Indian Journal of Science and Technology, Vol 9(48), DOI: 10.17485/ijst/2016/v9i48/103784, December 2016
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

The Factors Driving Testing in DevOps Setting- A Systematic Literature Survey

Jayasri Angara[1*], Srinivas Prasad[2] and Gutta Sridevi[1]

[1]K. L. University, Vijayawada - 522502, Andhra Pradesh, India; angara.jayasri@gmail.com, sridevi.gutta2012@gmail.com
[2]GMR Institution of Technology (A), Rajam - 532127, Andhra Pradesh, India; srinivas_prasad@hotmail.com

Abstract

Objectives: Software testing is no more "Essential" requirement but it is "Critical" for the software application or product survival. It has become mental discipline and the tone of an organization mindset especially in the context of emerging DevOps practice. A few limitations of agile methodology from operational and business readiness perspective are addressed in DevOps. DevOps got emerged out of continuous software delivery which captures the market opportunities and reduces the feedback time. However, the new DevOps processes are influencing profoundly on QA and Testing functions. The present study is to understand different motivational factors driving testing in DevOps Environment and attempts to identify key technical, cultural and managerial factors of testing in DevOps setting. **Methods/Statistical Analysis**: We conducted extensive literature survey on academic and industry reports in the context of DevOps testing. Finally we screened 29 most relevant articles out of 295 found among 5 major journal databases. Further we extended manual survey including whitepapers and industry reports. **Findings**: It is evident that most of the articles connected to DevOps testing got published after year 2011 and surged after 2014. This phenomenon shows the nascence of the research progress in this domain. We observed that DevOps in testing is closely associated with automation of test cases. In its advanced stage, it is attributed to auto generation of test cases through model driven frameworks. This has been stressed in more than 50% of review articles. More than 50% of articles connected to cloud, virtualization, simulation techniques of DevOps Testing. The factors like agility, scale, metric driven process, reduction of complexity and cost appeared in more than 16 review articles. DevOps demands alternative metrics for better collaboration and communication between various stakeholders of the system. DevOps testing is face of organization culture and human resource mindset. This has been stressed in most of the review articles. **Application/Improvements:** This research is constrained from couple of biases (authority and publication). Research valued opinions of the other researchers. The search string used (DevOps Testing) may have multiple synonyms but we de-risked this threat by manually verifying each and every result. Further we extended manual survey using this search string.

Keywords: Agile Testing, Continuous Integration, Continuous Testing, DevOps Testing

1. Introduction

The evolution of software development lifecycle (waterfall methodology to V-Model to agile of contemporary times) has brought change in the way testing is performed. Traditionally in waterfall lifecycle model, test execution is done after coding. Of late, the importance of lean, time to market led to introduction of agile process. Though agile process is producing working functionality at the end of every sprint (cycle), completed functionality would have to wait till the release date when operations team integrates it. Operations team has challenges in integrating/deploying continuous release cycles. On top it, operations team face challenges due to manual deployments, inconsistent environments (Development vs. Production), complex infrastructure issues, etc. Business teams are having challenges in going early live. The key objective of the agile which is shorter time to market is not fully realized[1]. In the recent times, industry is adopting DevOps practice to overcome this challenge. DevOps is a cross-disciplinary

*Author for correspondence

practice, promotes set of processes and methods to enhance communication and collaboration between business, development, testing and operations teams. The critical objective is to establish cultural change and collaboration between all stakeholders of delivery pipeline[2]. Among all stakeholders of software system, the role of testing becoming critical in the context of DevOps. DevOps allow continuous releases (continuous testing) without compromising quality and speed. Hence it necessitates the study of testing adoption in DevOps environment which helps in defining new test strategies, frameworks, metrics, etc.

The objective of this paper is to identify key technical, cultural and managerial factors of testing in DevOps setting. The present study addresses critical research question. What are the different motivational factors driving testing in DevOps Environment? It has been answered conducting literature survey on academic and industry reports. Section 2 describes research approach. Section 3 details our findings of the systematic literature study. Section 4 presents discussion and threats to validity. Section 5 summarizes our contribution.

