Association for Information Systems
AIS Electronic Library (AISeL)

PACIS 2017 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

Summer 7-19-2017

DevOps: Concepts, Practices, Tools, Benefits and Challenges

Georges Bou Ghantous
University of Technology Sydney, Georges.Boughantous-1@uts.edu.au

Asif Gill
University of Technology Sydney, asif.gill@uts.edu.au

DevOps: Concepts, Practices, Tools, Benefits and Challenges

Indicate Submission Type: Completed Research Paper

Georges Bou Ghantous
School of Software
University of Technology Sydney
Ultimo NSW 2007, Australia
Georges.Boughantous-1@uts.edu.au

Asif Qumer Gill
School of Software
University of Technology Sydney
Ultimo NSW 2007, Australia
Asif.Gill@uts.edu.au

Abstract

DevOps, originated in the context of agile software development, seems an appropriate approach to enable the continuous delivery and deployment of working software in small releases. Organizations are taking significant interest in adopting DevOps ways of working. The interest is there, however the challenge is how to effectively adopt DevOps in practice? Before disembarking on the journey of DevOps, there is a need to clearly understand the DevOps concepts, practice, tools, benefits and underlying challenges. Thus, in order to address the research question in hand, this paper adopts a Systematic Literature Review (SLR) approach to identify, review and synthesize the relevant studies published in public domain between: 2010-2016. SLR approach was applied to initially identify a set of 450 papers. Finally, 30 of 450 relevant papers were selected and reviewed to identify the eight key DevOps concepts, twenty practices, and a twelve categories tools. The research also identified seventeen benefits of using DevOps approach for application development and encountered four known challenges. The results of this review will serve as a knowledge base for researchers and practitioners, which can be used to effectively understand and establish the integrated DevOps capability in the local context.

Keywords: Architecture, automation, continuous development, continuous integration, DevOps

Introduction

Information system (IS) development is continuously evolving from documentation-driven to more agile and collaborative ways of working [31]. Initially, agile ways of working mainly focused on the development (Dev) aspects of the IS and little attention was given to the operations (Ops) [12, 14]. More recently, agile approaches introduced the integrated concept of DevOps [1]. It has been reported that "DevOps is a set of practices that is trying to bridge developer-operations gap at the core of things and at the same time covers all the aspects which help in speedy, optimized and high quality software delivery" [22]. DevOps practices focus on continuous-deployment, log-monitoring, automated testing and update components, etc. [7-8, 12, 14]. These practices can be supported by a number of tools such as repository, build and deploy management tools [21, 27].

According to Erich et al. [23]; "DevOps automation is supported by various design patterns which improve the continuous delivery of software application on cloud platform". DevOps claims to enable "Faster delivery of builds, features, and bug fixing thereby creating a continuous build pipeline" [8]; however, the adoption of DevOps is not a straightforward task [4]. There is a need to clearly understand the underlying DevOps concepts, practice, tools, benefits and challenges for its effective adoption for IS development. Thus, as a first step in DevOps adoption research, this study focuses on the following important research question (RQ):

- What is known about the DevOps concepts, practices, tools, benefits and challenges?

In order to address the above mentioned research question in hand, we applied a well-known systematic literature review (SLR) method [32], which is deemed to be an appropriate approach to systematically identify, review and synthesize the DevOps body of knowledge. This paper is organized as follows. Firstly, it discusses the research background and method. Secondly, it presents the SLR study results. Finally, it discusses the results and concludes with further research implication and opportunities.

Research Background and Method

The history of DevOps can be traced back to 2007 [1, 3-4]. There are a number of frameworks that are emerging to support DevOps such as secure continuous deployment framework [1], SQUID (Specification Quality in DevOps) framework [2], Composable DevOps Automated Ontology [6], DevOps pipeline automation [8]. This increasing number of DevOps frameworks requires systematically reviewing and synthesizing of DevOps approaches for its effective understanding and adoption. Thus, this paper address this important need and presents a systematic review of DevOps concepts, practices, tools, benefits and challenges using a SLR method [32]. This study included papers from the five well-known databases (Table 1). Based on the research question, a search key (SK) was formed and used to filter and select the relevant papers for this SLR study. Search Key is a logical expression built of a string concatenation of this paper's key-terms using OR-Boolean operator as a connector. The resulted Search key is:

SK = "DevOps Framework" OR "DevOps tools" OR "DevOps practices" OR "DevOps Architecture" OR "DevOps Automation" OR "DevOps Continuous Deployment" OR "DevOps Continuous Integration".

Search key was applied across all the databases to get the initial count of 450 papers, which were exported to EndNote and Excel Sheets including citations and abstracts of the identified papers. Firstly, papers with only title and abstracts were excluded from the results list, which left us with 284 full papers. Secondly, we screened and eliminated the duplicate papers from the result list, which resulted in a set of 253 papers. Thirdly, papers with relevant titles were included to the review list, which resulted in a set of 160 papers. Fourthly, we selected 102 papers based on the review of the abstracts of 160 papers. Fifthly, we selected 60 papers out of 102 based on the review and relevance of abstract and introduction. Please see figure 1 for the filtration process.

