionality, usability, reliability of testing. Inputs provided by the end-users to extend the quality of the product further and leads to its success.

1.5 Gamma Testing
used to assess the functioning of the fully ready system with all the particular requirements for its marketplace.

II. LITERATUREREVIEW

Here development company frequently assess and stores the responses and feedbacks of the system, by placing in the full fledged market and The organization usually monitors and records the responses and feedbacks of the software product, by putting it in the open market, and assess system based on the feedback and response given by the market

1.6 Control Flow Testing
it is a white box testing method to test flow of program, in this different paths are opted by the code and the test cases are written for evaluating these paths, and used for verifying different conditions of a code segment.

1.7 Regression Testing
it is a form of testing to ensure that program segment or change in code has not severely exaggerated existing characteristics. In this either complete or part of test cases that are explored earlier are done for confirming that system executes without any severe errors, and problem.

1.8 Why to Attenuate Comprehensive Test Suits?
1. Immense test cases lead more convolution as in [9].
2. Extensive the test cases more will be the probable number of errors.
3. Bug tracing is to be executed.
4. There is a need of immense bug trackers.
5. Overall outlay will be increased as in[2].

1.9 Projected Method
In projected research we attempt to lower numeral of test cases by exploring constant, max, min values in the complete test suit cases though fixing numeral of test paths as follows. Identify conditions from preliminary to concluding in CFG.
1. Locate the variables with crest and nominal values in the pathway, afterward the variable with max value is given soaring value and tiny variable is given low down value.
2. Dwindle test case execution time.
3. Ordering check suits.
4. Overall exertion is lowered.
5. Eschew decisive software breakdown
6. Rectify Immersive Bugs.
7. Expose significant Bugs at the earliest.
2.1 Criteria Based Testing
DeMillo and Offutt introduced procedure for test input breeding that uses pathway checking, emblematic corroborations, and lower numeral of test cases depending on condition [12].

2.2 Dynamic Domain Reduction (Ddr)
Offutt et al., have invented method which reduces domain to test executions and overall time required. It has proved immense diminution in test suits but it is less proficient and huge time consuming, and is moderately more expensive technique[13]. exhaustation, prioritization, lastly test suits will be executed so that it can shrink debugging exertion as in [6], afterward we order test suits based on test case rank, here rectification of fault, apfd calculations, proportion of error discovery, risk discovery examination, are performed, here numerous formulas are integrated and test case priority is allotted for dissimilar test suits then we run based on test case priority, lastly contrast among different prioritized test suits is evaluated based on error revealing rate which exhibit that contrast to existing techniques our procedure has optimal efficiency over existing, finally we execute test suits based on precedence, this method exposes maximum numeral of bugs and runs test cases with brutal test cases primary and helps in protecting product from breakdown. state and number of states of obtained values. This method promises range covering, but it is requires additional time and it will be overpriced method and more exertion desirable [5].

2.3 Ping Pong procedure
Here it executes diminutive numeral of checking by arranging alternatively, Here heuristic procedure is implemented. It won't guarantee best yield, but given better result by using the number of values of target

1.4 Test Case Reduction Using Multi Constraint Reduction Technique & Fault Detection (proposed). several researches are carried out formerly. In proposed methodology we performed test case diminution, fault detection, prioritization, and parallelization here soaring, small, steady variable in all pathway are detected by verifying all individual path, in this exclusive paths are examined by means of cyclomatic convolution and afterward we run test suits in prioritized and parallel manner which condensed running time and numeral of test suits.

| Author                | Approach                          | Advantage                                      | Disadvantage                               |
|-----------------------|-----------------------------------|-----------------------------------------------|---------------------------------------------|
| DeMilli, & Offutt     | Constraint based testing          | Lessens Number of test cases, analysing time  | No prioritization,                         |
| 1991 [12]             |                                   | and storage to some extent                    | No fault detection                         |
|                       |                                   |                                               | Results in huge test cases                 |
| Offutt et al.,        | D.D.R.                            | Achieved more depletion in test cases         | Less efficient                             |
| 1999 [13]             |                                   |                                               | No, fault detection,                       |
|                       |                                   |                                               | Large time                                 |
|                       |                                   |                                               | No test case ordering                      |
| Srikanth et al.       | Prioritization Techniques         | Achieved ordering of test cases, Diminished   | Diminished productivity,                   |
| 2005 [11]             |                                   | time of execution                             | huge test cases and expensive              |

III. EXISTING METHODOLOGY
Ddr Technique
deduce that specified range as i1(0 to 30),j1(0 to 50), k1(0 to 40) we followed below method.
1. Locate every possible condition from start to finish.
2. Inspect range of split point for specified field and for all variable satisfying constraint.
3. After that as per value of split we separate domain into 2 ranges. i1(0..15) ,(16.. 30) i2 into (10..30) ,( 31. 50) and final interval by using splitting is i1(0..10),(11.. 30) i2(31..50),i3 10 therefore finaltest checks=651=31*1+31*20.

