Report: DevOps Literature Review

Technical Report · October 2014

DOI: 10.13140/2.1.5125.1201

3 authors, including:

Floris Erich
National Institute of Advanced Industrial Science and Technology

Chintan Amrit
University of Amsterdam

Report: DevOps Literature Review

University of Twente

Floris Erich, Chintan Amrit, Maya Daneva

6 October 2014

Abstract

DevOps is a conceptual framework for reintegrating development and operations of Information Systems. We performed a Systematic Mapping Study to explore DevOps. 26 articles out of 139 were selected, studied and summarized. Based on this a concept table was constructed. We discovered that DevOps has not been adequately studied in scientific literature. There is relatively little research available on DevOps and the studies are often of low quality. We also found that DevOps is supported by a culture of collaboration, automation, measurement, information sharing and web service usage. DevOps benefits IS development and operations performance. It also has positive effects on web service development and quality assurance performance. Finally, our mapping study suggests that more research is needed to quantify these effects.

Contents

1 Background

Many organizations which develop and use Information Systems make a structural division of their software departments. One pattern which is often repeated is the separation between software development and system operations. Lately, there has been discussion about whether this division is warranted. This discussion centers around a concept called DevOps, which has thus far not been frequently discussed in academic literature. We hope to increase understanding of DevOps by reviewing the literature regarding the concept and some closely related concepts.

2 Review questions

The main research question is "How does DevOps influence Information System development performance?"

To be able to answer this, the following questions will need to be answered first:

RQ1 What are the relevant characteristics of DevOps?

RQ2 What effects can be observed when DevOps is being practiced?

RQ3 What are the supporting factors in a DevOps initiative?

RQ4 How are DevOps characteristics instilled?

3 Review methods

I used three search terms: *DevOps*; *"Continuous Delivery" AND Software*; and *"development and operations" AND software*. We applied the search terms to the databases of Scopus, Web of Science, IEEE Xplore and ACM Digital Library.

Papers considered for the review were (1) published in 2007 and onward; (2) related to problems found in the intersection of software development and software operations; and (3) considering development of Information Systems.

I first determined study quality on high level based on the type of the study and the venue in which the study was published. Study types were ranked based on the study design hierarchy for Software Engineering from Kitchenham [73], as described in Table 2. The study venue was ranked as follows (from highest quality to lowest quality): (1) Journal articles; (2) conference proceedings; and (3) industry reports.

Based on the major concepts highlighted in their titles and abstracts, I labeled each article. I discovered the following labels: Culture, automation,

Ref.	Culture	Automation	Measurement	Sharing	Services	QA	Structures & standards
[4]							✓
[6]							✓
[13]	✓	✓	✓	✓			
[15]				✓			
[29]				✓			
[31]		✓			✓	✓	
[34]	✓		✓				✓
[41]	✓	✓				✓	
[42]							✓
[48]		✓			✓	✓	
[55]					✓		
[59]		✓					✓
[67]							✓
[106]			✓				
[80]	✓						
[97]	✓						
[99]		✓					
[102]							✓
[108]						✓	
[112]		✓				✓	
[115]	✓			✓			
[116]				✓			
[121]		✓					
[123]					✓		
[128]	✓						
[136]	✓						

Table 1: Major concepts found in the reviewed literature

measurement, sharing, services, quality assurance and structures & standards. These labels are divided the categories of practice areas (culture, automation, measurement and sharing), application areas (services and quality assurance) and a problem area (structures & standards). Table 1 gives an overview of which labels were applied to which article.

Next, I assessed studies more in depth by considering which instruments were used for reducing bias, and how internal and external validity were maximized. The results of this can be found in the following table.

Ref.	E	T	Description
[4]	5	C	The construction of a System Dynamics model to achieve a "repetitive, risk-free and effortless Continuous Delivery Process" is described. Simulation is used to verify the results. The paper explains how validation could take place, but does not concretely define how it will take place.
[6]	5	J	Disciplined Agile Delivery is presented as an enterprise process for developing software, which includes DevOps. No validation takes place.

Continued on the next page.

Ref.	E	T	Description
[13]	5	C	The usage of Knowledge, Skills and Abilities as a framework for finding employees with DevOps skills is described. No validation is done and the process followed is quite unclear from the paper.
[15]	5	C	Ways for developers to elicit operations requirements from documents instead of face to face communication are described. No validation takes place.
[29]	5	J	How to support system engineering using knowledge management is described. No validation takes place.
[31]	5	C	A case study of using patterns which cross development and operations is described. The results are limited to a single case.
[34]	5	J	An experience report of using techniques argued to be part of DevOps, such as system thinking, is described. No concrete cases are described to validate the results.
[41]	5	J	A case study of the functioning of a company, which has a culture sharing characteristics with DevOps, is described. The results are limited to a single case.
[42]	4	J	A case study of the business implications on adopting DevOps is described. The results are limited to a single case.
[48]	4	C	An approach for applying testing techniques used in software development on operations is described. An experiment trying to prove that this is possible is described, but the experiment lacks rigor.
[55]	5	C	A platform for supporting development based on DevOps is described. No results of validation are presented.
[59]	5	J	It is described how DevOps supports continuous delivery. No concrete cases are described to validate the results.
[67]	5	J	The lack of involvement of operations in DevOps initiatives is described. No concrete cases are described to validate the results.
[106]	5	J	The importance of metrics in a DevOps initiative is argued. No concrete cases are described to validate the results.
[80]	5	R	The history of DevOps is described. Evidence is limited to industry examples.
[97]	5	J	The importance of human factors in a DevOps initiative is described. No concrete cases are described to validate the results.
[99]	4	C	Experience implementing Continuous Delivery is reported. The results are limited to a single case.
[102]	5	J	It is argued that DevOps can be used together with the CMMI and ITIL standards. No concrete cases are described to validate the results.
[108]	5	J	The importance of using DevOps practices in quality assurance is described. No concrete cases are described to validate the results.

Continued on the next page.

Ref.	E	T	Description
[112]	5	J	The use of DevOps practices in an academic setting is described. The results are limited to a single case.
[115]	4	J	The experience using DevOps at a small organization is described. The results are limited to a single case.
[116]	5	C	Logging is presented as a way to improve cooperation between development and operations. To validate some preliminary results, a study is described on high level and its results are presented. There is no validation of the final solution.
[121]	5	J	It is argued that software installation should be handled by automated processes. No concrete cases are described to validate the results.
[123]	4	C	The influence of Software-as-a-Service architecture on the business model is described. Validation was done by studying three cases.
[128]	3	C	Problems related to the cooperation between development and operations are explored. The research is validated by studying a focus group and two cases.
[136]	5	R	A method for building a DevOps culture is described. No concrete cases are described to validate the results.