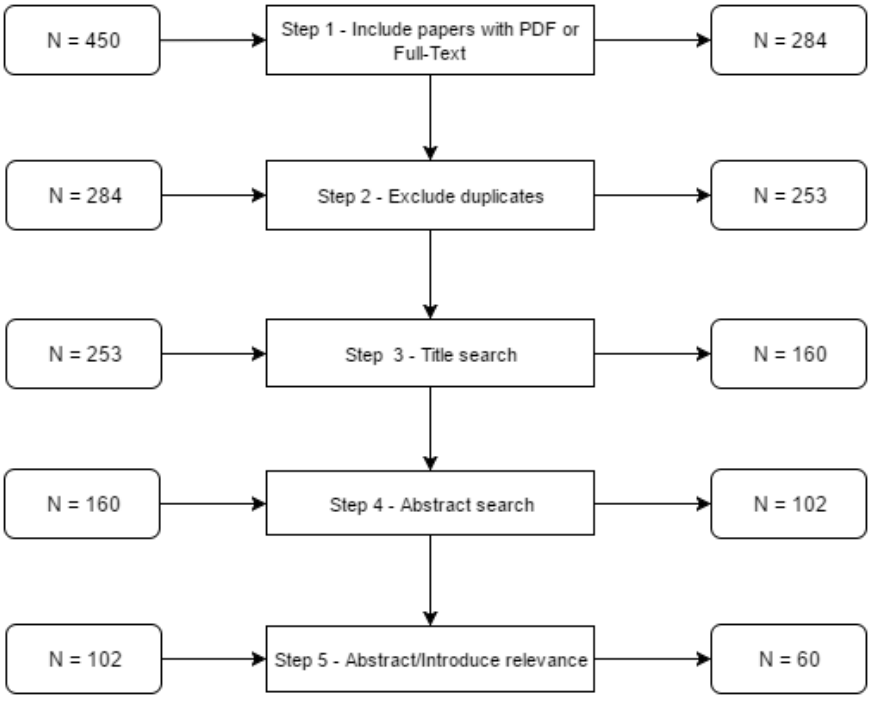

Figure. 1. Filtration Process Illustration, shows a step by step study selections

Table 1. Filtration Process Results								
Database	**Initial Results**	**Step 1**	**Step 2**	**Step 3**	**Step 4**	**Step 5**	**Filtered Results**	**Percentage**
IEEEXplore	72	72	69	51	35	28	**28**	47%
ACM Digital Library	9	9	8	7	7	6	**6**	10%
Elsevier Science Direct	14	9	9	2	2	2	**2**	3%
SpringerLink	27	9	9	6	4	4	**4**	7%
Google Scholar	328	185	158	94	54	20	**20**	33%
N=	**450**	284	253	160	102	60	**60**	**100%**

Table 1. Filtration Process Results

The filtration process resulted in 60 relevant papers. These papers were then reviewed to identify the final count of 30 relevant papers (S1-S30) that address the research question in hand. The review process utilized is a five-question quality screening approach [33]. The 30 selected studies satisfied the quality questionnaires in Table 2. DevOps concepts, practices, tools, benefits and challenges were extracted from these 30 studies. Next section presents the review results.

Table 2. Quality Criteria Questions	
1	Is there a clear statement of the aims of the research?
2	Does the paper provide relevant data related the research topics?
3	Is there a clear statement of findings?
4	How adequately has the research results been documented?
5	Is the study of value for research?

Table 2. Quality Criteria Questions

The SLR review process selection results are presented in Table 3.

Table 3. Review Process Results				
IEEEXplore	72	28	15	50%
ACM Digital Library	9	6	5	17%
Elsevier Science Direct	14	2	2	7%
SpringerLink	28	4	1	3%
Google Scholar	387	20	7	23%
	450	**60**	**30**	**100%**

Table 3. Review Process Results

Results

This review identified a final set of 30 relevant studies (S1 – S30) that were systematically identified and reviewed to address the research question in hand. This review resulted in a set of eight DevOps concepts, twenty practices, twelve categories of tools, seventeen benefits and four challenges. These findings are analysed and catalogued in order to enhance the understanding of DevOps, which is a precursor to its informed and less risky adoption.

DevOps Concepts

This SLR study identified communication and collaboration, continuous deployment, continuous delivery, continuous planning, automated pipeline, Quality Assurance as the conceptual elements describing the complex phenomenon of DevOps. These concepts and their paper sources are described in Table 4. Communication and collaboration (23%) along with the continuous delivery (21%) are the highly reported conceptual element (Table 4).