IV. PROJECTED PROCEDURE
In proposed method algorithm is used diminish test cases then we perform test case ordering by assigning priority for test cases and evaluation of A.P.F.D conducted in which proposed has given prolonged performance than previous technique, then parallelization carried out to reduce cost and execution cycles. in proposed method distinct independent paths are found and every path is exposed for minimum, maximum, and stable values and test case lessening is done by following steps given below then parallelization, prioritization performed. suppose we chosen path 1-2-4-8 let us assume we have inventive domains 1(0..30),j1(0..50),k1(0..40). We follow following steps.
1. discover nodes with conditions from beginning to end. m1 < m2, m2 = m3
2. discover low and high values in the path and assign to min and max variable.
3. discover stable values. ‘m3’ stable value derived on 2nd node is allotted to variable.
4. locate stable values. ‘m3’ stable value obtained on 2nd node is allotted to variable m3, then make use of obtained domain to derive reduced test cases for all distinctive paths as mentioned in Figure 1.

| Variable M1 | Variable M2 | Variable M3 | Test suits |
|-------------|-------------|-------------|------------|
| (0..30)     | (50)        | (10)        | Test 1     |
| (0.9)       | (10..50)    | (10)        | Test 2     |
| (10..30)    | (0..30)     | (20)        | Test 3     |
| (30)        | (0.50)      | (20)        | Test 4     |

V. RESULT EVALUATION

Here discrimination of projected technique is performed with the existing process Get Split by comparing produced entire test cases, entire diminution in checklist, executed execution time and error revealing rate in case of proposed technique required test cases 31*1*1=31 is test cases.

![Figure 1: Flow Graph](image)

### Table 2: Derived Test Cases

| Variables | M1  | M2  | M3  | Test suits |
|-----------|-----|-----|-----|------------|
| (0..30)   | (50)| (10)| (10)| Test 1     |
| (0.9)     | (10..50)| (10)| Test 2 |
| (10..30)  | (0..30)| (20)| Test 3 |
| (30)      | (0.50)| (20)| Test 4 |

### Table 3: Bugs Covered With different Time

| Bugs/Errors | Tst 1 | Tst 2 | Tst 3 | Tst 4 | Pr_ No | Pr-_rev | Pr-_prop |
|-------------|-------|-------|-------|-------|--------|---------|----------|
| Bug1        | 4     | 4     | 1     | 1     |
| Bug2        | #     | #     | #     | 1     | 4      | 1       |
| Bug3        | #     | #     | #     | 1     | 4      | 1       |
| Bug4        | #     | #     | #     | 1     | 2      | 3       |
| Bug5        | #     | #     | #     | 1     | 2      | 3       |
| Bug6        | #     | #     | #     | 1     | 2      | 3       |
| Total bugs  | 1     | 2     | 3     | 4     |
| Execution time | 1   | 3   | 5   | 9   |
| Ruthlessness | 4     | 6    | 8    | 11   | 16    | 14      | 11       |

Here adopted domain is m1(0..30), m2(0..50), m3(0..40) here Flt1 is bug value inputted less than minimum range in which values for m1=1 and Flt2 is error value inputted is higher than max domain and in value for m1=31, and Flt3 is error value given is less than min domain, where value for m2=4 and Flt4->error given above than max domain and value for m2=51 where Flt5->error value given is lower than min domain and value for m3=5 where Flt6->bug value given is higher than maximum range and value for m3=41, then for errors with relentlessness given and is specified in Table 3. We then evaluate rate of error, percentage of error and risk detection analysis, by using the formula given below, then values of (r,f,t), a.f.p.d and (r,d,a) are summed.

Rate of Fault = (Nj) / (TIMEj) * 10 as in Table 4.
Percentage of fault=Number of faults / overall bugs*10
Risk detection=((N.J) *(S.J) / (T.J))

Test rank =(R.F.D)+(P.F.D)+(R.D.A) given in Table 5.