2. The Research Approach

2.1 Research Overview

We resorted to non-experimental methods like content analysis in the form of observation and analysis of existing data sets to identify critical factors driving testing in DevOps ecosystem. These are presented in the Section 3. Content analysis method gives wide perspective of research direction in systematic way[3]. Among five types of texts available in content analysis method (written text, oral text, iconic text, audio-visual text, hypertexts), this study focuses on written and hypertexts available in books, journal papers and web sources. It is the study of mute evidence of texts[4]. We developed codebook with categories and sub categories classified in a staggered approach. We made sure reliability of the coding standards and mapping process with proper checks and balances. This codebook is used for analysis, interpretations and deriving key drivers of DevOps testing. Review protocols are defined and guidelines are adhered as per Cochrane Handbook for Systematic Reviews[5]. Codebook has been divided into 6 categories article description (year, name of journal or conference, name of the title, volume no, DOI), author key words, SDLC Phase associated to, relevancy with testing phase, key terms associated with testing and abstract[6].

2.2 Data Collection Process

The key term search process[6,7] has been adapted to databases of IEEE Xplore, ACM Digital Library, Science Direct, Springer, Wiley InterScience. We used search terms- "DevOps" And "Testing". The search string applied to Full Text and Metadata. The search process was done during the month of July 2016. We excluded book chapters, standards, courses, editorials, prefaces, tutorials, workshops, other than English language articles and poster sessions. Total number of hits was 295 as showed in Table 1.

However, after our initial review, 46 articles were excluded as they are completely non-relevant to survey theme. Finally total articles considered for review are 249. The following in Figure 1, represents the distribution of 249 articles between 5 databases. Out of 249 articles, 175 are having close relationship to DevOps and 74 are not appropriately relevant to the theme. Out of 175 articles, 49 are journal articles and 126 are conference proceedings. The distribution of articles across various software lifecycle phases is as follows which is showed in Figure 2 (2% of articles are related to Requirements Planning, 10% of articles are related Architecture and Design, 25% of articles related to Development, 18% of articles related to Testing, 24%

Table 1. Database Search Results

Data Bases (Journals and Conferences)	No of Hits
IEEE Xplore	211
Sciencedirect	34
ACM Digital Library	23
SpringerLink	12
Wiley InterScience	15
Total	**295**

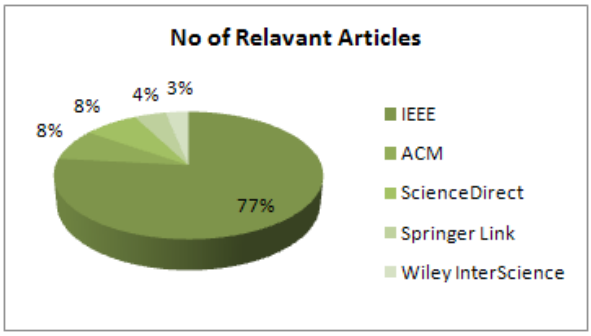

Figure 1. Article distribution per database.

of articles related to Deployment, 8% of articles related to Monitoring, 1% of articles related to Configuration Management, 1% of articles related to Migration, 10% of articles related to Project Management). In Figure 3 presents the article distribution over years from 2011-2016*. 2016* representation is partial figure which is till the month of July 2016.

The goal of this survey is to identify key drivers of technical, cultural and managerial factors impacting testing function in DevOps setting. It addresses the following research question "What are the different motivational factors driving testing in DevOps Environment?".

2.3 Systematic Review Findings

Out of 175 articles, 29 articles are having close relationship with testing activities of DevOps setting. 103 articles are having weak relationship with respective to testing. The key factors associated with Testing in DevOps are listed in the following in Table 2 and 3.