For performing data extraction, I used the research questions as format. For each study, I described in which way it contributed to each research question.

4 Included and excluded studies

The following table shows which articles where included for each search term.

Ref.	Incl.	Reason
Search term: *DevOps*		
[6]	✓	Contributes to RQ1, RQ3, RQ4
[13]	✓	Contributes to RQ1
[15]	✓	Contributes to RQ1, RQ3
[31]	✓	Contributes to RQ1, RQ3
[34]	✓	Contributes to RQ1, RQ3
[41]	✓	Contributes to RQ1, RQ4
[42]	✓	Contributes to RQ1, RQ2, RQ3, RQ4
[45]		Does not contribute to any research question
[48]	✓	Contributes to RQ3
[57]		Does not contribute to any research question
[55]	✓	Contributes to RQ1, RQ3
[54]		Does not contribute to any research question
[59]	✓	Contributes to RQ1, RQ3
[67]	✓	Contributes to RQ1, RQ3

Continued on the next page.

Ref.	Incl.	Reason
[106]	✓	Contributes to RQ1, RQ3
[80]	✓	Contributes to RQ1, RQ2
[97]	✓	Contributes to RQ1, RQ3, RQ4
[102]	✓	Contributes to RQ1, RQ2, RQ3
[108]	✓	Contributes to RQ1, RQ2, RQ3
[112]	✓	Contributes to RQ1
[115]	✓	Contributes to RQ1, RQ2, RQ3, RQ4
[121]	✓	Contributes to RQ1, RQ2
[136]	✓	Contributes to RQ1, RQ3, RQ4
[139]		Does not contribute to any research question
Search term:	*"Continuous Delivery" AND Software*	
[4]	✓	Contributes to RQ3
[10]		Does not contribute to any research question
[21]		Does not contribute to any research question
[32]		Does not contribute to any research question
[36]		Does not contribute to any research question
[48]	✓	Already considered for search term *DevOps*
[74]		Does not contribute to any research question
[85]		Does not contribute to any research question
[99]	✓	Contributes to RQ3
[104]		Does not contribute to any research question
[127]		Not related to Information Systems
Search term:	*"development and operations" AND Software*	
[1]		Not related to Information Systems
[3]		Not related to Information Systems
[5]		Does not contribute to any research question
[7]		Does not contribute to any research question
[8]		Does not contribute to any research question
[9]		Not related to Information Systems
[12]		Does not contribute to any research question
[13]	✓	Already considered for search term *DevOps*
[14]		Not related to Information Systems
[16]		Not related to Information Systems
[19]		Not related to Information Systems
[18]		Not related to Information Systems
[20]		Does not contribute to any research question
[22]		Not related to Information Systems
[23]		Does not contribute to any research question
[24]		Not related to Information Systems
[26]		Not related to Information Systems
[27]		Not related to Information Systems
[28]		Does not contribute to any research question
[29]	✓	Contributes to RQ3
[30]		Not related to Information Systems

Continued on the next page.

Ref.	Incl.	Reason
[33]		Not related to Information Systems
[35]		Does not contribute to any research question
[37]		Not related to Information Systems
[38]		Does not contribute to any research question
[39]		Does not contribute to any research question
[40]		Not related to Information Systems
[43]		Does not contribute to any research question
[44]		Not related to Information Systems
[46]		Does not contribute to any research question
[47]		Does not contribute to any research question
[49]		Does not contribute to any research question
[50]		Not related to Information Systems
[51]		Does not contribute to any research question
[52]		Does not contribute to any research question
[53]		Does not contribute to any research question
[55]	✓	Already considered for search term *DevOps*
[56]		Does not contribute to any research question
[58]		Not related to Information Systems
[60]		Not available in English
[61]		Not related to Information Systems
[62]		Does not contribute to any research question
[63]		Does not contribute to any research question
[64]		Not related to Information Systems
[65]		Not related to Information Systems
[66]		Does not contribute to any research question
[68]		Not related to Information Systems
[70]		Not related to Information Systems
[69]		Not related to Information Systems
[71]		Not related to Information Systems
[72]		Does not contribute to any research question
[75]		Not related to Information Systems
[76]		Not related to Information Systems
[77]		Does not contribute to any research question
[78]		Not related to Information Systems
[79]		Not related to Information Systems
[81]		Not related to Information Systems
[82]		Does not contribute to any research question
[83]		Not related to Information Systems
[84]		Not related to Information Systems
[86]		Not related to Information Systems
[87]		Not related to Information Systems
[88]		Does not contribute to any research question
[89]		Not related to Information Systems
[91]		Not related to Information Systems

Continued on the next page.

Ref.	Incl.	Reason
[90]		Not related to Information Systems
[92]		Not related to Information Systems
[95]		Not related to Information Systems
[94]		Not related to Information Systems
[93]		Not related to Information Systems
[96]		Not related to Information Systems
[98]		Not related to Information Systems
[2]		Not related to Information Systems
[120]		Not related to Information Systems
[100]		Does not contribute to any research question
[101]		Not related to Information Systems
[103]		Not related to Information Systems
[107]		Not related to Information Systems
[110]		Does not contribute to any research question
[109]		Does not contribute to any research question
[111]		Does not contribute to any research question
[113]		Not related to Information Systems
[114]		Does not contribute to any research question
[116]	✓	Contributes to RQ3
[117]		Not related to Information Systems
[118]		Not related to Information Systems
[119]		Does not contribute to any research question
[122]		Not related to Information Systems
[123]	✓	Contributes to RQ3
[124]		Not related to Information Systems
[125]		Not related to Information Systems
[126]		Not related to Information Systems
[128]	✓	Contributes to RQ1, RQ3
[129]		Not related to Information Systems
[130]		Not related to Information Systems
[131]		Does not contribute to any research question
[132]		Not related to Information Systems
[133]		Not related to Information Systems
[134]		Not related to Information Systems
[135]		Not related to Information Systems
[138]		Does not contribute to any research question
[137]		Not related to Information Systems
[141]		Not related to Information Systems
[140]		Does not contribute to any research question
[142]		Does not contribute to any research question