### Table 4: Evaluation of Rate of Fault, fault percentage, Risk and Test Case ordering.

| Test cases | Rate of Fault | Percentage of fault | Risk detection | Test rank |
|------------|---------------|---------------------|----------------|-----------|
| Tst1       | 10            | 1.66                | 4              | 15.66     |
| Tst2       | 6.6           | 3.33                | 4              | 13.9      |
| Tst3       | 6             | 5                   | 8              | 19        |
| Tst4       | 4.44          | 6.66                | 10             | 21.1      |

### Table 5. Test Suit Rank Calculation.

| Checklist | T.C.R=R.F.D+P.F.D+R.D.A |
|-----------|--------------------------|
| Tst1      | 15.66                    |
| Tst2      | 13.99                    |
| Tst3      | 19                       |
| Tst4      | 21.1                     |

### Table 6: Evaluation of projected Priority With Dissimilar Methods.

| No-Priority | Reverse-Priority | Proposed-Priority |
|-------------|------------------|-------------------|
| Tst1        | Tst4             | Tst4              |
| Tst2        | Tst3             | Tst3              |
| Tst3        | Tst2             | Tst1              |
| Tst4        | Tst1             | Tst2              |

### Table 7: Evaluated Average Bug in Percentage

| Priority Methods | Average Fault |
|------------------|---------------|
| No-Priority      | 46            |
| Reverse-Priority | 54            |
| Proposed-Priority| 58            |

Application 1: to evaluate continuous improvement of student, here if students first series mark is greater than 2nd series and 3rd series 25 marks/credits added if 2nd series is greater than first series and 3rd series 50 marks added and if 3rd internal is greater than first internal and 2nd internal 75 marks added as in Table 9.

Application 2: promoting banking/finance business providing added credit points for increase in deposits/loan above 5k or certain limit benchmarked by company and can enchased as cash (Table 10,11,12)

Application 3: promoting business by giving credit Promotions for increments in purchase amount for purchase amount above 500 rupees as in Table 13,14,15,16,18.
Table 8: Application Based Research Comparison of Proposed Method (Multi Constraint Based Reduction) with Existing Method.

| Appname                     | Domain          | Reduction technique                          | Total test cases | Execution time in seconds |
|-----------------------------|-----------------|----------------------------------------------|------------------|---------------------------|
| Student Continuous Improvement | (0,30) (0,50) (0,40) | Test Case With No Reduction                  | 64821            | 324105                    |
|                             |                 | Test Case With Criteria Reduction            | 24149            | 12074.5                   |
|                             |                 | Multi Constraint Based Reduction             | 31               | 15.5                      |
| Banking Business             | (0,31) (32,55)  | Test Case With No Reduction                  | 64512            | 32256                     |
|                             | (0,35)          | Test Case With Criteria Reduction            | 27648            | 13824                     |
|                             |                 | Multi Constraint Based Reduction             | 32               | 16                        |
| Business Promotion           | (0,25) (0,60) (0,35) | Test Case With No Reduction              | 57096            | 28548                     |
|                             |                 | Test Case With Criteria Reduction            | 32760            | 16380                     |
|                             |                 | Multi Constraint Based Reduction             | 26               | 13                        |

Table 9: Student Continuous Assessment.

| Reg. no | Name | Internal 1 | Internal 2 | Internal 3 | Result | Total   |
|---------|------|------------|------------|------------|--------|---------|
| 124101  | Raj  | 20         | 30         | 25         | 50 Marks Added | 125     |
| 124102  | Ravi | 15         | 20         | 40         | 75 Marks Added | 150     |
| 124103  | Ram  | 35         | 25         | 10         | 25 Marks Added | 95      |

Table 10: Banking Business Application 1.

Acc no  | Custname | Bank Deposit 1 | Bank Deposit 2 | Bank Deposit 3 | Result | Credits    |
--------|----------|----------------|----------------|----------------|--------|------------|
134101501 | Raj      | 18             | 45             | 35             | 50 credit points Added | 50      |
134101502 | Ravi     | 19             | 35             | 69             | 75 credit points Added | 75      |
134101503 | Latha    | 23             | 18             | 15             | 25 credit points Added | 25      |

Table 11: Banking Business Application 2.

