Evolution of Test case Prioritization Approach in Software Testing and Role of Genetic Algorithm-Research in Progress

Surendra Mahajan, Dr.S.D.Joshi, Dr.V.Khanna

Abstract— Numerous outgrowths can be noted such as effort, schedule as well as estimation of the testing. Hence, diversified approaches as well as methodologies have been anticipated for acquiring awareness of these outgrowths. Application of genetic algorithms for automatic test development has been a domain of attentiveness for many researchers. Genetic Algorithm (GA) is one analogous aspect of evolutionary algorithms. In this research paper, we evince contemplate of GA approach for approaching the diverse outgrowths dealt with during software testing. There is a confirmed curtailment between academic as well as practitioner convictions on software testing. This paper benchmarks to secure the gap by benchmarking both convictions regarding the merits and boundaries of test case Prioritization. The academic convictions are examined with a consistent literature review while the practitioner's convictions are approximated with a survey, where we accumulated acknowledgments from plural software experts. The consequences of the balanced literature review materialize that the conception of apotheosis regarding merits and boundaries is quite superficial as only 30 papers ascribe the apotheosis. The survey emerged that merits of test case prioritization were accompanied to test case reusability, repeatability, test case coverage as well as effort reserved in test case executions. The boundaries were high alpha invests in Prioritization setup, tool preference along with educating. Comprehensively, 51% of the respondents accepted that feasible tools in the market assist a poor fit for expects and GA can be an alpha grade towards test case prioritization.

Index Terms—Evolutionary algorithm, Genetic algorithm , Prioritization, Regression testing ,Testing, Test Cases, Test case coverage

1. INTRODUCTION

From a benchmark perspective regression testing is a well developed analysis area. Furthermore, at the alike time it is determined that there is a gap between academic analysis as well as the merits and dilemmas explicitly challenged in addressing software testing in business. In [1] the former certainty is by practitioners as well as academicians. The apotheosis of automated testing heads at 100% automation [2]. Furthermore, in conduct this apotheosis has not been yet achieved (e.g. [3]). The analysis consequences are frequent analyzed in the form of case studies, competence catalogs, as well as benchmarks, which ascribes brilliant summaries along with insights, but at the alike time bounds generalizability for the software business at large. This emphasizes the thirst to benchmark what the business at large endures with respect to automated testing merits and limitations [4].

In the former, evolutionary algorithms have been addressed in ample real life dilemmas. GA is one analogous evolutionary algorithm. GA has arisen as an exercised, robust optimization approach as well as search mechanism [23, 27]. A GA is a search approximation that is driven by the way nature evolves species effecting realistic determination of the fittest individuals. The achievable breakthroughs to dilemma being discovered are circumscribed by a population of chromosomes. A chromosome is a cord of binary digits along with each digit that makes up a chromosome is labeled a gene [27]. This alpha population can be collectively aimless or can be commenced manually applying approaches alike as greedy algorithm. The detail the approaches of GA are obtainable in diversified domains of testing analogous to analysis assessing, minimization of test cases in regression testing, prototype based testing as well as web testing [30].

We approximated the former research question: Are advantages as well as boundaries of benchmarked analyzes along with competence catalogs acclaimed in industry at comprehensive level? In direct to approach the above mentioned analysis gap and research challenge, this analyze brings about the bygone contributions:

(1) Determine a merit as well as confrontations that are based on benchmarked analyzes and competence catalogs in literature;

(2) We arranged contemplate as well as test whether the inspected merits and boundaries are common in business. We additionally exhibit the validity risks as well as counterfeit on the consequences.

2. SYSTEMATIC LITERATURE STUDY

We applied disciplined literature review [4] to determine merits as well as restrictions of automation software testing. Table I materializes the search keywords exercised for name, abstract, as well as keywords to determine journals accompanied to automated software testing that are based on exercised competence. Columns are annexed with AND, along with elements within columns are affixed with OR. The comb was concentrated on the duration 2011-2014 as testing tools have evolved in the last decade additionally dovetailed accrual authoritarian, which would control the approximation with respect to boundaries as well as advantages.