Table 4. DevOps Concepts

DevOps Concepts	Frequency	Percentage	Sources
Communication and Collaboration	9	23%	S_1, S_4, S_8, S_{12}, S_{13}, S_{19}, S_{20}, S_{24}, S_{27}
Continuous deployment	3	8%	S_2, S_3, S_{23}
Continuous delivery	8	21%	S_1, S_4, S_8, S_{13}, S_{18}, S_{22}, S_{23}, S_{27}
Automated pipeline	6	15%	S_3, S_4, S_{18}, S_{19}, S_{22}, S_{23}
Continuous planning	2	5%	S_{18}, S_{20}
Continuous feedback	4	10%	S_3, S_8, S_{22}, S_{24}
Roll back code	2	5%	S_2, S_{20}
Quality Assurance	5	13%	S_9, S_{13}, S_{20}, S_{22}, S_{27}
Total Source Connections	**39**	**100%**	

Table 4. DevOps Concepts

DevOps Practices (DP)

There are a number of DevOps practices that are being implemented by the organizations in order to achieve their objectives [8]. This review identified a set of twenty DevOps practices (PC1-PC20), which are detailed and catalogued in Appendix A – DevOps Practices. Automate sandbox deployment (PC2), synchronization of critical services such as transactions, performance, uptime, deployment schedule, run-time costs, version control, and project scope (PC11), and application release deployment pipeline automation (PC13) are highly mentioned practices in the selected studies (Table 5).

Table 5. DevOps Practices

Practices	Frequency	Percentage	Sources
PC_1	2	1.65%	[S_6] [S_{14}]
PC_2	9	7.44%	[S_6] [S_8] [S_{10}] [S_{13}] [S_{22}] [S_{14}] [S_{18}] [S_{20}] [S_{22}]
PC_3	7	5.79%	[S_6] [S_{12}] [S_{22}] [S_{14}] [S_{20}] [S_{22}] [S_{24}]
PC_4	5	4.13%	[S_6] [S_{10}] [S_8] [S_{20}] [S_{22}]
PC_5	3	2.48%	[S_6] [S_8] [S_{24}]
PC_6	4	3.31%	[S_3] [S_6] [S_{15}] [S_{24}]
PC_7	7	5.79%	[S_6] [S_{10}] [S_8] [S_{13}] [S_{20}] [S_{22}] [S_{27}]
PC_8	7	5.79%	[S_6] [S_8] [S_{10}] [S_{15}] [S_{18}] [S_{20}] [S_{22}]
PC_9	8	6.61%	[S_4] [S_6] [S_{10}] [S_{13}] [S_{15}] [S_{18}] [S_{20}] [S_{22}]
PC_{10}	8	6.61%	[S_3] [S_6] [S_8] [S_{15}] [S_{18}] [S_{20}] [S_{22}] [S_{24}]
PC_{11}	9	7.44%	[S_6] [S_8] [S_{14}] [S_{15}] [S_{20}] [S_{22}] [S_{23}] [S_{24}] [S_{27}]

PC_{12}	7	5.79%	[S_6] [S_8] [S_{14}] [S_{15}] [S_{20}] [S_{22}] [S_{23}]
PC_{13}	9	7.44%	[S_6] [S_8] [S_{13}] [S_{14}] [S_{15}] [S_{20}] [S_{22}] [S_{23}] [S_{27}]
PC_{14}	5	4.13%	[S_3] [S_6] [S_8] [S_9] [S_{20}]
PC_{15}	4	3.31%	[S_6] [S_8] [S_{18}] [S_{20}]
PC_{16}	7	5.79%	[S_6] [S_8] [S_9] [S_{18}] [S_{20}] [S_{23}] [S_{27}]
PC_{17}	5	4.13%	[S_6] [S_9] [S_{18}] [S_{20}] [S_{23}]
PC_{18}	6	4.96%	[S_6] [S_8] [S_{13}] [S_{18}] [S_{20}] [S_{23}]
PC_{19}	4	3.31%	[S_6] [S_8] [S_{18}] [S_{20}]
PC_{20}	5	4.13%	[S_3] [S_6] [S_8] [S_{18}] [S_{20}]
Total	**121**	**100%**	

Table 5. DevOps Practices

DevOps Tools

DevOps practices require the support of appropriate technology [28]. There are a number of DevOps tools (See Appendix B: DevOps Tools Catalogue) such as Jenkins and Codeship (Continuous Integration, Continuous Testing), Puppet and Ansible (Cloud Management), New Relic and AWS CloudWatch (Monitoring), Bitbucket and Github (Repository), MongoDB (NoSQL Database Management), and HipChat (DevOps team Communication). Cataloguing such tools and their usefulness will enable the organisations to make an informed decision about the different types of DevOps technology and their local needs [29]. In this review, we found twelve categories of DevOps tools, which are shown in Table 6. IaaS/PaaS along with the continuous integration and deployment are the highly reported technology categories for enabling DevOps.