Acc no  | Custname | Bank Deposit 1 | Bank Deposit 2 | Bank Deposit 3 | Result | Credits    |
--------|----------|----------------|----------------|----------------|--------|------------|
235101501 | deepak  | 38             | 75             | 45             | 50 credit points Added | 50      |
634101502 | nibha   | 30             | 55             | 85             | 75 credit points Added | 75      |
714101503 | sahana  | 85             | 67             | 35             | 25 credit points Added | 25      |

Table 12: Banking Business Application 3.

Acc no  | Custname | Bank Deposit 1 | Bank Deposit 2 | Bank Deposit 3 | Result | Credits    |
--------|----------|----------------|----------------|----------------|--------|------------|
235101501 | deepak  | 66             | 75             | 105            | 50 credit points Added | 75      |
634101502 | nibha   | 90             | 65             | 85             | 75 credit points Added | 25      |
714101503 | sahana  | 88             | 125            | 76             | 25 credit points Added | 50      |

Table 13: Business Promotion Application 1.

| Custname | Mobno  | Purchase1 Amt | Purchase2 Amt | Purchase3 Amt | Result | Credits |
|----------|--------|---------------|---------------|---------------|--------|---------|
| rana     | 9.188E+09 | 14            | 20            | 40            | 3rd purchase is greatest, so 25 marks added | 75      |
| Raju     | 8.618E+09 | 19            | 35            | 16            | 2nd purchase is greatest, so 50 marks added | 50      |
| ram      | 9.946E+09 | 23            | 18            | 15            | First purchase is greatest, so 25 marks added | 25      |
Table 14: Business Promotion Application 2.

| Custname | Mobno    | Purchase1 Amt | Purchase2 Amt | Purchase3 Amt | Result | Credits |
|----------|----------|---------------|---------------|---------------|--------|---------|
| rajani   | 7.188E+09| 34            | 40            | 50            | 3rd purchase is greatest, so 25 marks added | 75      |
| Ramya    | 8.618E+09| 29            | 65            | 36            | 2nd purchase is greatest, so 50 marks added | 50      |
| divya    | 6.946E+09| 73            | 68            | 45            | First purchase is greatest, so 25 marks added | 25      |

Purchase1 (1-10K), Purchase2(1-15K), Purchase3(1-20k)

| Test cases | executn time | reduced test cases | gain % | criteria based | executn time | reduced test cases | gain % | Multi constrai nt | executn time | reduced test cases | gain % |
|------------|--------------|--------------------|--------|----------------|--------------|--------------------|--------|-------------------|--------------|--------------------|--------|
| 3000       | 1500         | 3000               | No Gain| 1000           | 500          | 2000               | 66.6   | 10                | 5            | 2990               | 99.6   |

Table 15: Business Promotion

| ITEM/Purchases | QUANTITY | PART NUMBER | DESCRIPTION | REQUIRED DATE | UNIT PRICE | PRICE | Result |
|----------------|----------|-------------|-------------|---------------|------------|-------|--------|
| 1              | 10       | F54749      | NW Globe Chassis, Model 28/02/01 | 29.5 | 295 | 25 credit points |
| 2              | 15       | F85352      | NW Globe Light Kit 28/02/01 | 19 | 285 |
| 3              | 6        | F84352      | NW Globe 30" Oak Stand 28/02/01 | 30 | 180 |
| 4              | 20       | L86355      | NW Atlas Oak Stand–Bound 28/02/01 | 25 | 250 |
| 5              | 12       | L86362      | NW Atlas Young Readers Series 28/02/01 | 22.5 | 270 | 75 credit points |
| 6              | 30       | A86357      | NW Pen Roller ball 28/02/01 | 100 | 3000 |
| 7              | 40       | A86362      | NW Pen Fountain 28/02/01 | 100 | 4000 | 50 credit points |
| 8              | 50       | A86552      | NW Globe Beach Balls, 24" 28/02/01 | 90 | 4500 |
| 9              | 50       | A86553      | NW Globe Beach Balls, 24" 28/02/01 | 70 | 3500 |