The databases exercised for the search were IEEE Explore, Springer Link, Elsevier, and ACM. The destination approached in a total of 21332 journals. For the determination of analyses the descending steps were abducted(see Table I):

Table –I: PUBLICATION SEARCH RELEVANCE

Criteria	No. of articles (in ! out)
Step 1: Remove redundant documents	21332→15256
Step 2: Analyze relevant domain for 2011 to 2014 publications	15256 → 7451
Step 3: Analyze abstract for relevance of study	7451→ 998
Step 4: Analyze full-publication	998→ 150
Step 5: Note down comparative study points	150 → 30

Initially, equivalent summaries were excluded. Next, the titles of relevant domain were assimilated to choose whether the article concentrates on automation software testing in the context of software engineering. After that, the summaries were determined to assure whether they accommodated the search keywords, had benchmarked competence, and concentrated on the exercise of prioritization approaches, tools, mechanisms, as well as technology for automation software testing. Frequently, it was not expressive from the summary if a paper was communicative, and confronted advantages and boundaries of automation software testing. Analogous papers were not eliminated right away. Instead, the preambles, conclusion, along with full-text of these papers were comprehended, approaching to disagreement of dissociated papers.

Before to analyze determination based on titles along with summaries a test-set of 30 papers was exercised to disclose whether reviewers have the identical affirming of the inclusion and rejection measures. We approximated the Kappa k approximate (a benchmarking for assuring the acceptance between reviewers [5]) as well as apprehended an approximate k=0.725, which connotes considerable acceptance. Because, this defines that the inclusion/exclusion measures were articulately constructed. Consequence, the journals were alienated between authors for each inclusion/exclusion.

The evidence of each paper was amputated exercising a form holding fields for title, author name, year of publication, domain of analysis, research methodology, as well as communicative domain of research (challenges/benefits). The benchmark of the examination was done applying hypothetical benchmark in fusion with narrative abstracts.

2.1 Results

Table- II demonstrates which sources of apotheosis were exercised. It is distinct that the majority of analyzes are of benchmarked description ensuing a research methodology (investigations as well as industrial case), while 8 papers are competence reports.

Table-II PUBLICATION EVIDENCES

Base Approach	References	No. of articles (Total 30)
Research Analysis	[1],[2],[3],[4],[9],[14],[18],[19],[22],[23], [24],[30]	12
CMMI Company Case Studies	[5],[6],[7],[10],[15],[16],[17],[18], [20], [21]	10
Experience oriented Analysis	[8],[11],[12],[13],	8

3. INDUSTRIAL SURVEY

The aim of the practitioner inspect was to identify whether the advantages as well as boundaries are of appropriateness for the software industry. The online survey was allocated through "survey monkey". In addition, the survey was forwarded via e-mail to business contacts. In total, we received 79 precise acknowledgments. Architecture of Online Survey: Before to announcing the survey extensively it was checked by 20 respondents who are well-worn with the area of Automation Software Testing and the survey was updated based on their suggestions to amend its diction. The initial survey consisted demographic inquiries contemplating for the position of the respondent, experience level, business domain type, and development prototype exercised as well as queries were explained exercising advantages and boundaries from the balanced literature outcomes as input. Table –III shows the domain of system and number of responses received as an outcome of initial survey.

Table III DOMAIN RESPONSES TO INITIAL SURVEY

System Domain	Total Responses Received
Financial Systems	12
Educational Portals	15
Research Centers	5
Web Development	15
Real Estate Portals	16
Money Transfer Portals	7
System Development	9

Tables IV affirm the questions asked. The questions are in the form of a principle survey examining the caste of positive responses in counter-reply to acknowledged data. This will give clear direction about need for future development of regression test case prioritization to all researchers involved in similar studies. Whereas Table V gives outline for negative responses and in turn disadvantages of system which will lead us to conclude gross understandings for future development.

TABLES IV: INITIAL SURVEY OUTCOME FOR MERITS OF SYSTEM (For 79 responses).

