

Escalated Methods for Software Defects Audit in Repercussion and Effects Construe to Nature Inspired and Behavior Driven Mechanisms

Alankrita Aggarwal, Kanwalvir Singh Dhindsa, PK Suri



Abstract: *In the current scenario, enormous software suites are available but there is need to integrate the higher degree of robustness. This is in order to develop the full proof software application with the higher degree of performance and accuracy with minimum scaling of the error factors. There is need to work on the software audit because of the enormous assaults and vulnerabilities from different sources. The approach of software audit and management strategies can be elevated using Nature Inspired Approaches and DevOps based integrations. This manuscript is having the cavernous evaluation of different state of the art approaches for software defects management using the patterns of nature inspired and behavior driven mechanisms and found that the effective results and outcomes can be achieved using such approaches. The presented work is presenting the usage of behavior driven mechanism because of the key requirements to have the multi-dimensional evaluation of the software products.*

Keywords: *Nature Inspired Approach, Software Audit, Software Defects*

I. INTRODUCTION

Software testing is a procedure, to assess the usefulness of a software application with a purpose to discover whether the created software met the predetermined prerequisites or not and to distinguish the defects to guarantee that the item is sans defect so as to deliver the quality item [1, 2]. As per ANSI/IEEE 1059 standard it is a procedure of breaking down a software thing to distinguish the contrasts among existing and required conditions (i.e., defects) and to assess the highlights of the software thing [3, 4].

Software testing can be expressed as the way toward checking and approving that a software or application is sans bug, meets the specialized prerequisites as guided by it's structure and development and meets the client necessities adequately and effectively with taking care of all the uncommon and limit cases [5].

The procedure of software testing points not just at discovering deficiencies in the current software yet in addition at discovering measures to improve the software as far as productivity, precision and convenience [6, 7, 8]. It principally goes for estimating detail, usefulness and execution of a software program or application. The Defect is a mistake happened in the software any surprising things accompanies the software is known as a defect. No software exists without defect or any bugs [9, 10].

You can't evacuate the defect forever however you can lessen the quantity of defects. Defect the executives is a procedure to distinguish the defect of the software [11, 12]. The Development group needs a defect the board device with the goal that they can discover defect effectively and at an in all respects beginning period of procedure in light of the fact that when the defect is distinguished the expense of fixing it will be low, however in the event that the defect is recognized in the later phase of development process, at that point the expense of fixing that defect will be more [13, 14].

Defect the board will work like a spine to building up a group in finding the defect in the beginning time in an exceptionally simple manner. The little interest in defect the executives apparatus will give you the greatest return [15, 16, 17]. Defect the executives works in the parallel method for software development process. Software testing group discovers every one of the bugs and defect and report to building up a group to fix it. Defect the board, records all the defect of the software these records can be seen later also in the event that you need to audit or need to watch that how you have fixed that. These records can be utilized the information base for the future reference [18, 19].

II. USAGE OF DEVOPS FOR SOFTWARE DEFECTS AND AUDIT

From last decade, the ways to develop, deploy, test and rebuild the software applications are changed a lot. The traditional software development processes are now elevating with high performance practices with the use of "DevOps". This term "DevOps" come across many times in different implementations and there is need to understand it in easy terms [20, 21, 22].

In one way, DevOps is considered as a software development method while others relate it to the tools and technologies for continuous delivery and configuration management [8].

Revised Manuscript Received on October 30, 2019.

* Correspondence Author

Alankrita Aggarwal*, PhD Research Scholar, Department of Computer Science and Engineering, IKGPTU, Kapurthala, Punjab- 144603 India, alankrita.agg@gmail.com

Kanwalvir Singh Dhindsa, Department of Computer Science and Engineering, Baba Banda Singh Bahadur Engineering College, Fatehgarh Sahib, Punjab-140406, India, kdhindsa@gmail.com

PK Suri, Department of Computer Science & Applications, Kurukshetra University, Kurukshetra-136038 Haryana, India, pksurikuk@gmail.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

Escalated Methods for Software Defects Audit in Repercussion and Effects Construe to Nature Inspired and Behavior Driven Mechanisms

In general, DevOps is an initiative or movement by which the phases of software development as well as the involvement of professionals are integrated with the real time collaboration and communication with the version control [9]. One of the key goals of using DevOps is to strengthen the link between software development process (Dev) and operations of Information Technology (Ops) so that the complexities and timelines with overall system development life cycle can be reduced [23].