Table 6. DevOps Tool Categories			
DevOps Tools Categories	**Sources**	**Frequency**	**Percentage**
Source Control Management	S_1, S_3	2	2.78%
Continuous Integration	S_1, S_4, S_{14}, S_{16}, S_{17}, S_{19}, S_{21}, S_{28}, S_{29}, S_{30}	10	13.89%
Continuous Deployment	S_1, S_4, S_5, S_{14}, S_{17}, S_{19}, S_{21}, S_{28}, S_{29}, S_{30}	10	13.89%
IaaS/PaaS	S_1, S_3, S_5, S_{14}, S_{16}, S_{17}, S_{19}, S_{21}, S_{25}, S_{28}, S_{29}	11	15.28%
Monitoring	S_1, S_3, S_{25}	3	4.17%
Database Management	S_1, S_3	2	2.78%
Containerization	S_1, S_3, S_4, S_7, S_{16}, S_{28}	6	8.33%
Configuration And Provisioning	S_1, S_3, S_4, S_5, S_{17}, S_{19}, S_{21}, S_{28}, S_{29}, S_{30}	10	13.89%
Logging/Security	S_1	1	1.39%
Build	S_1, S_4, S_{30}	3	4.17%
Testing	S_1, S_{14}	2	2.78%
Collaboration	S_1, S_{12}, S_{18}	3	4.17%
	Total	**72**	**100%**

Table 6. DevOps Tool Categories

This study purposely did not include the rating of the tools because this is an academic study and it does not promote or demote any tools or vendors. However, a catalogue of identified tools can be provided to reviewers on request. Organisation should make their own judgement based on this review and their requirements when selecting specific tools to support DevOps.

DevOps Benefits and Challenges

It is found in this review that DevOps approach seems to offer a number of benefits; however, there are some challenges as well. This review identified a large list of seventeen benefits compared to only four challenges (Table 7). These benefits and challenges provide useful insights that should be considered in the context of DevOps adoption (Table 7).

Table 7. Benefits and Challenges

Benefits	**Challenges**
-Code version control provided by Bitbucket. -Parallel deployment (Codeship) -Enable scheduled deployment exp. Codeship. -Enable scheduled testing (Jenkins, Codeship). -Provide QA automated testing by Codeship test commands or Jenkins plugins. -Provide continuous integration (Codeship). -Provide real-time automated monitoring -Provide real-time communication by HipChat. -Provide cloud and database management using AWS integration with MongoDB. -Provide Rollback of code and continuous planning based on Retro-QA results. -Continuous innovations and development of new ideas based on continuous planning. -Rapid delivery using cycle build-test-deploy -High scalability of resources: no down time -Provide real-time visibility of the pipeline. -Regular feedbacks from logs and dynamic learning (Codeship, AWS, Bitbucket, etc.) -Secure pipeline using tools authentication -Automated cloud deployment (AWS, Heroku). -Scalable, repeatable and automated processes	-Overcoming the Dev vs. Ops Mentality: -Moving from Legacy Infrastructure to Micro-services: Older infrastructure and applications can be problematic. Using Infrastructure-as-code together with micro-services is yet another step towards a future of continuous innovation. -Too Much Focus on Tools: DevOps relies on variety of tools to construct a SD pipeline. The integration of those tools can prove problematic and difficult to integrate and maintain at times.-Resistance to Change The move to DevOps can seem scary to a majority of team members and key stakeholders. Packaging it as an evolution of current development practices rather than a revolution can help that issue. -Dev and Ops Toolset Clashes: Tools can create another problem of Dev and Ops teams having completely separate toolsets and metrics.

Sources	S_3, S_7, S_8, S_9, S_{14}, S_{15}, S_{22}, S_{25}, S_{30}	S_{11}, S_{12}, S_{22}, S_{24}	**Total**
Frequency	9	4	**13**
Percentage	70%	30%	**100%**

Table 7. Benefits and Challenges

Discussion

DevOps is emerging an appropriate approach to address the important integration concern of development and operations capabilities to complement the contemporary agile approaches. However, it is still challenging for organisations whether DevOps can be effectively adopted at an optimal scale. Before adopting DevOps, there is a need to clearly understand the DevOps concepts, practices, tools and underlying benefits and challenges. This study addressed this important need and

applied a well-known SLR method to systematically review and synthesize the literature published on DevOps in the public domain.

Firstly, this SLR study findings reveal that there is not a single or universal definition of DevOps. This study makes an important contribution to knowledge body by systematically uncovering the underlying eight key conceptual elements of DevOps (see Table 4). It has been observed from the analysis of results that despite DevOps is perceived as an automation centric approach, however, the key building blocks are human communication and collaboration, continuous delivery and automated pipeline.

Secondly, this study identified a set of twenty concrete DevOps practices that detail the abstract DevOps conceptual elements (see Table 5). It has been observed from the analysis of results that the focus is on automated code sandboxes (PC2), auto-generated testing reports (PC9), automated monitoring (PC10), synchronization (PC11) and automated application release deployments (PC13).