Rank	Description
1	Evidence obtained from at least one properly-designed randomised controlled trial
2	Evidence obtained from well-designed pseudo-randomised controlled trials (i.e. nonrandom allocation to treatment)
3-1	Evidence obtained from comparative studies with concurrent controls and allocation not randomised, cohort studies, case-control studies or interrupted time series with a control group.
3-2	Evidence obtained from comparative studies with historical control, two or more single arm studies, or interrupted time series without a parallel control group
4-1	Evidence obtained from a randomised experiment performed in an artificial setting
4-2	Evidence obtained from case series, either post-test or pre-test/post-test
4-3	Evidence obtained from a quasi-random experiment performed in an artificial setting
5	Evidence obtained from expert opinion based on theory or consensus

Table 2: Study design hierarchy for Software Engineering from [73]

5 Results

We selected 14 journal articles, 10 conference proceedings and 2 industry reports out of 139 articles found. From the journal articles, 10 originated from the Cutter IT Journal, which released a special issue about DevOps. Table 5 summarizes each article.

5.1 Culture of collaboration

I identified eight papers which related to a culture of collaboration. Each will be shortly discussed in the next paragraphs.

The United States government used Knowledge, skills and abilities (KSA's) as a framework for selecting candidate employees. KSA's can also be used to select proper candidates for a position within an organization using DevOps [13]. But the usage of KSA's was disputed by the United States Office of Personnel Management. In fact, KSA's are no longer used by the US government.

To become more competitive, organizations are building more heavily integrated products, which sometimes even exceed the borders of an organization. Examples of this are device ecospheres such as Android, which runs on mobile phones from various manufacturers, but also runs on tablets, smartwatches, in cars and on various other devices. An approach to reach this integration is by adopting a system thinking approach. The departments of development and operations traditionally collaborate in developing new systems. Both parties are each other's customer and supplier. The development department depends on the operations department for managing systems required for developing large scale information systems. This includes bug tracking, project management and version management software. At the same time the operations department depends on the development department for developing tooling and applications for operating systems, as well as implementing features to improve aspects such

Ref.	Description
[4]	Continuous Delivery processes can be modeled using System Dynamics
[13]	Knowledge, Skills and Abilities allows to find and train employees to use DevOps
[15]	Operations related resources should be studied to elicit operational requirements
[29]	Three-tier knowledge management supports transition of innovations to systems
[31]	Development and operations patterns improve web applications scalability
[34]	DevOps is part of a revolution of organization-wide collaboration
[41]	More employee trust and responsibilities reduces need for operations personnel
[42]	Setting up a central supportive group aids the transition to DevOps
[48]	Applying Behavior Driven Development practices on operations supports DevOps
[55]	DevOps application life cycle toolkit supports software mass customization
[59]	To stay competitive organizations have to improve their IT services using DevOps
[67]	DevOps should more often be considered from the perspective of operations
[106]	DevOps should be supported by a framework of metrics
[80]	To solve problems between developers and operations they need more cooperation
[97]	DevOps should be people focused instead of tool- and process-centric
[99]	Implementing Continuous Delivery can be aided through various lessons learned
[102]	DevOps can be used in with process standards such as CMMI and ITIL
[108]	Quality assurance can use DevOps tools for better system monitoring
[112]	Managing software configurations as source code aids operations automation
[115]	Various tools and practices can be applied to support DevOps
[116]	More attention on logging aids DevOps
[121]	Software shouldn't be installed by hand, this should be managed as source code
[123]	Software-as-a-Service has influenced the business model of organizations
[128]	DevOps is needed to successfully deploy and operate software systems
[136]	There is a process companies can follow to create a DevOps culture

Table 5: Literature descriptions

as performance, security and stability. Because of this, adopting a system thinking approach has a lot of influence on the culture of collaboration [34].

Facebook employs very few employees which dedicate their work to operations. Instead, employees are trusted to and responsible for developing software which performs good in operations [41]. This merging of the work performed by employees to cover both development and operations leads to a different culture of collaboration, in which there is no clear separation between development and operations personnel.

The division of work between development and operations has a long history starting from the early days of computing [80]. What is important to realize when talking about DevOps, is that it does not signal the end for development and operations work. What is happening is that work that used to be done exclusively by development and operations, is now being performed by the other party. This has signified the ambiguity of having a clear division between development and operations.

One of the core values of Agile Software Development is "Individuals and interactions over processes and tools" [17]. When considering DevOps from an agile perspective, this core value should be kept in high regard [97]. Yet, a lot of the discussion around DevOps focuses instead on tools and processes.

Cultural transformations can be daunting. Some best practices to aid in the transformation to a DevOps culture are [115]:

- Consider the cultural change from both the perspective of the driver and the participants.
- When exceptional situations arise, one has to be flexible in temporarily letting go of DevOps values. A process has to be in place to integrate changes once the crisis passes.
- Let the teams decide on which tools they use based on their own skills and expertise.
- Encourage transparency between development and operations personnel.

The process of cooperation between development and operations is not handled optimally. This influences productivity, software quality and service quality. Software process methodologies should be improved to increase this cooperation. More research in this area has to be performed to study how these improvements can be realized. [128]

A DevOps culture is defined by the characteristics of open communication, incentive and responsibility alignment, respect and trust [136].

5.2 Automation

I identified eight papers related to automation. There are two important conclusions related to automation drawn from papers I described in the previous section. First, hiring people with the right knowledge of automation is important to support DevOps [13]. This should be identified by looking at a person's resume. Second, there are a lot of opportunities for automating steps in the software development process. Organizations should trust employees to make the right decisions and make them responsible for automating the process. [41]

The other six papers will be shortly discussed in the next paragraphs.

DevOps automation is supported by various design patterns. Some of these are[31]:

- Use cloud storage for storing big files.
- Process asynchronous jobs using queues.
- Use Platform-as-a-Service instead of Infrastructure-as-a-Service.
- Use memcached user sessions to load balance the application server.
- Use a cloud based e-mail delivery service.
- Use a cloud based logging service.
- Use a Realtime User Monitoring tool.

Two trends supporting agility in software development are Test Driven Development (TDD) and Behavioral Driven Development (BDD). In TDD, developers first write an automated test-case before writing actual production code. BDD is a variant on TDD, in which the test is a behavioral description of a program. In both processes a similar loop is followed:

1. Describe a piece of functionality by writing a test (TDD) or behavior description (BDD).
2. Write production code which realizes the functionality. Use the description from (1) to verify proper functioning.
3. Refactor the code to meet the desired quality. Keep using the description from (1) to verify proper functioning.