There are number of open source tools and frameworks available for DevOps so that the process of development, deployment, testing, configuration management and redeployment can be done.

Following are the prominent tools and libraries which are used in association with the features of DevOps

- Behat: Test Automation
- Watir: Testing Tool
- Supergiant: Container Management
- Ansible: I.T. Operations including Cloud Provisioning, Deployment, Configuration Management
- Nagios: Infrastructure Monitoring
- SaltStack: Even Driven Orchestration, Configuration Management
- Chef: Cloud Computing
- Docker: Portability
- Git: Source Code Management
- Puppet: Software Deployment with Reliability, Audit and Agility
- Hudson: Testing

III. NATURE ISPIRED APPROACH ASSOCIATED BEHAVIOUR DRIVEN DEVELOPMENT (BDD)

In actual implementations, the BDD refers to a development approach rather than a traditional software tool and it is having the base of Test Driven Development (TDD) and it makes use of Nature Inspired Mechanisms for the automation of software defects mechanisms.

The classical Test Driven Development (TDD) describes the way in which the specific software performs different functions [11]. BDD describes the detailed view and process by which the end user makes use of the specific software application [24]. In addition, the BDD based approach enable the higher level of communication and collaboration in the team members of development. BDD based approach is more diversified and focused towards the behavior of the system to present the detailed view.

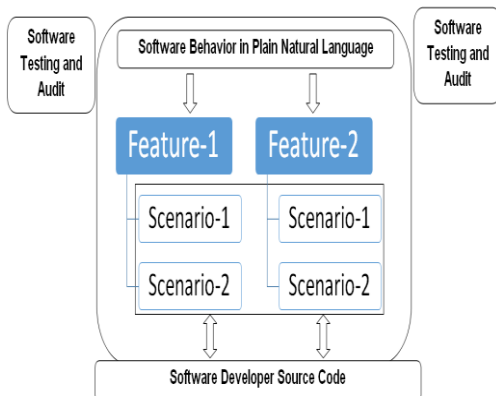


Figure 1: Behavior Driven Development (BDD) for Software Defects Audit [11]

In BDD, the story framework is implemented to describe the detailed architecture and flow of the process to be followed.

In story framework, there are three key components

Role: User ("As a")

Feature: Requirements ("I want")

Benefit: Outcome ("so that")

Behat is one of the powerful and multi-featured frameworks with Behavior-Driven Development (BDD) strategy. It is developed in PHP and supports the syntax structure like traditional sentences in English language so that the process of test case writing will be easy and understandable for all.

Behat is distributed under Free and Open Source with huge functionalities and flexibilities to customize for higher degree of accuracy and performance in the test cases. Using Composer is the official method to install Behat. Composer refers to the package manager for PHP. In addition to the fresh installation of Behat, the new released version can be updated using this approach.

This is another method to install Behat. In this download the behat.phar file from <https://github.com/Behat/Behat/releases>. After downloading, the behat.phar file is moved to the Project Directory and then the following command can be used to check the installed version of Behat

```
$ php behat.phar -V
```

Here, .phar file refers PHP Archive. It is a package format so that the bundle of files and libraries can be created and distributed. Basic Format of the Scenario is having the following format in which the scenarios are written in the natural language.

To work with different examples of BDD, the following scenarios can be analyzed:

IV. IMPLEMENTATION SCENARIO1: INTERNET SEARCHING

If a user is willing to search the tutorials on "Blockchain Programming" from Internet, then following key points will be used:

1. Internet should be connected
2. Open Web Browser
3. Open a Search Engine
4. Enter the search string or keyword "Blockchain Programming"
5. View results

This process can be followed using Behat as:

Feature: Blockchain Programming

In order to see tutorials

As a researcher

I need to be able to get the tutorials and programming source codes of Blockchain Programming

The above mentioned is the very basic scenario of using Behat framework for the behavior driven development programming in which the feature and its required aspects are specified.