Rank	**Questions related to benefits**	**Answers on Scale1**					Median
		1	**2**	**3**	**4**	**5**	
M. R.1	Automation Software Testing is a fast testing facility	0 0%	5 6.32%	15 18.98%	9 11.39%	49 62%	4
M. R.2	Automation Software Testing need to put less efforts	3 3.79%	3 3.79%	16 20.25%	12 15.18%	45 56.96%	4
M. R.3	Automation Software Testing can be predefined for similar products /reusable scripts are available	2 2.53%	4 5.06%	15 18.98%	28 35.44%	30 37.97%	3
M. R.4	Automation Software Testing will reduce cost estimation	2 2.53%	1 1.26%	11 13.92%	30 37.97%	35 44.30%	3
M. R.5	Automation Software Testing is easy to carry multiple test simultaneously	0 0	4 5.06%	7 8.86%	19 24.05%	49 62.02%	2
M. R.6	Automation Software Testing will generated accurate result in less time	0 0%	2 2.53%	4 5.06%	21 26.58%	52 65.82%	2
Rank:1=strongly disagree, 2=disagree, 3=neutral, 4=agree, 5= strongly agree							

Tables V: INITIAL SURVEY OUTCOME FOR DEMERITS OF SYSTEM (For 79 responses).

Rank	Questions related to benefits	Answers on Scale1					Median
		1	2	3	4	5	
D. R1	manual testing is better than Automated testing	0	2	8	37	32	3
		0%	2.53%	10.12%	46.83%	40.50%	
D. R2	testing tools available in the market are not universally compatible	0	0	1	51	27	3
		0%	0%	1.26%	64.55%	34.17%	
D. R3	Cost is more for Automated testing than manual testing	0	2	0	18	59	4
		0%	2.53%	0%	22.78%	74.68%	
D. R4	Automated testing require tester with software development background as he/she need to spare time to write scripts prior to testing	8	6	2	25	38	3
		10.12%	7.59%	2.53%	31.64%	48.10%	
D. R5	Test case prioritizations not flexible in case of automation testing	3	10	16	12	38	3
		3.79%	12.65%	20.25%	15.18%	48.10%	

Rank: 1=strongly disagree, 2=disagree, 3=neutral, 4=agree, 5= strongly agree

3.1. Results

3.1.1 Demographics:

The majority of respondents have affirmed quality assurance role, succeeded by programmers conducting testing. Only little other roles, analogous as system architect, system designer, researcher additionally project manager, responded the survey. With respect to collective experience in years we can inspect that the respondents confine a great range of experience, with 46.17% of them claiming lower than 5 years of experience, 24.79% having 5 to 9 years, 11.35% having 10 to 15 years, and 2.67% having more than 15 years.

For domain bifurcation (Table III) we assigned the practitioners to assist numerous acknowledges. The consequences materialize that comprehensive domains are encased, with the majority of them being for Financial Systems, Educational Portals, Research Centers, Web Development, Real Estate Portals, Money Transfer Portals, and System Development. The majority of the respondents used agile software development, followed by waterfall-process/plan driven development. Lean software development is only used by few respondents. Other models were used by rest of the respondents.

3.1.2. Merits:

Table IV shows initial survey outcome for merits of system for 79 responses. As they appeared in the survey, and also includes references to the benefits identified in the systematic literature review to illustrate which benefit in the review led to each question. Furthermore, the answers on a 5-point Likert scale are shown, including the total number of answers and the percentages, as well as the median value. The answers are ranked in ascending order with respect to the sum of the number of responses answering agree (4) and strongly agree (5). Overall, it is visible that the benefits of automation testing software that were found in literature are strongly supported by the respondents, with at least half of them agreeing or fully agreeing to statements.

In the following paragraphs we reflect on each answer, taking the free text answers by the respondents into account.

M.R1Overall, 62%% of the respondents choose to completely agree or agree on that benefit of high reusability of test cases. In the free text answers it was highlighted that the success is conditional upon right test strategies, which e.g. relate to challenges such as skills of testers with respect to technical and tool knowledge, as well as investment in tools (see Table IV).