In operations, a similar process can be used. Here, the target is not software functionality but system behavior. One way to realize BDD in operations is by using Cucumber-Nagios, a toolkit which allows the writing of Cucumber behavior descriptions, ran using the Nagios network monitoring software. [48]

Bugs in software can cause major problems for an organization, impacting both their reputation and financials. To avoid bugs in released software, organizations often decide to shelve software for a period of time before releasing it to the customer. They believe this shelve period allows more bugs to be discovered. Improving software is the work of engineers testing the software, finding bugs and fixing them. Unlike a good wine, software does not improve on its own.

Continuous Delivery is a process in which functionality is only considered ready when it is being used in practice. Instead of having this shelve period, software goes through a predefined pipeline which ends directly at the customer and involves various automated tests. The false sense of security offered by shelving software is compensated by an ability to release software earlier and more often.

Because this pipeline passes through the boundaries of development and operations, DevOps is a crucial element in realizing Continuous Delivery. [59]

Continuous Delivery should be considered from both an engineering perspective and a business perspective. The engineering perspective focuses on how the software development pipeline is constructed and how it is supported by automated tooling. The business perspective focuses on how various departments play a role in the software development pipeline. [99]

Tools can be used to support automation. This improves scalability and testability, while reducing the work for operations personnel. Companies use varied combinations of tools to support their DevOps effort. For example, at the Friedrich-Alexander-University in Erlangen, Germany, Puppet and Nagios are used to support DevOps practices. This allows the configuration of a network architecture to be supported by tools from software engineering (such as version control) [112].

The setup and configuration of IT systems is complex. This complexity can be managed by approaching IT systems as if they were software components. Configuration tools allow us to manage systems in a efficient and consistent manner. These configuration tools can be controlled by writing configuration rules in a text file. These configuration rules are specified in the form of code. The approach of managing system infrastructure in this way is called Infrastructure as Code. [121]

5.3 Measurement

I identified three papers related to measurement. There are two important conclusions related to automation drawn from papers I described in the previous sections. First, it is beneficial to hire employees with experience in measurement techniques such as thread modeling and risk analysis [13]. Second, the traditional way of measuring employee performance for development and operations personnel leads to friction [34]. Development personnel is measured based on features delivered. This encourages a high pace of delivery. Operations personnel is measured based on system stability. This encourages a slower pace of delivery. This paradoxical situation leads to a lot of friction.

The third paper I reviewed expands upon this second point. It discusses how a metrics-driven framework can be used to define shared and actionable metrics for adopting DevOps. This framework acts as a common language for collaboration between development and operations personnel. [106]

5.4 Sharing

I identified three papers related to sharing. An important conclusion related to sharing drawn from a paper I described in a previous section is that to facilitate sharing, developers and operations should try to make their documentation understandable by both sides. Hiring employees who have experience in both development and operations facilitates this. [13]

What also facilitates sharing of information between development and operations personnel is when developers elicit requirements regarding operations by studying various resources. These resources can be standards, organizational process descriptions, academic studies of operational failures and more [15]. Both product and operation process requirements should be considered in the requirements engineering process.

Knowledge management is a process which facilitates sharing. Three tier knowledge management is a way of representing types of knowledge. The tiers are exploration, evaluation and execution. Exploration concerns the observation of trends in technology and developing plans to innovate and improve. Evaluation concerns the study of feasibility and specification of software. Execution concerns construction, testing and operations of a software product. Because this knowledge management includes both the development and operations of software systems, both development and operations personnel should share the same knowledge management resources. [29]

5.5 Services

Services is one of the areas in which DevOps makes a significant contribution. The concept of services relates to two trends. First, software companies are moving from a product model to a services model. In the past software could be post-ordered, ordered in a shop, or tailor-made software could be ordered. Today, software is often offered to customers as a service or relies on using (sometimes third party) services. Second, resources which in the past were owned by companies are today offered as services. This includes Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS).

I identified four papers related to services. There are two important conclusions related to services drawn from papers I described in the previous sections. First, organizations can use various third party services offered using cloud computing, instead of conventional libraries [31]. This includes for example storage, logging, mail and caching. Second, the communication with third party services can be tested using Behavior Driven Operations (as explained under Automation).

The management of services benefits from using unified frameworks [55]. As services rely on cooperation between development and operations personnel, supporting DevOps principles in service management frameworks is beneficial.

Service based models such as SaaS cause organizations to reassess their business models [123]. Considering DevOps is sometimes considered a competitive advantage, it should be part of the business model as well. The integration of development and operations activities forces organizations to reassess their business models.

5.6 Quality assurance

Quality assurance is another area in which DevOps makes a significant contribution. In Information Systems, quality assurance links together the parties of development, operations, customer support and the customer itself. The task of quality assurance is hence divided over these four parties. Agile software development and DevOps can be seen as both a gift and a curse for quality assurance. Quality assurance is facing a problem: Their work load has become hard to predict. Quality assurance has been affected by the adoption of agile software development. When companies still used a more waterfall-like software development process, quality assurance could easily predict their work load. Organizations released software at fixed intervals, separated by a relatively large amount of time.

DevOps facilitates quality assurance by bringing these parties closer together through cooperation and tooling. Because of this, DevOps offers an opportunity for quality assurance personnel to gather more data than they could in the past [108]. Also, Realtime User Monitoring is a pattern which can be used for detecting problems faced by end end-user early [31]. The responsibility of providing quality assurance can be assigned to an employee who also does development and operations tasks. This is the case at Facebook for example [41]. By using BDOps, the different roles surrounding quality assurance can work together on specifying how the system should deliver the desired user experience [48]. Furthermore, by managing software configuration as source code customers get access to a more consistent experience, software will be available to them earlier and new features will be released more often [112].

5.7 Structures and standards

Structures and standards are an important problem area for DevOps. To support DevOps, organizations will need to make structural changes to facilitate the organization-wide collaboration [34, 59]. At the same time, organizations are afraid that these structural changes are incompatible with standards they use for their organizational processes. Yet, there are plenty of opportunities to combine DevOps with CMMI and ITIL, and the processes actual seem to gel together [102]. DevOps can be incorporated into existing methods, while there are also larger enterprise process models which include DevOps, such as Disciplined Agile Delivery [6]. An effective way of facilitating a transformation to DevOps is by setting up a central supportive group which can give recommendations and advice regarding the DevOps transformation [42]. It is important that such a group, or any person advocating a transition to DevOps, does not see DevOps merely from the perspectives of development, from which DevOps largely oriented [67]. Once a software development process is described, it can be mathematically verified by using System Dynamics [4]. One should however consider the pitfalls in modeling social systems using advanced models from mathematics and physics [25].