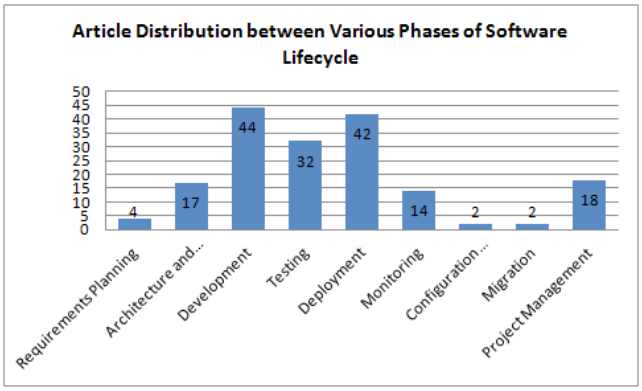

Figure 2. Article distribution across software lifecycle phases.

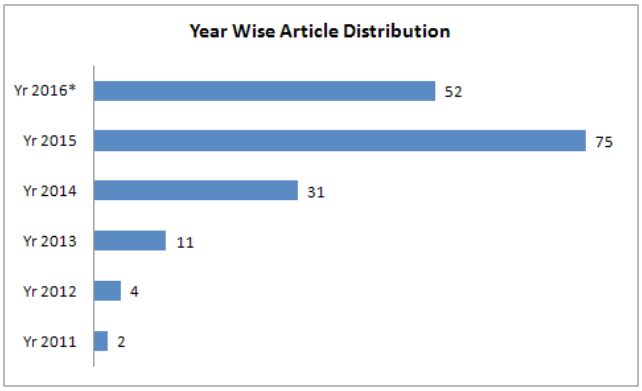

Figure 3. Article distribution over years.

2.4 Extended Manual Literature Survey Review

This section presents the outcome of extended manual literature survey. The critical factors of DevOps culture, roles and responsibilities and trends are presented in this section. DevOps Testing is a mental discipline and Organizational Mindset[8]. Collaborative Development, Continuous Testing, Continuous Release and Deployment, Continuous Monitoring and Continuous Customer Feedback, Continuous Improvement, Continuous Adoption are associated with DevOps[9–11]. DevOps promotes shared resources, optimization of processes and reduces the waste[12]. DevOps demands alternative metrics. They should be totally focused on the quality and stability of the actual services, rather productivity measurements of traditional software engineering process[13]. DevOps in combination of Cloud and Testing becomes Testing-as-a-Service (TaaS). It is effective, collaborative and communication system established between developers, testers, operations and business communities. Test case is developed before a single line of code is developed. Before coding starts, development team is nurtured towards build and test processes. QA partially plays operations role in terms of setting up QA environment (integration testing/system testing), aiding in build automation and providing assistance to operations team. This preparation builds the mindset and culture of collaboration[14]. Developer partially plays tester's role in terms of test automation. QA team constructively provides feedback to Developer during unit testing phase as well integration testing which prevent defects. Tester partially plays developer role in test automation[15]. Test manager should have natural ability to coordinate between development managers, make smooth integration and conduct system testing[16]. Agile process makes less focus on documentation. On the other hand, design of intelligent documentation is essential for DevOps success. Documentation is powerful tool for improving the quality of code, operational processes, and knowledge management function[17]. DevOps goal is to provide feedback on critical quality parameters, performance, reliability, etc and educate developer by automatic means[18]. Test Driven and Behaviour Driven Development (TDD and BDD) are becoming standards in the industry. Customer experience driven testing (behaviour driven testing + exploratory user scenario based testing) is becoming de-facto standard in testing domain. Tester should be

Table 2. Terms and factors driving testing in DevOps (Systematic literature survey)

Table 3 Annexure Source #	Culture/ Collaboration/ Agility / Scale / Reduce Complexity & Cost	Automation (Testcase / deployment) / Auto generation of Test cases	Continuous * / Continuous Testing / TDD/BDD	Virtualization / Cloud Management/ Simulation Testing	Orchestration Framework / Micro services /Open Source Architecture/ Reuse methods	Testbed Design & Development	Alternative Metrics / Project Management/ Skills
Annex1	√					√	
Annex2					√		
Annex3		√			√		
Annex4	√	√	√				
Annex5				√			
Annex6	√				√		
Annex7	√	√		√	√		
Annex8		√		√		√	
Annex9	√	√		√		√	√
Annex10		√	√	√	√		
Annex11		√		√		√	√
Annex12	√				√		√
Annex13	√	√					√
Annex14	√	√	√	√			
Annex15	√	√		√			
Annex16	√	√	√	√	√	√	√
Annex17	√	√				√	√
Annex18		√	√				
Annex19	√	√			√		
Annex20	√	√	√				√
Annex21					√	√	
Annex22	√	√		√		√	
Annex23		√	√				
Annex24		√		√			√
Annex25				√			
Annex26			√	√			
Annex27					√	√	
Annex28				√	√		
Annex29			√	√	√		