Purchase1 (1-15K), Purchase2(1-18K), Purchase3(1-20k)

| Test cases | executn time | reduced test cases | gain % | criteria based | executn time | reduced test cases | gain % | Multi constrai nt | executn time | reduced test cases | gain % |
|------------|--------------|--------------------|--------|----------------|--------------|--------------------|--------|-------------------|--------------|--------------------|--------|
| 5400       | 2700         | 5400               | No Gain| 900            | 450          | 4500               | 83.3   | 15                | 7.5          | 5385               | 99.72  |
Table 16: Business Promotion

| ITEM/Purchases | QUANTITY | PART NUMBER | DESCRIPTION     | REQUIRED DATE | UNIT PRICE | PRICE | Result               |
|----------------|----------|-------------|-----------------|---------------|------------|-------|----------------------|
| 1              | 1        | F54749      | Nw Laptop       | 9/5/2019      | 1000       | 1000  | 75 credit points     |
| 2              | 10       | F85352      | Nw Nokia 204    | 1/5/2019      | 290        | 2900  | 25 credit points     |
| 3              | 1        | F84352      | Nw Fridge Samsung | 4/5/2019    | 15000      | 15000 |                      |
| 4              | 2        | L86355      | Nw Setopbox     | 11/5/2019     | 1750       | 3500  |                      |
| 5              | 10       | L86362      | Nw Camera       | 13/05/19      | 1200       | 12000 | 25 credit points     |
| 6              | 2        | A86357      | Nw Keyboard     | 16/05/19      | 1500       | 3000  |                      |
| 7              | 25       | A86362      | Nw Mouse        | 17/05/19      | 160        | 4000  | 50 credit points     |
| 8              | 50       | A86552      | Nw Dframe       | 18/05/19      | 900        | 4500  |                      |
| 9              | 25       | A86362      | Nw Samsungnari  | 19/05/19      | 120        | 3000  |                      |

Table 17: Business Promotion

| ITEM/Purchases | QUANTITY | PART NUMBER | DESCRIPTION     | REQUIRED DATE | UNIT PRICE | PRICE | Result               |
|----------------|----------|-------------|-----------------|---------------|------------|-------|----------------------|
| 1              | 10       | F54749      | Bamboo Blinds   | 9/5/2019      | 29.5       | 295   | 25 credit points     |
| 2              | 15       | F85352      | Writing Sets    | 1/5/2019      | 19         | 285   |                      |
| 3              | 6        | F84352      | Umbrellas       | 4/5/2019      | 30         | 180   |                      |
| 4              | 10       | L86355      | Cane Chair      | 11/5/2019     | 15         | 150   |                      |
| 5              | 10       | L86362      | Cane Baskets    | 13/05/19      | 25         | 250   | 75 credit points     |
| 6              | 30       | A86357      | Tea Chests      | 16/05/19      | 10         | 300   |                      |

Purchase1 (1-20K), Purchase2 (1-25K), Purchase3 (1-30k)

| Test cases with no | execut time | reduced test cases | gain % | criteria based | execut time | reduced test cases | gain % | Multi constrain t | execut time | reduce d test cases | gain % |
|--------------------|-------------|-------------------|--------|----------------|-------------|-------------------|--------|-------------------|-------------|--------------------|--------|
|                    | 15000       | 7500              | 15000  | No Gain        | 3000        | 1500              | 12000  | 80                | 20          | 14980              | 99.86  |

Table 18: Business Promotion

| ITEM/Purchases | QUANTITY | PART NUMBER | DESCRIPTION     | REQUIRED DATE | UNIT PRICE | PRICE | Result               |
|----------------|----------|-------------|-----------------|---------------|------------|-------|----------------------|
| 1              | 100      | F54749      | Black Dragon    | 9/5/2019      | 200        | 20000 | 25 credit points     |
| 2              | 15       | F85352      | Black Dragon    | 1/5/2019      | 19         | 285   |                      |
| 3              | 6        | F84352      | Dinner Set (24) | 4/5/2019      | 3000       | 18000 |                      |
| 4              | 100      | L86355      | Wall units      | 11/5/2019     | 200        | 200000|                      |
| 5              | 100      | L86362      | Wall units      | 13/05/19      | 210        | 21000 | 75 credit points     |
Implementation Of Control Flow Graph

```c
int m1,m2,m3,t;
clrscr();
printf("Enter m1 in ist,2nd and m2 in 3rd for rej
00
00
00
marks in each interval is top so 50 mks bonus
Total=-?\n\n\n```