Thirdly, it is clear from the analysis that the key theme is around "automation". Thus, this study identified a large number of tools, which are organised into twelve categories to support the automation of DevOps pipeline and underlying practices.

Further, this paper also presented a set of seventeen benefits and four challenges of DevOps. Hence, In order to address the research question in hand, we have distilled a number of DevOps related concepts, practices, tools, benefits and challenges, from systematically selected studies, to help facilitating the understanding and adoption of emerging DevOps ways of working in practice. This demonstrates the applicability of this academic research to practice. Thus, this study has implications for both researchers and practitioners. For instance, researchers may be interested to further research and develop theoretical DevOps capability assessment and adoption models for practitioners. Practitioners could be interested to use the theoretical models and assess the identified DevOps concepts, practices and tools for identifying the gaps, and developing new features for existing or new DevOps frameworks. This is important for the continuous research and development of DevOps ways of working.

Conclusion

DevOps is an emerging concept in agile SD. DevOps is perceived as an automated tool-chain for agile SD. Organisations are showing significant interest in DevOps, however, they lack clarity and understanding of DevOps and underpinning concepts, practices, tools, benefits and challenges. This paper is an attempt to provide such clarity and understanding to facilitate the informed and less risk adoption of DevOps. Thus, this paper applied a SLR method and systematically identified a final set of 30 papers. These papers were then reviewed in detail to extract the relevant data and developed catalogues of DevOps concepts, practices, tools, benefits and challenges. These catalogues provide a collective knowledge base of DevOps that can be used by researchers and practitioners to further enhance their understanding and enable effective adoption of DevOps approach in their local context.

References

[1] Farroha, B. S., Farroha D. L. 2014. "A Framework for Managing Mission Needs, Compliance and Trust in the DevOps Environment" Department of Defense, Ft. Meade, MD; IEEE Military Communications Conference. [S1]

[2] Di Nitto E., Jamshidi P., Guerriro M., Spais I., Tamburri D. 2016. "A Software Architecture Framework for Quality-Aware DevOps". Saarbrücken, Germany ACM. 978-1-4503-4411-1 [S2]

[3] Chen H., Kazman R., Haziyev S., Kropov V. 2015 "Architectural Support for DevOps in a Neo-Metropolis BDaaS Platform" System Architecture Group, Softserve Inc. Austin, TX, USA; Dimitri Chtchourov, GTO Group, Cisco Systems, San Fransico, USA, IEEE 34th Symposium on Reliable Distributed Systems Workshops. [S3]

[4] Wettinger J., Andrikopoulos V., Leymann F. 2015. "Automated Capturing and Systematic Usage of DevOps Knowledge for Cloud Applications" Institute of Architecture of Application Systems (IAAS), University of Stuttgart, Stuttgart, Germany; 2015 International Conference on Cloud Engineering. [S4]

[5] Kumar K. S., Thakur S., Chaurasiya H., Nagdev H. 2015. "Automated Provisioning of Application in IAAS Cloud using Ansible Configuration Management" Computer Science and Engineering ASET, Amity University Noida, India; 1st International conference on next generation Computing Technologies(NGCT-2015) Dehradun, India, 4-5 September 2015. [S5]