capable enough to "shift left" (development perspective) as well as "shift right" (operations perspective)[19]. DevOps is function of sharing, measurement and automation[20]. DevOps promotes continuous testing which is combination of automation, prioritization of test cases and value added testing. It may reduce overall development time as much as 15%[21–23].

However, DevOps propels high resource utilization of key resources for both development and testing. It results in over using of resources and too often approached directly to do ad-hoc tasks[24]. Unless these issues are resolved from testing and QA perspective, there is no value generated from DevOps testing adoption. In DevOps setting, infrastructure is code. Tester is no more

Table 3. Annexure - Systematic literature survey

Annexure Source #	Article Title	Authors	Publisher
Annex1	A Modular Approach to Collaborative Development in an OpenStack Testbed	D. Bruneo; F. Longo; G. Merlino; N. Peditto; C. Romeo; F. Verboso; A. Puliafito	IEEE
Annex2	Dimensions for Evaluating Cloud Resource Orchestration Frameworks	A. Khoshkbarforoushha; M. Wang; R. Ranjan; L. Wang; L. Alem; S. U. Khan; B. Benatallah	IEEE
Annex3	Orchestration framework for automated Ajax-based web application testing	H. H. Deyab; R. B. Atan	IEEE
Annex4	Consistent Testing Framework for Continuous Service Development	C. Sathawornwichit; S. Hosono	IEEE
Annex5	Operation Changes Recommendation Method Using Histories of Operation Changes in Cloud Computing Environment	S. Kitajima; S. Kikuchi; Y. Matsumotoy	IEEE
Annex6	Evaluating the monolithic and the microservice architecture pattern to deploy web applications in the cloud	M. Villamizar; O. Garcés; H. Castro; M. Verano; L. Salamanca; R. Casallas; S. Gil	IEEE
Annex7	SENDIM for Incremental Development of Cloud Networks: Simulation, Emulation and Deployment Integration Middleware	P. Kathiravelu; L. Veiga	IEEE
Annex8	Chaos Engineering	A. Basiri; N. Behnam; R. de Rooij; L. Hochstein; L. Kosewski; J. Reynolds; C. Rosenthal	IEEE
Annex9	Automated and Isolated Tests for Complex Middleware Products: The Case of BPEL Engines	S. Harrer; C. C. Röck ; G. Wirtz	IEEE
Annex10	Test orchestration a framework for Continuous Integration and Continuous deployment	N. Rathod; A. Surve	IEEE
Annex11	Practical Challenges in Test Environment Management	R. Ramler; J. Gmeiner	IEEE
Annex12	Open source and standards: The role of open source in the dialogue between research and standardization	S. A. Wright; D. Druta	IEEE
Annex13	Development and Deployment at Facebook	D. G. Feitelson; E. Frachtenberg; K. L. Beck	IEEE
Annex14	Continuously delivering your network	S. Gebert; C. Schwartz; T. Zinner; P. Tran-Gia	IEEE
Annex15	Cloud Crawler: a declarative performance evaluation environment for infrastructure-as-a-service clouds	M. Cunha, N. C. Mendonça and A. Sampaio	Wiley Online Library
Annex16	Continuous software engineering: A roadmap and agenda	Brian Fitzgerald, Klaas-Jan Stol*	ScienceDirect
Annex17	When and what to automate in software testing?	Vahid Garousi, Mika V. Mäntylä,	ScienceDirect
Annex18	Emerging themes in agile software development: Introduction to the special section on continuous value delivery	Torgeir Dingsøyr, Casper Lassenius	ScienceDirect
Annex19	Model-driven engineering of information systems: 10 years and 1000 versions	Jim Davies, Jeremy Gibbons, James Welch, Edward Crichton,	ScienceDirect
Annex20	Challenges, Benefits and Best Practices of Performance Focused DevOps	Wolfgang Gottesheim	ACM