6 Discussion

Until now development and operations have mostly been studied as two different fields. However, I believe my research shows that there is some merit in studying the combination of both. This is because in industry, many organizations are reintegrating development and operations. I understand that academic research should not primarily be swayed by trends in industry, which are often the subject of hype. But at the same time, academic research should support industry developments by finding evidence which supports or rejects the value proposition of these developments.

My survey of the academic literature available on DevOps shows that there is some interest in DevOps from an academic perspective in three ways: DevOps is considered as (i) the subject of discussion, (ii) a supporting factor of some other subject (iii) a factor supported by some other subject. Yet, I consider the study design quality of the discovered literature to be low.

A problem frequently discovered in the literature is the lack of a concrete shared definition of DevOps. For example, I have defined it as a conceptual framework, but some authors see DevOps as a job description while others see it a skill set. Research could benefit from a clarification of DevOps by creating a taxonomy [11]. A good starting point for such a taxonomy is the CAMS framework [59]. I suggest that DevOps research can be classified using an extended version of the CAMS framework, including the concepts of services, quality assurance and structures & standards.

I believe the different views of DevOps requires us to look at DevOps from multiple perspectives (see [105] for an example). This allows us to unite the conflicting definitions of DevOps under separate names, such as DevOps as a role in the SDLC process, DevOps as a skill set and DevOps as a conceptual framework for supporting development and operations of Information Systems.

I think the research is particularly vulnerable to two biases, the argument from authority bias and the publication bias. Most articles selected in the review were based on expert opinion. While I have no reason to doubt these opinions, one should be aware that experts can be wrong. When one blindly follows expert opinion, one is vulnerable to the argument from authority bias. That is why expert opinion should be backed up with other sources of evidence. In my research I have found little evidence of DevOps having a positive effect on software development besides expert opinion. The research is also vulnerable to publication bias, which means there is a tendency to publish only positive results. This means that there might be organizations which struggle with DevOps and might have abandoned DevOps adoption, yet nothing is published regarding this. I control for these biases by being aware of them and regularly reflecting on the risks they pose.

7 Conclusions

DevOps is a major change in Information System development. DevOps reduces the gap between developers, operations and the end user, allowing for earlier problem detection. In the past, after each Scrum sprint software would work according to specifications, but these would not be validated by the actual end user. With DevOps, we can implement continuous development and release software to the end user frequently. DevOps also allows developers and operations to work together more efficiently and effectively.

The most important finding of this literature review is perhaps that there is no DevOps process or methodology. DevOps is not a one size fits all solution to solve a problem in software engineering like Scrum is for example. This complicates my work of studying companies practicing DevOps. I am not able to talk about a "DevOps process". DevOps should be considered as an artifact adapted to match the unique environment of an organization under study.

DevOps is a conceptual framework, comparable with Agile Software Development. Organizations need to incorporate DevOps principles and practices in their processes. To accomplish this, they will need to restructure themselves. In this chapter I have given recommendations for organizations to follow when adopting DevOps. This is merely the first step in creating a comprehensive guide for companies to adopt DevOps. There are other sources which organizations can use in their adoption[1].

[1] Some physical resources include The Phoenix Project, DevOps for Developers, DevOps for Dummies and the upcoming DevOps Cookbook. However most activity concerning DevOps is taking place at conferences and in blogs

7.1 Recommendations

For supporting a culture of collaboration:

- Focus on people as the central element of a DevOps culture.
- Hire people with skills in both development and operations.
- Support people in their attempts to adopt DevOps principles and practices.
- Look beyond person's job descriptions and both formal and ingrained roles.
- Take an organization-wide perspective: Look beyond the boundaries of departments and teams.
- Take a systems thinking approach to both the organizational processes as the software development processes.
- Encourage cooperation between development and operations personnel.
- Empower people by trusting them and giving them responsibilities.
- Do not force tools or methodologies on teams, let them pick their own.
- Encourage the adoption of DevOps principles and practices in Software Development methodologies.
- Focus on growing the characteristics of a DevOps culture.

For supporting automation:

- Make an overview of the software development pipeline, incorporating both an engineering and a business perspective.
- Implement continuous delivery up to a point that software can be released on demand.
- Explore and adopt patterns and tools for supporting automation.
- Automate the configuration of systems.
- Automate the testing of network infrastructure.

For supporting measurement, develop shared metrics to reduce the friction between development and operations personnel.

For facilitating sharing:

- Facilitate communication between development and operations personnel to improve understanding.
- Create shared knowledge management systems and encourage their use by both parties.

- Let developers pro-actively explore operation requirements by studying standards, process descriptions, academic studies, etcetera.

For the usage of services:

- Use cloud services in your products to simplify your network infrastructure.
- Incorporate tests for cloud services by using BDOps.
- Make services supporting development and operations, such as version control, accessible for both.
- Consider changes to your business model when adopting DevOps.

For facilitating quality assurance:

- Use Realtime User Monitoring to detect problems early.
- Involve quality assurance with the transition to DevOps.
- Make quality assurance a stakeholder when considering reporting.

For solving problems considering structures & standards:

- Set up a central group to facilitate the transition to DevOps.
- Consider input from both development and operations personnel equally.
- Use standards such as CMMI and ITIL to your advantage.
- Draw inspiration from enterprise processes such as Disciplined Agile Delivery.
- Make forecasts on DevOps primarily from an empirical perspective, avoid the use of mathematical modeling unless there is significant expertise available.

By having done this literature review, I have now created a framework which I can use when studying DevOps in practice. The actions which organizations have taken to adopt DevOps can be categorized under the four practice areas. Next, I can study how these actions influence the application areas. Finally, I can study how each company have tacked the structures and standards problem.

References

[1] E.D. Adamides et al. "Supporting collaboration in the development and management of lean supply networks". In: *Production Planning and Control* 19.1 (2008), pp. 35–52.

[2] "Advanced Software and Control for Astronomy II". In: vol. 7019. 2008.