V. IMPLEMENTATION SCENARIO 2: ONLINE SEARCH AND SHOPPING FROM E-COMMERCE PORTAL

In this example, the scenario to search a suitable smart phone from an E-commerce web portal is mentioned. There are certain conditions and rules which are required to be met.

A. Feature: Smartphone Search and Online Order

Search the Specific Model

Integrate the Results with HTTP Client

Analytics of the Results

Thresholding the Outcomes

B. Rules:

- Tax is 10%
- Courier Delivery Charge for Shopping Cart under INR 20000 is INR 100
- Courier Delivery Charge for Shopping Cart over INR 20000 is INR 200

C. Scenario: Buying a single smart phone under INR 20000

Given there is a "Mobile Cover", which costs INR 100
When I add the "Mobile Cover" to the Shopping Cart
Then I should have 1 product in the Shopping Cart
And the overall Shopping Cart price should be under INR 23000

D. Scenario: Buying a single smart phone over INR 20000

Given there is a "Mobile Cover", which costs INR 100
And I add the "Power Bank", which costs INR 5000
When I add the "Mobile Cover" to the Shopping Cart
And I add the "Power Bank", which costs INR 5000
Then I should have 1 product in the Shopping Cart
And the overall Shopping Cart price should be under INR 30000

Once the instructions are written, these scenarios are executed using the following inside the Project Directory
`$ /bin/behav`

On running, the source code will be associated and tested with the test cases and the test scenarios will be displayed in terms of failed or passed.

For example, if amazon.in is the website mentioned in the source code for test automation, the website will be scanned with the mentioned conditions of price, shopping cart and other parameters in the scenario so that the user can get the accurate output. In addition, this approach gives the software quality assurance team a detailed view of the outputs which can be generated using different phrases and idioms mentioned in the scenarios of the test project.

VI. IMPLEMENTATION WITH LOCUST FOR NATURE INSPIRED MECHANISM FOR LOAD TESTING

The load testing and associated stress testing can be done with the usage of Locust that is the key tool for the analytics of the patterns and to have the greater performance. Locust is having the plain code without any complex settings. The easiest way to install Locust is from PyPI, using pip:

```
> pip install locustio
```

Figure 2: The View of locustfile.py and Instruction to Run

The code of Python locustfile.py can be directly executed in the command prompt to test the web application.

Figure 3: Executing Locust Code for Testing the Web Application

A. Key Features of Locust

- Open Source and License Free Distribution
- Testing and Audit on different aspects
- Multi-Dimensional Software Audit

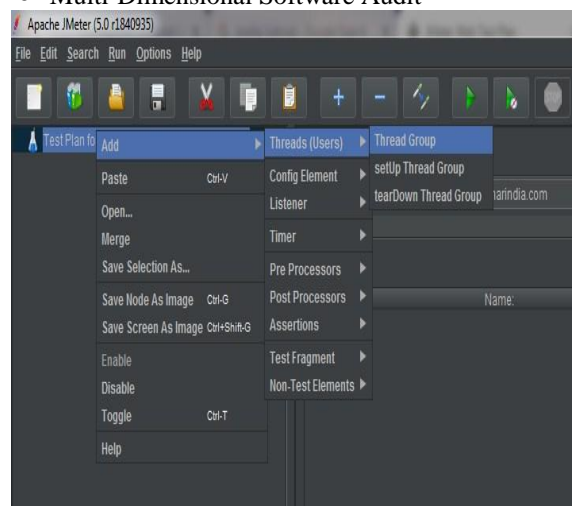
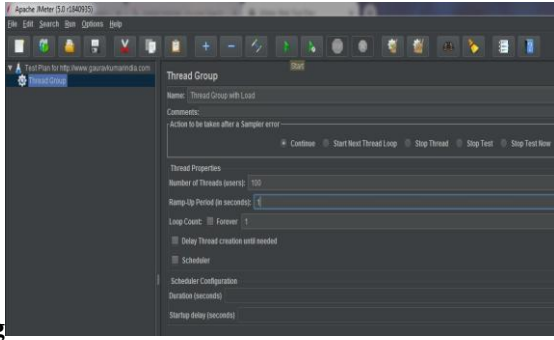