[6] McCarthy M. A., Herger L. M., Khan S. M., Belgodere B. M. 2016. "Composable DevOps Automated Ontology Based DevOps Maturity Analysis" Dept. of Comp. and Elect.Eng.and Computer Science Florida Atlantic University, Boca Raton, FL, USA, IEEE International Conference on Services Computing. [S6]
[7] Moore J., Kortuem G., Smith A., Chowdhury N., Cavero J., Gooch D. 2016. "DevOps for the Urban IoT" The Open University, Milton Keynes, UK, ACM. ISBN 978-1-4503-4204-9/16/05. [S7]
[8] Soni M. 2015. "End to End Automation on Cloud with Build Pipeline: The case for DevOps in Insurance Industry, IGATE, Gandhinagar, India; 2015 International Conference on Cloud Computing in Emerging Markets. [S8]
[9] Wahaballa A., Wahballaa O., Abdellatief M. , Xiong H., Qin Z. 2015. "Toward Unified DevOps Model" University of Electronic Science and Technology of China, Chengdu, China, IEEE 978-1-4799-8353-7. [S9]
[10] Virmani M. 2015. "Understanding DevOps & Bridging the gap from Continuous Integration to Continuous Delivery" Development Manager IBM Rational; Fifth international conference on Innovative Computing Technology (INTECH 2015) [S10]
[11] Syed M. H., Fernandez E. B. 2016. "Cloud Ecosystems support for Internet of Things and DevOps using Patterns" Florida Atlantic University, Boca Raton, FL, USA; 2016 First International Conference on Internet-of-Things Design and Implementation IEEE 978-1-4673-9948-7. [S11]
[12] Diel E., Marczak S., Cruzes D. S., 2016. "Communication Challenges and Strategies in Distributed DevOps" 2016 11th International Conference on Global Software Engineering IEEE 2329-6313/16. [S12]
[13] Mahanta P., Pole A. K., Adige V. S., Rajkumar M. 2016. "DevOps Culture and its impact on Cloud Delivery and Software Development" SAP Labs Pvt. LTD. Bangalore, India, 2016 IEEE 978-1-5090-0673-1/16. [S13]
[14] Bayser M., Azevedo L. G., Cerqueira R. 2015. "ResearchOps: The Case for DevOps in Scientific Applications" IBM Research, Av. Pasteur Batofogo, Rio de Janeiro, RJ, Brazil, IFIP 978-3-901882-76-0. [S14]
[15] John W., Meirosu C., Pechenot B., Skoldstrom P., Kreuger P., Steinert R. 2015. "Scalable Software Defined Monitoring for Service Provider DevOps" Fourth European Workshop on Software Defined Networks, 2015 IEEE 978-1-5090-0180-4/15. [S15]
[16] Hanappi O., Hummer W., Dustdar S. 2016. "Asserting Reliable Convergence for Configuration Management Scripts" Distributed Systems Group, Vienna University of Technology, Austria, 2016 ACM. 978-1-4503-4444-9/16/11. [S16]
[17] Bass L., Holz H., Rimba P., Tran A. B., Zhu Z. 2015. "Securing a Deployment Pipeline" Software Systems Research Group, NICTA School of Computer Science and Engineering, UNSW, Sydney, Australia. 2015 IEEE/ACM 3rd International Workshop on Release Engineering IEEE 978-1-4673-7070-7/15. [S17]
[18] Ur Rahman A. A., Williams L. 2016. "Software Security in DevOps: Synthesizing Practitioners' Perceptions and Practices" Department of Computer Science, North Carolina State University, Raleigh, NC, USA; 2016 International Workshop on Continuous Software Evolution and Delivery, 2016 ACM. ISBN 978-1-4503-4157-8/16/05. [S18]
[19] Wettinger J., Breitenbucher U., Leymann F. 2014. "Standards-based DevOps Automation and Integration Using TOSCA" Institute of Architecture of Application Systems (IAAS) University of Stuttgart, Stuttgart, Germany; 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, IEEE 978-1-4799-7881-6/14. [S19]
[20] Jabbari R, Bin Ali N., Petersen K. 2016. "What is DevOps? A Systematic Mapping Study on Definitions and Practices" ., Blekinge Institute of Technology Karlskrona, Sweden; Binish Tanveer Fraunhofer Institute for Experimental Software Engineering, IESE Kaiserslautern, Germany, XP '16 Workshops, May 24 2016, Edinburgh, Scotland Uk 2016 ACM. ISBN 978-1-4503-4134-9/16/05. [S20]
[21] Ahmadighohandizi F., Systa K. 2015 "ICDO: Integrated Cloud-based Development Tool for DevOps Department of Pervasive Computing, Tampere University of Technology Korkeakoulunkatu 10, FI-33720 Tampere, Finland. [S21]
[22] Nagpal S., Shadab A. 2014. "Literature Review: Promises and Challenges of DevOps" University of Waterloo. [S22]
[23] Erich F., Amrit C., Daneva M. 2014 "Report: DevOps Literature Review" University of Twente, ,DOI: 10.13140/2.1.5125.1201. [S23]
[24] ZHENG J., LIU Y. 2016. "Exploring DevOps for Data Analytical System with Essential Demands Elicitation" School of Software Engineering, Tongji University Shanghai, China; Jin Linz, Shanghai Maitrox Electronics Co., Ltd, Shanghai, China, DOI reference number: 10.18293/SEKE2016-220. [S24]