[3] F.a Aikman, M.b Vincent, and R.a Patchen. “Development and evolution of operational forecast systems for the coastal and Estuarine environment in NOAA’s National Ocean Service”. In: 2008, pp. 671–684.

[4] O. Akerele, M. Ramachandran, and M. Dixon. “System dynamics modeling of agile continuous delivery process”. In: 2013, pp. 60–63.

[5] M.a Akiyama et al. “Development and operation of JACIC/LCDM registry”. In: 2011, pp. 405–410.

[6] S.W. Ambler. “Disciplined agile delivery and collaborative DevOps”. In: *Cutter IT Journal* 24.12 (2011), pp. 18–23.

[7] D. Ardagna, C. Ghezzi, and R. Mirandola. “Rethinking the use of models in software architecture”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5281 LNCS (2008), pp. 1–27.

[8] D.a Ardagna et al. “MODAClouds: A model-driven approach for the design and execution of applications on multiple clouds”. In: 2012, pp. 50–56.

[9] P. Arlos and M. Fiedler. “A method to estimate the timestamp accuracy of measurement hardware and software tools”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4427 LNCS (2007), pp. 197–206.

[10] I.A. Bahrudin et al. “Adapting extreme programming approach in developing electronic document online system (eDoc)”. In: *Applied Mechanics and Materials* 321-324 (2013), pp. 2938–2941.

[11] Kenneth Bailey. *Typologies and Taxonomies: An Introduction to Classification Techniques*. Ed. by Astrid Virding. 1st ed. SAGE Publications, Inc, June 13, 1994. ISBN: 0803952597.

[12] B.a b Balis et al. “Development and execution environment for Early Warning Systems for natural disasters”. In: 2013, pp. 575–582.

[13] S.K. Bang et al. “A grounded theory analysis of modern web applications: Knowledge, skills, and abilities for DevOps”. In: Orlando, FL, 2013, pp. 61–62.

[14] D. Barnhart et al. “Development and operation of a micro-satellite dynamic test facility for distributed flight operations”. In: 2009.

[15] L. Bass et al. “Eliciting operations requirements for applications”. In: San Francisco, CA, 2013, pp. 5–8.

[16] G.a Batori, Z.b Theisz, and D.a Asztalos. “Domain specific modeling methodology for reconfigurable networked systems”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4735 LNCS (2007), pp. 316–330.

[17] Kent Beck et al. *Manifesto for Agile Software Development*. 2001. URL: `http://www.agilemanifesto.org/`.

[18] N. Bencomo, A. Belaggoun, and V. Issarny. “Dynamic decision networks for decision-making in self-adaptive systems: A case study”. In: 2013, pp. 113–122.

[19] N. Bencomo et al. “Genie: Supporting the model driven development of reflective, component-based adaptive systems”. In: 2008, pp. 811–814.

[20] G.Yu. Berdichevskii et al. “Information-diagnostics system-a mandatory component for monitoring the technical condition of water-development works”. In: *Power Technology and Engineering* 43.5 (2009), pp. 275–279.

[21] R.G.a Bias, S.b Lucas, and T.L.c Latham. *The HURIE method: A case study combining requirements gathering and user interface evaluation.* 2007, pp. 163–183.

[22] J.J.a Billings et al. “Designing a component-based architecture for the modeling and simulation of nuclear fuels and reactors”. In: 2009.

[23] A. Boccardi, A. Giubileo, and C. Cadegiani. “New software for OPEX estimation: Cost driver estimation (CODE) tool with montecarlo risk analysis simulation”. In: 2010, pp. 171–176.

[24] A.V. Bogdanov, A. Degtyarev, and E.N. Staokova. “Example of a potential grid technology application in shipbuilding”. In: 2007, pp. 3–8.

[25] Nicholas J. L. Brown, Alan D. Sokal, and Harris L. Friedman. “The Complex Dynamics of Wishful Thinking: The Critical Positivity Ratio”. In: *American Psychologist* 68 (2013), pp. 801–813.

[26] C.J.a Carter and M.J.b Rippa. “Applications of high-rate data logging to telescope system development and operations”. In: vol. 7019. 2008.

[27] C. Cheng et al. “Multi-mission automated instrument product generation implemented capabilities”. In: 2008.

[28] M. Cheng. “The CWB network information exchange environment (NICE)”. In: 2011.

[29] R.D.a Corbin, C.B.b Dunbar, and Q.c Zhu. “A three-tier knowledge management scheme for software engineering support and innovation”. In: *Journal of Systems and Software* 80.9 (2007), pp. 1494–1505.

[30] F. Croce and A. Simonic. “ECSS E-70-32 test platform features and applicability area”. In: 2008.

[31] Cukier. “DevOps Patterns to Scale Web Applications using Cloud Services”. In: *Proceedings - SPLASH ’13.* Indianapolis, Indiana, USA, 2013, pp. 143–152.

[32] S. Datta and R. Van Engelen. “An examination of the effects of offshore and outsourced development on the delegation of responsibilities to software components”. In: *Lecture Notes in Business Information Processing* 16 LNBIP (2009), pp. 73–89.

[33] J.P. Davim. *Mechatronics.* 2013.

[34] D. DeGrandis. “Devops: So you say you want a revolution?” In: *Cutter IT Journal* 24.8 (2011), pp. 34–39.

[35] I.a Derbel, L.L.a Jilani, and A.b Mili. “ACME+ for software architecture analysis”. In: 2013, pp. 429–437.

[36] J.a Diaz, J.a Garbajosa, and J.A.b Calvo-Manzano. “Mapping CMMI level 2 to scrum practices: An experience report”. In: *Communications in Computer and Information Science* 42 (2009), pp. 93–104.

[37] B.a Dillman and A.b Cramp. “Federation testing using mock federates”. In: 2010, pp. 185–192.

[38] G. Disterer. “ITIL conform transition of development projects into it operations”. In: 2010, pp. 137–144.

[39] T. Ebadi, M. Purvis, and M. Purvis. “A distributed and concurrent framework for facilitating cooperation in dynamic environments”. In: vol. 2. 2010, pp. 287–294.

[40] E. Ermilova and H. Afsarmanesh. “An ontology based approach for development and operation of virtual organization breeding environments”. In: 2008, pp. 147–152.

[41] D.G.a Feitelson, E.b Frachtenberg, and K.L.b Beck. “Development and deployment at facebook”. In: *IEEE Internet Computing* 17.4 (2013), pp. 8–17.