Figure 4: Creating a Thread Group for Load



Testing
Figure 5: Specifying Threads (Users) in Apache JMeter

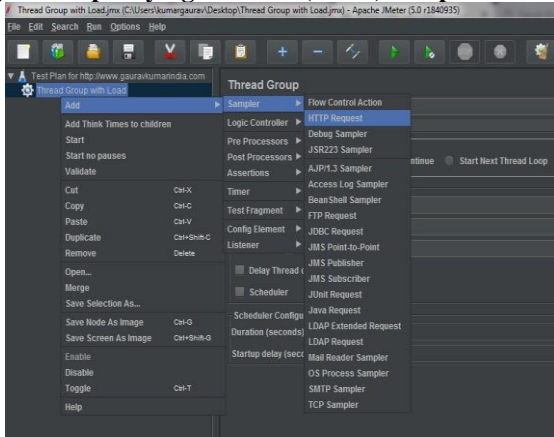


Figure 6: Generation of HTTP Requests for Generation of Load

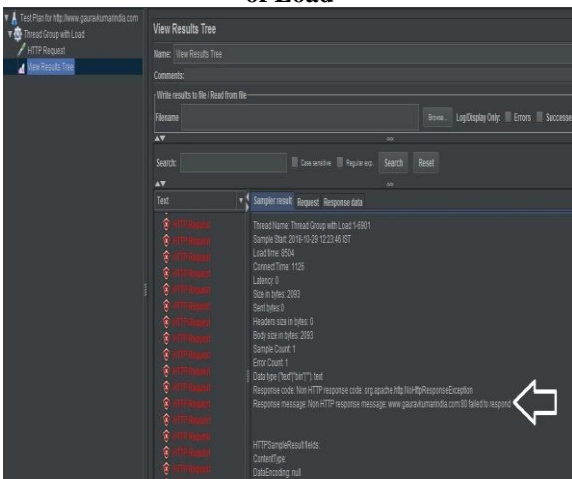


Figure 7: Examination of Logs and Response from Server

A. Methodology to Fetch the Outcomes

1. Identification of the Software Suite or Product under audit and rigorous testing
2. Framing the test cases
3. Mapping of Test Case to different test runs
 - a. HULK
 - b. DDoS
 - c. Stress
4. Presenting the outcomes on Results with different parameters
5. Fetching the Simulation Patterns
6. Logs Analysis and Reports Presentations

VII. HTTP UNBEARABLE LOAD KING (HULK) AUTOMATION FOR LOAD TESTING USING NATURE INSPIRED SWARM APPROACH

Table 1: Implementation Aspects with the Software Testing

Implementation Scenario	Performance of Non NIA	NIA Integrated Approach
1	85	92
2	83	92
3	87	96
4	82	92
5	86	92

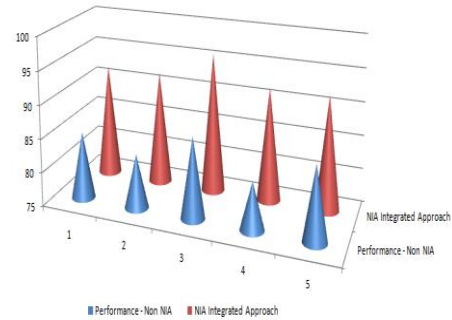


Figure 8: Performance Evaluation

The performance evaluation is quite effectual in case of Nature Inspired Approach and the simulated patterns for the cumulative outcomes.