Continued

Annex21	Feedback-driven Combinatorial Test Design and Execution	Itai Segall, Rachel Tzoref-Brill	
Annex22	A DevOps Approach to Integration of Software Components in an EU Research Project	Mark Stillwell, Jose G. F. Coutinho	ACM
Annex23	Continuous Software Engineering and Beyond: Trends and Challenges	Brian Fitzgerald, Klaas-Jan Stol	ACM
Annex24	Coverage-based Metrics for Cloud Adaptation	Yonit Magid, Rachel Tzoref-Brill, Marcel Zalmanovici	ACM
Annex25	DICE Fault Injection Tool	Craig Sheridan, Darren Whigham, Matej Artač	ACM
Annex26	PET: Continuous Performance Evaluation Tool	Johannes Kroß, Felix Willnecker, Thomas Zwickl, Helmut Krcmar	ACM
Annex27	A Tool for Verification of Big-data Applications	Marcello M. Bersani, Francesco Marconi, Matteo Rossi, Madalina Erascu	ACM
Annex28	A Systematic Approach for Performance Evaluation Using Process Mining: The POSIDONIA Operations Case Study	Simona Bernardi, José Ignacio Requeno, Christophe Joubert, Alberto Romeu	ACM
Annex29	Model-based Performance Evaluations in Continuous Delivery Pipelines	Markus Dlugi, Andreas Brunnert, Helmut Krcmar	ACM

defect finder, but defect preventer. Continuous testing raises integration issues much earlier in the software life cycle, makes defect fixing cheaper, quicker, frees tester's precious time for exploratory testing and value added test activities. DevOps may demand high risk coverage, optimal cost, and better utilization of hardware, software and seamless communication/collaboration between business users, domain experts, testers and developers. While companies adopt continuous delivery and DevOps, they may encounter problems fast like deployment issues, memory leaks, inefficient coding, non-optimized database access, etc. They need to be detected proactively and if possible through automated means[25,26].

DevOps process has to be metric driven. We need to design measurement criteria in the form of KSA (Knowledge, Skills, and Abilities). The new set of metrics can be in the form of tangible and non-tangible. Tangible metrics are like test coverage, test case productivity, test case complexity, test automation percentage, tester efficiency, collaboration/handshake index, defect density, traceability percentage, etc. The non-tangible metrics could be like overall test confidence index, test estimation as it involves lot of risk and confidence factors[27]. There should be some mechanism to design a test metric while code is in the development system. In such case, development has not yet been completed and not handed over to tester but developer pre-releases this version to check the feel good factor. This metric helps in determining the confidence level. Tester does high level code walk through and test relevant test cases associated to that version of the code. Version control system should be redesigned to allow this feature which is important from DevOps point of view. Similarly, QA should able to share customer experience related metrics to operations team which are highly critical for planning server side metrics. QA has major role in prioritizing bugs and release plans. The intelligent role of QA is to present data appropriately which helps in decision making of certain release cycle but not listing huge pile of bugs with no-go as aim[28]. The other critical performance indicators are percentage of delivered features accepted (for first time go-live), price point for accepted feature and test velocity. Majority of DevOps services are provided in shared platform. There is a need to develop common framework which ensures process for requirements collection, user experience (one of the important parameters of customer satisfaction) and combinatorial test design (optimization methods)[29]. These types of metrics should be defined and measured on continuous basis. While DevOps encourages faster and smaller releases to meet minimal lead times, scope of integration and system testing increases. This has to be properly managed while considering effort estimation. At present, the maturity of parameterized test estimation is low. In most cases, test estimation effort is derived from

generic software development estimation effort, in which testing phase is one of the phases of software development life cycle[30]. This situation necessitates re-design of software test estimation model in the context of DevOps. It should be user friendly and should able to provide benchmarking like COSMIC function points[31].

Finally, DevOps or Agile utilizes every resource employed in the process (be it developer or tester or release engineer or project manager)[32]. DevOps is completely people centric while focuses on customer value[33,34].