M.R2: This question is relatively analogous to the prior one (M.R1), however contemplates additional precisely about repeatability in order to rescuing time. In aggregate 56.96% accept or entirely accept on that advantage. Furthermore, it was highlighted that this should not be the goal for testing. The impetus should be to "rather to conduct gainful benchmarking in the duration allotted by the project stakeholders".

M.R3: 37.97% accept that Automation Software Testing can be predefined for similar products /reusable scripts are available. Practitioners ascribed caveat of how gainful test case coverage should be comprehended in this case.

M.R4: This question adheres to re-running benchmarks and with that maintaining duration and expense along with that in parallel to manual benchmarking, which 44.30% of the respondents accept with. Here it was beckoned out that if the equivalent conditions are commenced it would be applied to rerun the equivalent benchmark case, furthermore, practitioners conducting manual benchmarking can apply their competence as well as cognizance of alterations to think over benchmarking performance. Additionally, distinguishing castes of benchmarks accumulate longer duration to automate, and there communal expediency is more beneficial, one respondent remarking that, test cases conducted many times maintain duration as well as expense, although several castes of automated benchmarks acquire much more time than manual.

M.R5: Automation Software Testing is easy to carry multiple test simultaneously. Also the collaboration of Automation Software Testing with higher contemplation in product degree as well as the adeptness of running into deadlines was accepted on by 62.02% of the respondents.

M.R6: Automation Software Testing will generate accurate result in less time. This advantage adheres to that application-specific test case infrastructure assists to alleviate the exercise that prioritization mandates from testers. In total 65.82% acceded to that, while no definite acknowledgments were assisted by the practitioners.

3.1.3 Demerits:

The restrictions are exhibited in the identical manner as the advantages (see Table V). The table materializes that the restrictions are also firmly asserted by practitioners.

D.R1: The original question adheres to manual testing is better than Automated testing and alpha expenditure expected in automated benchmarking as well as that it compels duration to mature until advantages are beholden. The boundary is determined by 40.50% of the respondents. In connection to manual benchmarking the investment in automated benchmarking is considered as beneficial, as one respondent betokens out "Manual benchmarking, if consistent again and again, is a very big loss of cost. Automated benchmarking is an investment, as well as obligates only duration to be far more beneficial than manual regression tests".

D.R2: Automation testing tools available in the market are not universally compatible and initially, the expense for same is higher as well as needs maturation. This acknowledgment determines authoritative affirm by 34.17% of the respondents. In fact, one respondent acknowledges that "Test case Prioritization desires at least as much interminability as the developed software with respects to the Technical Debt". In order to bypass some of the alpha expenses, one respondent cues out that "If you consider earlier you can commence manual test scripts that can also be applied for automated benchmarking, since allow duration needed for conversion".

D.R3: Cost is more for Automated testing than manual testing and with respect to beneficial prioritization and needed skills 81% affirm to those confrontations.

D.R4: 74.68% respondents were strongly agree that, Automated testing require tester with software development background as he/she need to spare time to write scripts prior to testing which increases cost and efforts.

D.R5: The major outcome regarding this demerit is those test case prioritizations not flexible in case of automation testing as per 48.10% respondents.

4. PRIORITIZATION PIT FALLS

Contributed that the literature examine is a matter of to analysis in all its ways there is a chance of deviation. To alleviate this alarm we examined as well as filtered our criteria for contemplate determination and only when adhering a high level of acceptance alienated the work to make sure that everyone has a allegorical willing. All borderline cases where there was a waver in decision and determination was conversed to further alleviate motivation. Overall, this risk is still existent, furthermore, efforts have been acquired to alleviate deviation in analyze decision.

In scrutinize there is a threat that the questions might be confused. Hence, the study was benchmarked for comprehensibility prior to driving it communal to a larger populace. But, challenges might still defect extent for interpretation, which could not be absolutely ruled out.

Another average threat in examines is that they are deviated towards a definite population. This risk is fractionally under control as counterclaims came from a dissimilarity of areas, although the web domain was articulately the domain with the apex count of counterclaims. The core risk in this examine is that there was no conflict driven between approach of prioritization. In effect, it is unknown whether the results are lead toward one of the approaches.