Table 2: Implementation Scenarios with the Outcomes

Implementation Scenario	Accuracy - Non NIA	NIA Integrated Approach
1	80	95
2	82	95
3	86	96
4	86	93
5	86	96

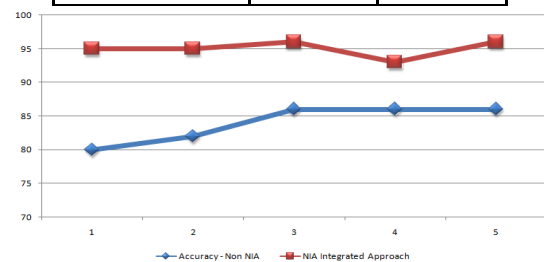


Figure 8: Evaluation of Accuracy between nature inspired and non nature inspired

The results and test case simulations are giving the outcomes in favor of NI based approach and giving the better results as compared to the traditional approach.

VIII. CONCLUSION

With the increasing usage of software products for different applications, there is need to work on the software audit for multiple assaults and testing strategies. A number of test automation libraries and frameworks are available in Free and Open Source Distribution with their own features and set of testing strategies.

The performance of these tools can be compared with the pragmatic results on same type of scenarios so that the detailed view of their efficiency and assessment can be generated. In the present situation, huge programming suites are accessible however there is have to incorporate the higher level of heartiness. This is so as to build up the full confirmation programming application with the higher level of execution and exactness with least scaling of the mistake factors. The methodology of programming review and the board systems can be raised utilizing Nature Inspired Approaches and DevOps based mixes. This work is having the huge assessment of various best in class approaches for programming deserts the executives utilizing the examples of nature motivated and conduct driven components and found that the successful outcomes and results can be accomplished utilizing such methodologies.

ACKNOWLEDGMENT

The authors are thankful to Inder Kumar Gujral Punjab technical University, Kapurthala Punjab -144603 (India) to facilitate us to pursue this research. Our sincere thank also goes to the anonymous reviewers for their valuable remarks.

REFERENCES

1. Myers, G. J., Badgett, T., Thomas, T. M., & Sandler, C. (2004). The art of software testing (Vol. 2). Chichester: John Wiley & Sons.
2. Beizer, B. (2003). Software testing techniques. Dreamtech Press.
3. Hetzel, W. C., & Hetzel, B. (1988). The complete guide to software testing (pp. 201-202). Wellesley, MA: QED Information Sciences.
4. Kuhn, D. R., Wallace, D. R., & Gallo, A. M. (2004). Software fault interactions and implications for software testing. IEEE transactions on software engineering, 30(6), 418-421.
5. Bertolino, A. (2007, May). Software testing research: Achievements, challenges, dreams. In 2007 Future of Software Engineering (pp. 85-103). IEEE Computer Society.
6. Bass, L., Weber, I., & Zhu, L. (2015). DevOps: A software architect's perspective. Addison-Wesley Professional.
7. Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2015, May). Dimensions of devops. In International conference on agile software development (pp. 212-217). Springer, Cham.
8. Jabbari, R., bin Ali, N., Petersen, K., & Tanveer, B. (2016, May). What is devops: A systematic mapping study on definitions and practices. In Proceedings of the Scientific Workshop Proceedings of XP2016 (p. 12). ACM.
9. Virmani, M. (2015, May). Understanding DevOps & bridging the gap from continuous integration to continuous delivery. In Fifth International Conference on the Innovative Computing Technology (INTECH 2015) (pp. 78-82). IEEE.
10. Cois, C. A., Yankel, J., & Connell, A. (2014, October). Modern DevOps: Optimizing software development through effective system interactions. In 2014 IEEE International Professional Communication Conference (IPCC) (pp. 1-7). IEEE.
11. Solis, C., & Wang, X. (2011, August). A study of the characteristics of behaviour driven development. In 2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications (pp. 383-387). IEEE.
12. Soeken, M., Wille, R., & Drechsler, R. (2012, May). Assisted behavior driven development using natural language processing. In International Conference on Modelling Techniques and Tools for Computer Performance Evaluation (pp. 269-287). Springer, Berlin, Heidelberg.
13. Rahman, M., & Gao, J. (2015, March). A reusable automated acceptance testing architecture for microservices in behavior-driven development. In 2015 IEEE Symposium on Service-Oriented System Engineering (pp. 321-325). IEEE.
14. de Carvalho, R. A., & Manhaes, R. S. (2010). Mapping business process modeling constructs to behavior driven development ubiquitous language. arXiv preprint arXiv:1006.4892.
15. Tavares, H. L., Rezende, G. G., Santos, V. M. D., Manhaes, R. S., & de Carvalho, R. A. (2010). A tool stack for implementing Behaviour-Driven Development in Python Language. arXiv preprint arXiv:1007.1722.