3. Discussion and Threats to Validity

Agile is meant for parallel development and test. DevOps is an evolution beyond agile. DevOps is a philosophy itself. Industry is adopting with vigor and trying to define, establish processes and standards. It is a journey. The role of academic research is to provide critical evaluation of the research progress and establishing the philosophy.

Our systematic literature survey and extended manual literature survey on DevOps Testing has predominantly adopted a quantitative approach with the support of qualitative research. We found that only 29 articles having close association to Testing. From the year-wise distribution of articles, it is evident that most of the articles connected to DevOps testing got published post 2011 and surged only after 2014. This phenomenon shows the nascence of the research progress in this space. We observed that DevOps in testing is closely associated with automation of test cases and at it advanced stage, auto generation of test cases through model driven frameworks. This thought has been stressed in more than 50% of articles. Test driven development and Behavioural driven developments are providing perfect balance between business interests and technical interests. They are also driving concept of shared resources, tools and technologies, optimization of cost and time. More than 50% of articles are connected Cloud, Virtualization, Simulation techniques of DevOps Testing. DevOps testing is continuous. The elements of agility, scale, metric driven process, reduction of complexity and cost appeared in more than 16 articles. DevOps demands alternative metrics for better collaboration and communication between various stakeholders of the system. DevOps testing is face of organization culture and human resource mindset. This is sounded in all most all articles associated to DevOps.

This research is constrained from couple of biases (authority and publication). Research valued opinions of the other researchers which may be wrong. Results of the publications may be biased towards positive results. The search string used (DevOps Testing) may have multiple synonyms but we de-risked this threat by manually verifying each and every result and also conducted manual search based on this search string.

4. Conclusions

In the software development process, DevOps testing is promising improvement in quality, collaboration and communication between all stakeholders including business users. It is poised to address issues raised in agile development methodology. However, we found that DevOps testing has not been systematically studied in academic scientific literature. We did not find any real-time case studies in the context of DevOps testing frameworks in academic journals. It propels to have more action research in this area. The traditional isolated QA / Testing skills may find challenge and limited role to play in DevOps. QA/Tester has to scale beyond regular testing function and aid development and operations teams to meet DevOps philosophical objectives. DevOps testing needs design of alternative metrics/measures which elevate the culture of an organization, collaboration and build proper benchmarking base.

5. References

1. Claps GG, Svensson RB, Aurum A. On the journey to continuous deployment: Technical and social challenges along the way. Information and Software Technology. Elsevier. 2015; 57:21–31.
2. Humble J, Farley D. Continuous delivery: Reliable software releases through build, test, and deployment automation. Pearson Education Inc; 2011.
3. Downe-Wamboldt B. Content analysis: Method, applications, and issues. Health care for Women International. 1992; 13:313–21.
4. Hodder I. The interpretation of documents and material culture. Thousand Oaks etc Sage; 1994. p. 155.
5. Higgins JPT, Green S. Cochrane handbook for systematic reviews of interventions (version 5.1. 0). The Cochrane Collaboration; 2011.
6. Thakurta R. Research trends on software requirement prioritization. International Journal of Software Engineering and Its Applications. 2014; 8(6):287–98.