[25] Cukier D. 2013 "DevOps Patterns to Scale Web Applications using Cloud Services" Department of Computer Science - University of Sao Paulo / Elo7, DOI: 10.1145/2508075.2508432. [S25]
[26] Mohamed S. I. 2016 "DevOps Maturity CaPCulator DOMC - Value oriented approach" Department of ECE, October University for Modern Sciences and Arts (MSA), Cairo, Egypt, International Journal of Engineering Research & Science (IJOER) ISSN: [2395-6992] [Vol-2, Issue-2, February- 2016]. [S26]
[27] Lwakatare L. E., Kuvaja P., Oivo M. 2016. "An Exploratory Study of DevOps Extending the Dimensions of DevOps with Practices" M3S, Faculty of Information and Electrical Engineering, University of Oulu. [S27]
[28] Wettinger J., Breitenbücher U., Kopp O., Leymann F. 2016. "Streamlining DevOps automation for Cloud applications using TOSCA as standardized meta-model" Institute of Architecture of Application Systems (IAAS), University of Stuttgart, Universitätsstr. 38, 70569 Stuttgart, Germany; Future Generation Computer Systems 56 (2016) 317–332. [S28]
[29] Domaschkaa J., Griesingera F., Baura D., Rossinib A. 2015 "Beyond Mere Application Structure Thoughts on the Future of Cloud Orchestration Tools" University of Ulm, Germany, SINTEF, Oslo, Norway; Procedia Computer Science 68 (2015) 151 – 162. [S29]
[30] Schaefer A., Reichenbach M., Fey D. 2013. "Continuous Integration and Automation for DevOps", H. K. Kim et al. (eds.), IAENG Transactions on Engineering Technologies, Lecture Notes in Electrical Engineering 170, DOI: 10.1007/978-94-007-4786-9_28. [S30]
[31] Fowler M., Highsmith J. 2001. "The Agile Manifesto; Software Development, IEEE Software (2001), Cutter IT Journal.
[32] Kitchenham B., Charters S. 2007 "Guidelines for Performing Systematic Literature Reviews in Software Engineering" Ver. 2.3 EBSE Technical Report, EBSE, 2007.
[33] Dyba T., Dingsoyr T. 2008 "Empirical studies of agile software development: A systematic review" SINTEF ICT, S.P. Andersensv. 15B, NO-7465 Trondheim, Norway; (2008)

Appendix A – DevOps Practices

PC1: Create development sandboxes for minimum code deployment. [S6] [S14]
PC2: Automate sandboxes deployment through the development pipeline. [S6] [S8] [S10] [S13] [S22] [S14] [S18] [S20] [S22]
PC3: Provide continuous collaboration system in real-time. [S6] [S12] [S22] [S14] [S20] [S22] [S24]
PC4: Automate testing sandboxes to run in conjunction with development sandboxes. [S6] [S10] [S8] [S20] [S22]
PC5: Perform Retro-QA tests on build sandboxes. [S6] [S8] [S24]
PC6: Keep variance (code/quality/behaviour) between development and production to minimum. [S3] [S6] [S15] [S24]
PC7: Use DevOps tools to automate deployment, build, testing, update, synchronize, continuous deployment of sandbox code. [S6] [S10] [S8] [S13] [S20] [S22] [S27]
PC8: Developers must be able to access the IT operations incident reports and synchronize with operations to improve project supportability. [S6] [S8] [S10] [S15] [S18] [S20] [S22]
PC9: Testing reports (Auto-generated by a DevOps tool, Sandbox test units, quality testing) must be shared between Developers and Operations. [S4] [S6] [S10] [S13] [S15] [S18] [S20] [S22]
PC10: Monitoring logs (Generated by DevOps tools or Retro-QA monitoring logs) must be shared between Development and Operations. [S3] [S6] [S8] [S15] [S18] [S20] [S22] [S24]
PC11: DevOps team synchronizes critical services such as transactions, performance, uptime, deployment schedule, run-time costs, version control, and project scope. [S6] [S8] [S14] [S15] [S20] [S22] [S23] [S24] [S27]
PC 12: DevOps team use central repository for versioning, synchronization of application source code. [S6] [S8] [S14] [S15] [S20] [S22] [S23]
PC13: Application release deployments must be fully automated across the development pipeline. [S6] [S8] [S13] [S14] [S15] [S20] [S22] [S23] [S27]
PC14: DevOps team must provide overall visibility into project scope and release timing to stakeholders. [S3] [S6] [S8] [S9] [S20]
PC15: DevOps team must provide self-service and resources management of platform (Cloud, Hybrid, and Server) to stakeholders. [S6] [S8] [S18] [S20]
PC16: DevOps team must be able to increase release frequency to satisfy business demand. [S6] [S8] [S9] [S18] [S20] [S23] [S27]
PC17: DevOps team must have clear insight into the SD project to ensure business reliability and application performance on a cross-platform environment. [S6] [S9] [S18] [S20] [S23]

PC18: DevOps team must provide safe deployment parameters in order to avoid excessive workload on the infrastructure. [S6] [S8] [S13] [S18] [S20] [S23]
PC19: DevOps team must be able to update system iterations or sandboxes based on monitoring reports and defect logs on any stakeholders' infrastructure. [S6] [S8] [S18] [S20]
PC20: DevOps team optimize SD project based on Behaviour-Driven Development and Retro-QA results of a process. [S3] [S6] [S8] [S18] [S20].