[42] L. Fitzpatrick and M. Dillon. “The business case for devops: A five-year retrospective”. In: *Cutter IT Journal* 24.8 (2011), pp. 19–27.

[43] T. Furusawa. “Attempting to increase longevity of applications based on new SaaS/cloud technology”. In: *Fujitsu Scientific and Technical Journal* 46.2 (2010), pp. 223–228.

[44] J.P.a Gareri et al. “Application of scene projection technologies at the AMRDEC SSDD HWIL facilities”. In: vol. 8356. 2012.

[45] I.a Gat and J.D.b Heintz. “From assessment to reduction: How cutter consortium helps rein in millions of dollars in technical debt”. In: 2011, pp. 24–26.

[46] K.a Geihs et al. “A comprehensive solution for application-level adaptation”. In: *Software - Practice and Experience* 39.4 (2009), pp. 385–422.

[47] P. Georgiou and G. Tsakonas. “Digital scholarly publishing and archiving services by academic libraries: Case study of the University of Patras”. In: *LIBER Quarterly* 20.2 (2010).

[48] K. Gohil, N. Alapati, and S. Joglekar. “Towards behavior driven operations (BDOps)”. In: vol. 2011. 2. Bangalore, 2011, pp. 262–264.

[49] Q.a Gu, M.b Parkin, and P.a Lago. “A taxonomy of service engineering stakeholder types”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6994 LNCS (2011), pp. 206–219.

[50] S. Heinz. “Automating the ENERGY STAR submittal process”. In: vol. 3. 2008, pp. 1530–1554.

[51] A. Herzfeldt et al. “A conceputal framework of requirements for the development of e-learning offerings from a product service system perspective”. In: vol. 2. 2011, pp. 1370–1378.

[52] K. Hoshino and Y. Kunii. “Three-layered architecture for teleoperators and its module arrangement”. In: 2012, pp. 615–619.

[53] K. Hoshino and Y. Kunii. “Three-layered architecture with variable task flow for teleoperator”. In: 2013, pp. 500–505.

[54] S. Hosono. “A DevOps framework to shorten delivery time for cloud applications”. In: *International Journal of Computational Science and Engineering* 7.4 (2012), pp. 329–344.

[55] S.a Hosono and Y.b Shimomura. “Application lifecycle kit for mass customization on PaaS platforms”. In: Honolulu, HI, 2012, pp. 397–398.

[56] S.a Hosono and Y.b Shimomura. “Production methods for cloud-compliant web services: A case of service lambda for cloud computing”. In: *Journal of Japan Industrial Management Association* 63.4 (2013), pp. 245–257.

[57] S.a Hosono et al. “Fast development platforms and methods for cloud applications”. In: Jeju Island, 2011, pp. 94–101.

[58] C.-Y.a Huang and M.R.b Lyu. “Estimation and analysis of some generalized multiple change-point software reliability models”. In: *IEEE Transactions on Reliability* 60.2 (2011), pp. 498–514.

[59] J.a Humble and J.b Molesky. “Why enterprises must adopt devops to enable continuous delivery”. In: *Cutter IT Journal* 24.8 (2011), pp. 6–12.

[60] D. Isaychenko. “Life after implementation: Operation-friendly software development”. In: 2009, pp. 148–152.

[61] W. Jaeger and V.H. Sanchez-Espinoza. “Uncertainty and sensitivity analysis for the HELIOS loop within the LACANES benchmark”. In: vol. 1. 2010, pp. 500–509.

[62] S. Jehan, I. Pill, and F. Wotawa. “Functional SOA testing based on constraints”. In: 2013, pp. 33–39.

[63] G.-Y.a b Jiang, W.b Zhang, and A.-D.b Chang. “Data organization method for traffic information acquisition system based on GPS-equipped floating vehicle”. In: *Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)* 40.2 (2010), pp. 397–401.

[64] B. Jilete and P. Munoz. “Earth observation micro satellite constellation for disaster monitoring in Africa”. In: vol. 5. 2011, pp. 3620–3630.

[65] M. John. “Development and operation of space-based disease early warning models”. In: vol. 10. 2010, pp. 8071–8075.

[66] H. Kawanami et al. “Development and operation of speech-oriented information guidance systems, kita-chan and kita-robo”. In: 2011, pp. 558–561.

[67] B. Keyworth. “Where is IT operations within devops?” In: *Cutter IT Journal* 24.12 (2011), pp. 12–17.

[68] M.M. Khabiri. “Geosynthetic material suitable depth staying to control failure of pavement rutting”. In: *Advanced Materials Research* 255-260 (2011), pp. 3454–3458.

[69] M.a Kim, Y.a Kim, and H.b Kim. “Unit testing of flash memory device driver through a SAT-based model checker”. In: 2008, pp. 198–207.

[70] M.a Kim et al. “Formal verification of a flash memory device driver - An experience report”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5156 LNCS (2008), pp. 144–159.

[71] M.a Kim et al. “Pre-testing flash device driver through model checking techniques”. In: 2008, pp. 475–484.

[72] Todd King, Steve Joy, and Raymond J. Walker. “Design, development and operation of a distributed data inventory system”. In: 1994, pp. 287–290.

[73] Barbara Kitchenham. *Procedures for Performing Systematic Reviews.* 2004.

[74] M.a Kulas et al. “Instrument control software development process for the multi-star AO System ARGOS”. In: vol. 8451. 2012.

[75] M. Laiacker et al. “Modular scalable system for operation and testing of UAVs”. In: 2013, pp. 1460–1465.

[76] C. Landauer. “There is nothing so practical as a good theory”. In: 2011, pp. 184–191.

[77] Y.a Li et al. “Evaluating a service-oriented travel portal”. In: 2010, pp. 229–235.

[78] J. Lin and Y. Chen. “The finite element analysis and optimization of micro free piston swing engine’s strain interference”. In: *Advanced Materials Research* 139-141 (2010), pp. 938–942.

[79] K. Liu. “Research of the atomization characteristics of low pollution air blast nozzle”. In: *Advanced Materials Research* 655-657 (2013), pp. 133–136.

[80] Loukides. *What is DevOps?* Sebastopol, CA: O’Reilly Media, 2012.

[81] A.V.a Mal’Ko et al. “Organization of technical-condition monitoring of water-development works at the Svetlinskaya HPP (Vilyui HPP-3)”. In: *Power Technology and Engineering* 47.1 (2013), pp. 22–29.