7. DevOps literature review. Available from: http://www.utwente.nl/bms/iebis/staff/amrit/devopsreport.pdf
8. Black R. Pragmatic software testing: Becoming an effective and efficient test professional. Indianapolis, IN: John Wiley and Sons; 2007. p. 6–9.
9. Devops in the federal sector. https://www.reisystems.com/REI%20Case%20Studies/DevOps%20in%20the%20Federal%20Sector_REI%20Systems_2014.pdf
10. Limoncelli TA, Chalup SR, Hogan CJ. DevOps culture. The practice of cloud system administration: Designing and operating large distributed systems. Boston: Addison-Wesley Professional; 2014. p. 172–93.
11. Goltz U, et al. Design for future: Managed software evolution. Computer Science-Research and Development. NY: Springer-Verlag. 2015; 30(3-4):321–31.
12. Drogseth DN, Sturm R, Twing D. Chapter 7– CMDB system use cases: Carving out the right place to start. in CMDB systems: Making change work in the age of cloud and agile. Burlington: Morgan Kaufmann Publishers; 2015. p. 117–39.
13. Fuggetta A, Di Nitto E. Software process. FOSE 2014 Proceedings of the on Future of Software Engineering; New York. 2014. p. 1–12.
14. Technology Forecast. Available from: www.pwc.com/techforecast
15. Bartusevicsa A, Novickisa L. Models for implementation of software configuration management. Procedia Computer Science. Elsevier. 2015; 43:3–10.
16. Gregory J, Crispin L. Chapter 23-testing and DevOps: In more agile testing: Learning journeys for the whole team. Addison-Wesley Professional; 2014.
17. Sacks M. Chapter 6: Production launches. Pro Website Development and Operations: Streamlining DevOps for large-scale websites. 1st ed. Berkely, CA: Apress; 2012. p. 80–92.
18. Perez JF, Wang W, Casale G. Towards a DevOps approach for software quality engineering. WOSP '15 Proceedings of the 2015 Workshop on Challenges in Performance Methods for Software Development; Austin, TX. 2015. p. 5–10.
19. World Quality Report. Available from: http://www.asktester.com/wp-content/uploads/2015/10/world_quality_report_2015_16_capgemini_hp_sogeti.pdf
20. Humble J, Molesky J. Why enterprises must adopt devops to enable continuous delivery. Cutter IT J. 2011; 24(8):6–12.
21. Saff D, Ernst MD. Reducing wasted development time via continuous testing. Proceedings of the 14th International Symposium on Software Reliability Engineering; 2003. p. 281–92.
22. Marijan D, Gotlieb A, Sen S. Test case prioritization for continuous regression testing: An industrial case study. Proceedings of the 29th IEEE International Conference on Software Maintenance; 2013. p. 540–43.
23. Gladston A, Nehemiah K, Narayanasamy P, Kannan A. Test case prioritization for regression testing using immune operator. The International Arab Journal of Information Technology. 2016; 13(6).
24. DevOps: Hidden risks and how to achieve results. Available from: http://www.dynatrace.com/content/dam/en/general/ebook-devops.pdf
25. Gottesheim W. Challenges, benefits and best practices of performance focused DevOps. ACM LT'15 Proceedings of the 4th International Workshop on Large-Scale Testing; Austin, TX. 2015. p. 3–3.
26. Gajda W. Chapter 1. Getting Started with Vagrant. 1st ed. Pro Vagrant. Berkely, CA, USA: Apress; 2015.
27. Bang SK, Chung S, Choh Y, Dupuis M. A grounded theory analysis of modern web applications: Knowledge, skills, and abilities for DevOps. ACM RIIT'13 Proceedings of the 2nd Annual Conference on Research in Information Technology; Orlando, FL. 2013. p. 61–2.
28. Roche J. Adopting DevOps practices in quality assurance: Merging the art and science of software development. Quality Assurance. 2013; 11(9).
29. Segall I, Tzoref-Brill R. Feedback-driven combinatorial test design and execution. ACM SYSTOR'15 Proceedings of the 8th ACM International Systems and Storage Conference; Haifa, Israel. 2015.
30. Jayakumar KR, Abran A. A survey of software test estimation techniques. Journal of Software Engineering and Applications. 2013; 6:47–52.
31. Fehlmann T, Santillo L. From story points to COSMIC function points in agile software development- A six sigma perspective. IWSM/MetriKon, Software Metrik Kongress; Stuttgart, Germany. 2010. p. 1–11.
32. Gandomani TJ, Zulzalil H. Compatibility of agile software development methods and CMMI. Indian Journal of Science and Technology. 2013; 6(8):5089–94.
33. Hamid, Saul S, et al. Factors limiting the implementations of agile practices in the software industry: A pilot systematic review. Indian Journal of Science and Technology. 2015; 8(30):1–11.
34. Sheshasaayee A, Hannah V. Identifying bottlenecks in agile software development using theory of constraints principles. Indian Journal of Science and Technology. 2015; 8(29):1–10.