5. CONCLUSION

This paper drives trident presents. Alpha, we conducted a methodical review of software test case prioritization advantages as well as boundaries in academic literature. We accumulated 21332papers, which were lessened to 30 analysis works (see Table II). Thus, the number of apotheosis on these factors is quite superficial as ample advantages as well as boundaries are assisted up by only one or two grounds. Additionally, we judged that while advantages always came from stronger drives of evidence, boundaries were accrual commonly accounted on competence reports. We account that this is driven by journal influence revering the advantages. We assume that authoritarian further work on this domain is to approximate the boundaries of test case Prioritization with conscientious benchmarked analyzes, i.e. case studies and experiments.

Second, we done examine of the practitioners' belief of software test case prioritization gains along with extremities. The results displayed that the focal advantages of test case prioritization are reusability, repeatability along with performance maintained in test case accomplishments. These results acknowledge the dominance of test case prioritization when numerous regressions testing rounds are expected. Additionally, the practitioners convince that prioritization alters test case coverage, which defines that prioritization has advantages even when extreme regression testing is not expected. Deferring the boundaries, we determined that prioritization discovers a high alpha expense in predetermining the test cases, accumulating a test case prioritization tool, and training the people. Non-surprisingly, the extension of automated test cases was also considered as problematic. Additionally 56.96% of the practitioners appraise that current test case prioritization tools allow a destitute advantageous for their desires.

The limitations of test case prioritization considered by the practitioners should delineate authoritarian future research courses. In this survey, numerous reference papers of GA in asymmetric castes of software benchmarking are examined. The GA is also applied with fuzzy as well as in the neural networks in many types of benchmarking. It is determined that by applying GA, the results as well as the performance of testing can be elaborated. Our future research will comprise addressing GA for regression testing in web based approaches. In future, we plan to apply GA along with other computing techniques like web technologies or neural networks for regression test prioritization. We will also plan to use GA in integration testing for finding optimal test order.

6. REFERENCES:

[1] Yoo, Shin, and Mark Harman. "Regression testing minimization, selection and prioritization: a survey." *Software Testing, Verification and Reliability* 22.2 (2012): 67-120.

[2] Bryce, Renee C., Sreedevi Sampath, and Atif M. Memon. "Developing a single model and test prioritization strategies for event-driven software." *Software Engineering, IEEE Transactions on* 37.1 (2011): 48-64.

[3] Nie, Changhai, and Hareton Leung. "A survey of combinatorial testing." *ACM Computing Surveys (CSUR)* 43.2 (2011): 11.

[4] Jia, Yue, and Mark Harman. "An analysis and survey of the development of mutation testing." *Software Engineering, IEEE Transactions on* 37.5 (2011): 649-678.

[5] Carlson, Ryan, Hyunsook Do, and Anne Denton. "A clustering approach to improving test case prioritization: An industrial case study." *Software Maintenance (ICSM), 2011 27th IEEE International Conference on*. IEEE, 2011.

[6] Cartaxo, Emanuela G., Patrícia DL Machado, and Francisco G. Oliveira Neto. "On the use of a similarity function for test case

selection in the context of model-based testing." *Software Testing, Verification and Reliability* 21.2 (2011): 75-100.

[7] Kaur, Arvinder, and Shubhra Goyal. "A GENETIC ALGORITHM FOR REGRESSION TEST CASE PRIORITIZATION USING CODE COVERAGE." *International Journal on Computer Science & Engineering* 3.5 (2011).

[8] Kaur, Arvinder, and Shivangi Goyal. "A Bee Colony Optimization Algorithm for Fault Coverage Based Regression Test Suite Prioritization." *International Journal of Advanced Science & Technology* 29 (2011).

[9] Carlson, Ryan, Hyunsook Do, and Anne Denton. "A clustering approach to improving test case prioritization: An industrial case study." *Software Maintenance (ICSM), 2011 27th IEEE International Conference on*. IEEE, 2011.

[10] Nie, Changhai, and Hareton Leung. "A survey of combinatorial testing." *ACM Computing Surveys (CSUR)* 43.2 (2011): 11.