```c
if(m1>m2)
    t=m1;
else if(m1<m2)
    w=0;
else if(m1<m3)
    w=0;
```

```c
a.p.f.d=(1-(4+4+1+2+2+3)/6*4+1/2*4)=(1-(.666+.125))=(.466)=(46\%)
in case of reverse priority a.p.f.d=1-(1+(1+4+3+3+2)/6*4+1/2*4)=
(1-.5833+.125)=(54\%)
```
In projected method we initially compute average fault=58. afterward we parallelize derived test suits by referring Table 2, here m1 found in three paths so domain of m1 spitted into 3 parts m1) 0….10 m12)11……20 m13)21….30. Similarly range of m2 divided into 4 parts m21) 10….20 m22)21……30 m23)31….40 m24)41…50. Since m3 is constant we not divide m3. afterward requirements of the user considered along with priority and test cases executed in parallel manner hence we have overall numeral of test cases=[31*51*41]*4=259284 and condensed test case for path1=31, but using existing method test cases=651, condensed Total test cases=31*1*1+10*41*1+21*31*1+1*51*1]=[31*41+651*51]=1143, as in Table 2, then we assigning each test case constant .5 second but without parallelization running time essential for lessened test cases is 1143* .5=571.5. Hence whole amount of test suits without test case diminution = 259284 and running time=129642, but for chronological implementation condensed test cases =1143, and execution time=571.5 and in case of PRIORITIZED parallel running test cases=1143 and running time=142.875, where CPU having equivalent capability and error recognition rate of proposed method is also supplementary, in proposed technique test cases are executed with priority by reducing fault discovery time as specified in Table6.

VI. CONCLUSION

Each and every method has gain as well as pitfalls, d.d.r uses split points and isolated field, ping pong along with other established method consequences in massive numeral of test cases, tracing time, huge effort and expenditure. In case of proposed method has enhanced performance by attenuating overall test cases, prioritizing them and expose enormous number of bugs by assigning test case ranking, apfd computation and then generating prioritized test suits, attenuating test cases as in Figure.2. Here we compared proposed procedure with many other prevailing methods by application oriented research based comparison along with the evaluated result as in Table 8, and prolonged performance is accomplished by test case rank and finally parallelized and prioritized test cases are executed to lessen running time, total expenditure necessary for performing testing.

REFERENCES

1. Wang, R., Qu, B., & Lu, Y. (2015). Empirical study of the effects of different profiles on regression test case reduction. IET Software, 9(2), 29-38.
2. Boehm, B., & Huang, L.G. (2003). Value-based software engineering: A case study. Computer, 36(3), 33-41.
3. Arnicane, V. (2009). Complexity of equivalence class and boundary value testing methods. International Journal of Computer Science and Information Technology, 7(1)(3), 80-101.
4. Sawant, A.A., Bari, P.H., & Chawan, P.M. (2012). Software testing techniques and strategies. International Journal of Engineering Research and Applications (IJERA), 2(3), 980-986.
5. Jeng, B. and Weyuker, E.J. (1994). A simplified domain-testing strategy. ACM Transactions on Software Engineering and Methodology (TOSEM), 3(3), 254-270.
6. Gümöthy, T., Beszédes, Á., & Forgács, I. (1999). An efficient relevant slicing method for debugging. In Software Engineering—ESEC/FSE’99 (pp. 303-321). Springer, Berlin, Heidelberg.
7. Biswas, S., Mall, R., Satpathy, M., & Sukumaran, S. (2011). Regression test selection techniques: A survey. Informatica, 35(3).
8. Saifan, A.A., Aslakini, E., Alawnbi, H., & Sbaih, A.A. (2016). Test case reduction using data mining technique. International Journal of Software Innovation (IJSI), 4(4), 56-70.
9. Assi, R.A., Masri, W., & Zaraket, F. (2016). UCov: a user-defined coverage criterion for test case intent verification. Software Testing, Verification and Reliability, 26(6), 460-491.
10. Kumar, A. (2016). Evaluation of software testing techniques through software testability index. AKGEC, Vol. 3, pp. 342-349.
11. Srikanth, H., Williams, L., & Osborne, J. (2005). System test case prioritization of new and regression test cases. In 2005 International Symposium on Empirical Software Engineering, 2005, 64-73.
12. DeMilli, R.A., & Offutt, A.J. (1991). Constraint-based automatic test data generation. IEEE Transactions on Software Engineering, 17(9), 900-910.
13. Offutt, A.J., Jin, Z., & Pan, J. (1999). The dynamic domain reduction procedure for test data generation. Software: Practice and Experience, 29(2), 167-193.

AUTHORS PROFILE

MR.PRADEEP UDUPA, has completed his MTECH (CSE), MCA MPHIL and PERSUISING PHD (C.S.E). His field of interest includes software engineering, testing, operating system, simulation, dbms, wireless communication, adbms, Oops currently working as assistant professor in engineering college in Kerala.