Appendix B – DevOps Tools Catalogue:

Table 8. Source Control Management	
Tools	**Features**
Github https://github.com	- Github is a web-based Git (private and public accounts) repository designed for version control and source code management. - Github provides team collaboration - Provide logs containing (Commit history, tracking labels, pull requests, code review comments, email notifications, task lists, read-me code information file)
Bitbucket https://bitbucket.org	- Similar features to Github - Offers both free public and private commercial accounts

Table 8. Source Control Management

Table 9. Continuous Integration	
Tools	**Features**
Codeship https://codeship.com	- Use Docker abilities to automate development and deployment - Enable developers to create their own test units - Provide team notifications with code changes and test results - Deploy and run code in parallel simultaneously with tests - Integrate many programming languages (Java, Ruby, Python, PHP, GO) - Integrate many platforms (Heroku, AWS) - Integrate various databases (MySQL, MongoDB)
Travis CI https://travis-ci.com	- Used to build, test, deploy code hosted on Github - Enables automated continuous integration with Github - Notify team with test results through email, postings or any IRC channel - Support various programming languages (Java, C, C++, C#, Perl, Python, Ruby, Node.js) - Provide its own command-line UI - Enable parallel deployment and testing

Table 9: Continuous Integration

Table 10. Continuous Deployment	
Tools	**Features**
Codeship https://codeship.com	- Enable multiple deployment sequential or parallel - Enable developers to run deployments commands on an authenticated remote server using SSH. This feature allows developers to trigger deployment/update on external systems for stakeholders.
Travis CI https://travis-ci.com	- Enable developers to setup continuous deployment schedule. - Enable developments to automate deployment schedule. - Integrated deployment with Github.

Table 10: Continuous Deployment

Table 11. IaaS/PaaS	
Tools	**Features**
Heroku	- Heroku is a PaaS that support (Ruby, Java, Node.js, Python, PHP) - Heroku Git server handle application pushes with repository

https://www.heroku.com	- Heroku integrates with Github, Bitbucket - Enables automated continuous deployment - Provide logs and maintain version control of code - Heroku Logplex collects all application reporting

Table 11. IaaS/PaaS

Table 12. Monitoring	
Tools	**Features**
Nagios https://www.nagios.org	- Nagios is an open source application that monitors systems - Nagios also provides remote monitoring through its Remote Plugin Executor which supports SSH and SSL encrypted tunnels. - Nagios enable developers to build reporting units using programming languages (Shell Scripts, C++, Perl, Ruby, Python, C#, etc.) - Nagios also provide a powerful tool for DevOps driven SD applications that consist automated log file rotation and creating in a parallel enabled service distribution.
New Relic https://newrelic.com	- New Relic provides insight into an SD application at runtime. - New Relic delivers unique monitoring log metrics of cloud application development and it deployment from UI to backend. - New Relic provides continuous automated reporting on health, status, runtime, build, deployment and performance or a cloud application.

Table 12: Monitoring

Table 13. Database Management	
Tools	**Features**
MongoDB https://www.mongodb.com	- MongoDB is a free open source, cross-platform, document-oriented database program. - Classified as NoSQL database application, MongoDB avoid tradition table-based relational database in favor of JSON-like documents with dynamic schema. - MongoDB provides developers with Ad hoc queries, Aggregation using MapReduce and Server-side JS.

Table 13: Database Management

Table 14. Logging/Security	
Tools	**Features**
Loggly https://www.loggly.com	- Loggly is a cloud-based log management and analytical service. - Loggly summarizes automatically a software application log and provides real-time analysis for software processes. - Loggly increases delivery speed and provide guided-data log to DevOps team based on application troubleshooting results. - Loggly manages logs from any source or application test units coded in any language (Java, PHP, Node.js, Python, .NET, JS, Docker, Linux, windows, Apache)
Papertrail https://papertrailapp.com/	- Cloud based log monitoring system. - Integrates with Heroku metrics logs - Integrates with HipChat collaborative tool

Table 14. Logging/Security

Table 15. Build	
Tools	**Features**
Codeship https://codeship.com	- Codeship provides build capability for DevOps team form end-to-end in the development pipeline.

Travis CI https://travis-ci.com	- Travis CI provide a powerful build environment that can be setup in .travis.yml

Table 15. Build

Table 16. Testing	
Tools	**Features**
Cucumber https://cucumber.io	- Runs automated acceptance tests written in a behavior-driven development style. - Cucumber merges SD specifications and test documentation into one cohesive log. - Cucumber uses Gherkin, a language that defines Cucumber test cases which is designed to be human readable non-technical.
Junit http://junit.org/junit4	- Junit builds test functions from normal functions by providing *@Test* annotation to the method header. - Automated Test units are composed of collection of annotated Java methods that handle particular exceptions or provide run-time report about a component or process behavior.

Table 16. Testing

Table 17. Collaboration	
Tools	**Features**
Slack https://slack.com	- Slack is a cloud-based collaboration tool it improves DevOps team communication by offering an IRC-like features which can handle files exchange from integrated could such as Trello, Google Drive, DropBox , Heroku, Github, etc.
HipChat https://www.hipchat.com	- HipChat is a web-based service for internal private chat and messaging. - HipChat supports group and one-on-one chats, it also support video calling (group and pair) between team members. - HipChat relays messages through SMS services as well and allows a user a 5GB storage capability. - HipChat integrates the team progress from different repositories such as Bitbucket, Github, etc.

Table 17: Collaboration