[11] Engström, E., Per Runeson, and Andreas Ljung. "Improving Regression Testing Transparency and Efficiency with History-Based Prioritization--An Industrial Case Study." *Software Testing, Verification and Validation (ICST), 2011 IEEE Fourth International Conference on*. IEEE, 2011.

[12] Thomas, Stephen W., et al. "Static test case prioritization using topic models." *Empirical Software Engineering* 19.1 (2014): 182-212.

[13] Anwar, Zeeshan, and Ali Ahsan. "Exploration and analysis of regression test suite optimization." *ACM SIGSOFT Software Engineering Notes* 39.1 (2014): 1-5.

[14] Mei, Lijun, et al. "Test Pair Selection for Test Case Prioritization in Regression Testing for WS-BPEL Programs." *International Journal of Web Services Research (IJWSR)* 10.1 (2013): 73-102.

[15] Jacob, T. Prem, and T. Ravi. "A Novel Approach for Test Suite Prioritization using Clustering." *Indian Journal of Advances in Computer Sciences and Technology* 1.1 (2013): 59-66.

[16] Pravin, Albert, And Subramaniam Srinivasan. "Effective Test Case Selection And Prioritization In Regression Testing." *Journal Of Computer Science* 9.5 (2013).

[17] Zhang, Lingming, et al. "Bridging the gap between the total and additional test-case prioritization strategies." *Software Engineering (ICSE), 2013 35th International Conference on*. IEEE, 2013.

[18] Anand, Saswat, et al. "An orchestrated survey of methodologies for automated software test case generation." *Journal of Systems and Software* 86.8 (2013): 1978-2001.

[19] Sharma, Chayanika, Sangeeta Sabharwal, and Ritu Sibal. "A Survey on Software Testing Techniques using Genetic Algorithm." *International Journal of Computer Science Issues (IJCSI)* 10.1 (2013).

[20] Ouriques, Joao Felipe S., Emanuela G. Cartaxo, and Patricia DL Machado. "On the Influence of Model Structure and Test Case Profile on the Prioritization of Test Cases in the Context of Model-Based Testing." *Software Engineering (SBES), 2013 27th Brazilian Symposium on*. IEEE, 2013.

[21] Zhu, Hong, et al. "Toward a mature industrial practice of software test automation." *Software Quality Journal* 22.2 (2014): 239-240.

[22] Jia, Yue, and Mark Harman. "An analysis and survey of the development of mutation testing." *Software Engineering, IEEE Transactions on* 37.5 (2011): 649-678.

[23] Kaushik, Nilam, et al. "Dynamic prioritization in regression testing." *Software Testing, Verification and Validation Workshops (ICSTW), 2011 IEEE Fourth International Conference on*. IEEE, 2011.

[24] Li, Bixin, et al. "Automatic test case selection for regression testing of composite service based on extensible BPEL flow graph." *Journal of Systems and Software* 85.6 (2012): 1300-1324.

[25] Czerwonka, Jacek, et al. "Crane: Failure prediction, change analysis and test prioritization in practice--experiences from windows." *Software Testing, Verification and Validation (ICST), 2011 IEEE Fourth International Conference on*. IEEE, 2011.

[26] Rogstad, Erik, et al. "Industrial experiences with automated regression testing of a legacy database application." *Software Maintenance (ICSM), 2011 27th IEEE International Conference on*. IEEE, 2011.

[27] Mei, Lijun, et al. "XML-manipulating test case prioritization for XML-manipulating services." *Journal of Systems and Software* 84.4 (2011): 603-619.

[28] Thomas, Stephen W., et al. "Static test case prioritization using topic models." *Empirical Software Engineering* 19.1 (2014): 182-212.

[29] Ståhl, Daniel, and Jan Bosch. "Modeling continuous integration practice differences in industry software development." *Journal of Systems and Software* 87 (2014): 48-59.

[30] Vats, Prashant, Manju Mandot, and Anjana Gosain. "A comparative study of Genetic Algorithms for its applications in Object oriented testing." *Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on*. IEEE, 2014.