16. Sherman, Y., Hare, U., & Kinreich, I. (2002). U.S. Patent No. 6,434,513. Washington, DC: U.S. Patent and Trademark Office.
17. Barber, S. (2004, September). Creating effective load models for performance testing with incomplete empirical data. In Proceedings. Sixth IEEE International Workshop on Web Site Evolution (pp. 51-59). IEEE.
18. Menascé, D. A. (2002). Load testing of web sites. IEEE Internet Computing, 6(4), 70-74.
19. Malik, H., Hemmati, H., & Hassan, A. E. (2013, May). Automatic detection of performance deviations in the load testing of large scale systems. In Proceedings of the 2013 International Conference on Software Engineering (pp. 1012-1021). IEEE Press.
20. Jiang, Z. M. (2010, July). Automated analysis of load testing results. In Proceedings of the 19th international symposium on Software testing and analysis (pp. 143-146). ACM.
21. Mittal, Lalit Mohan Goyal, Sumit Kaur, Iqbaldeep Kaur, Amit Verma, D. Jude Mamta Hemanth, "Performance Enhanced Growing Convolutional Neural Network Based Approach for Brain Tumor Segmentation in Magnetic Resonance Brain Images", Applied Soft Computing, (SCIE Indexed journal), 4.004 (Second Revision Submitted)
22. Mamta Mittal, Lalit Mohan Goyal, Sumit Kaur, Iqbaldeep Kaur, Amit Verma, D. Jude Hemanth, (2019), "Deep learning based enhanced tumor segmentation approach for MR brain images", Applied Soft Computing, Vol 78, pp. 346-354
23. Mamta Mittal, Amit Verma, Iqbaldeep Kaur, Bhavneet Kaur, Meenakshi Sharma, Lalit Mohan Goyal, Sudipta Roy & Tai-hoon

AUTHORS PROFILE



Alankrita Aggarwal is a PhD research scholar from IK Gujral Punjab Technical University Jalandhar-144603, Punjab (India). She is M.tech and BE in Computer science and engineering from Maharishi Dayanand University ,Rohtak (India) having a rich experience approx 15 years of teaching as well as research experience .Her areas of interest are soft computing, software engineering and neural networks. She has published 22 research papers in reputed International journals and guided more than 12 M.tech dissertations. She also worked as the organizing secretary and coordinator of various Workshops. She is a Member of International Associations of Engineers (IAENG) also lifetime member of ISTE.



Dr. Kanwalvir Singh Dhindsa is professor in computer science and engineering department at Baba Banda Singh Bahadur Engineering College, Fatehgarh Sahib-140407,Punjab(India) affiliated to IKGPTU Jalandhar Punjab (India).He is having more than 20 years of Industrial ,teaching as well as research experience. He has published research papers in reputed International journals and guided Phd as well as M.tech students. His areas of interest are cloud computing, Big data, IoT, Mobile Computing, Database & security, Web Engineering. He is also member of Computer Society of India (CSI), Indian Society of Technical education (ISTE), Institution of Engineers (IEI).He is also convener of various national as well international conferences and workshops.



Dr. P.K.Suri is a former Professor, Dean Academic and Chairman of department of computer science and applications, Kurukshetra University, Kurukshetra-136119,Haryana,India and HCTM Technical Campus, Kaithal (Haryana), India. He has forty years of teaching and research experience with various designations in both the institutes. He received his MSc Degree from IIT Roorkee (formerly known as University of Roorkee) in the year 1972.He has completed hid Phd Degree at Faculty of Engineering, Kurukshetra University ,Kurukshetra in year 1981.His research interests include simulation, cloud computing, Adhoc networks, wireless networks, distributed systems, software engineering. He has attended number of national and international conferences and published a number of research papers in national and international journals. He has guided more than 20 Phd Research scholars.