[82] E.a Manalif, L.F.a Capretz, and D.b Ho. “Software ecosystems risks”. In: 2013, pp. 417–422.

[83] B.S. Manoj and A.H. Baker. "Communication challenges in emergency response". In: *Communications of the ACM* 50.3 (2007), pp. 51–53.

[84] A.a Mantineo, S.a Scaglioni, and E.b Vicari. "Quality assurance/product assurance contribution to cost reduction (ESOC experience)". In: 2007.

[85] J.a Martinez et al. "Software product line engineering approach for enhancing agile methodologies". In: *Lecture Notes in Business Information Processing* 31 LNBIP (2009), pp. 247–248.

[86] K.B.a Matthews et al. "Raising the bar? - The challenges of evaluating the outcomes of environmental modelling and software". In: *Environmental Modelling and Software* 26.3 (2011), pp. 247–257.

[87] M. McCurdy. "Planning tools for Mars surface operations: Human-computer interaction lessons learned". In: 2009.

[88] S. McPherson et al. "Recommendations for enterprise service SLA guidance in the DoD". In: 2010, pp. 948–953.

[89] E.a Mercier et al. "The project office of the Gaia data processing and analysis consortium". In: vol. 7738. 2010.

[90] M. Merri. "Smart software for complex space mission data systems at the European Space Agency". In: 2009.

[91] M.a Merri et al. "Cutting the cost of ESA mission ground software". In: *European Space Agency Bulletin* 130 (2007), pp. 62–69.

[92] C.a Middour et al. "Kepler Science Operations Center architecture". In: vol. 7740. 2010.

[93] N.a b Mohamed et al. "A Service-Oriented Middleware for Building Collaborative UAVs". In: *Journal of Intelligent and Robotic Systems: Theory and Applications* (2013), pp. 1–13.

[94] N.a Mohamed and J.b Al-Jaroodi. "Service-oriented middleware for collaborative UAVs". In: 2013, pp. 185–192.

[95] N.a Mohamed et al. "Middleware requirements for collaborative unmanned aerial vehicles". In: 2013, pp. 1051–1060.

[96] S.H.b Mousavinezhad et al. "Electric energy and power educational programs development workshop". In: 2011.

[97] E. Mueller. "Devops and the people who practice it: Winning their hearts and minds". In: *Cutter IT Journal* 24.12 (2011), pp. 6–11.

[98] Y.a Mukuhira, H.a Asanuma, and M.b Haring. "Correlation between the coulomb stress and occurrence of the large induced seismicity at Basel, Switzerland in 2006". In: vol. 36 1. 2012, pp. 523–528.

[99] S. Neely and S. Stolt. "Continuous delivery? Easy! Just change everything (well, maybe it is not that easy)". In: 2013, pp. 121–128.

[100] F.a Pasian et al. "Science ground segment for the ESA Euclid mission". In: vol. 8451. 2012.

[101] B.G. Penaflor et al. “Custom open source solutions for DIII-D data acquisition and control systems”. In: *Fusion Engineering and Design* 87.12 (2012), pp. 1977–1980.

[102] B. Phifer. “Next-generation process integration: CMMI and ITIL do devops”. In: *Cutter IT Journal* 24.8 (2011), pp. 28–33.

[103] F. Pierfederici, M. Swam, and G. Greene. “Applying decades of HST experience to JWST data processing”. In: vol. 8448. 2012.

[104] M.a Poppendieck and M.A.b Cusumano. “Lean software development: A tutorial”. In: *IEEE Software* 29.5 (2012), pp. 26–32.

[105] Pruijt. “Multiple Personalities: the Case of Business Process Reengineering”. In: *Journal of Organizational Change Management* 11.3 (Jan. 1998), pp. 260–268. DOI: 10.1108/09534819810216283.

[106] A. Le-Quoc. “Metrics-driven devops”. In: *Cutter IT Journal* 24.12 (2011), pp. 24–29.

[107] G.a Rocchitta et al. “Development of a distributed, fully automated, bidirectional telemetry system for amperometric microsensor and biosensor applications”. In: *Sensors and Actuators, B: Chemical* 126.2 (2007), pp. 700–709.

[108] J. Roche. “Adopting devops practices in quality assurance”. In: *Communications of the ACM* 56.11 (2013), pp. 38–43.

[109] M.Q. Saleem, J. Jaafar, and M. Fadzil Hassan. “Security modelling along business process model of SOA systems using modified ”UML-SOA-Sec””. In: vol. 2. 2012, pp. 880–884.

[110] M.Q. Saleem, J.B. Jaafar, and M.F. Hassan. “Model-based security engineering of SOA systems using modified ”UML-SOA-Sec””. In: *Advances in Information Sciences and Service Sciences* 4.9 (2012), pp. 79–88.

[111] R.M. Savola. “Strategies for security measurement objective decomposition”. In: 2012.

[112] A. Schaefer, M. Reichenbach, and D. Fey. “Continuous integration and automation for DevOps”. In: *Lecture Notes in Electrical Engineering* 170 LNEE (2013), pp. 345–358.

[113] A.a Schroeder, M.D.b Van Zwaag, and M.a Hammer. “A middleware architecture for human-centred pervasive adaptive applications”. In: 2008, pp. 138–143.

[114] A.a Sen, K.b Ramamurthy, and A.P.b Sinha. “A model of data warehousing process maturity”. In: *IEEE Transactions on Software Engineering* 38.2 (2012), pp. 336–353.

[115] E. Shamow. “Devops at advance internet: How we got in the door”. In: *Cutter IT Journal* 24.8 (2011), pp. 13–18.

[116] Shang. “Bridging the divide between software developers and operators using logs”. In: *Proceedings - International Conference on Software Engineering*. 2012, pp. 1583–1586.

[117] P. Shi and Y. Shang. "The vibration analysis of eco-friendly vehicle based on the electric motor excitation". In: *Lecture Notes in Electrical Engineering* 201 LNEE.VOL. 13 (2013), pp. 471–480.

[118] P. Skocik and P. Neumann. "Industrial process in laboratory environment". In: 2013, pp. 320–323.

[119] T.C.a Sorensen et al. "A University-developed Comprehensive Open-architecture Space Mission Operations System (COSMOS) to operate multiple space vehicles". In: 2012.

[120] "SpaceOps 2010 Conference". In: 2010.

[121] D. Spinellis. "Don't install software by hand". In: *IEEE Software* 29.4 (2012), pp. 86–87.

[122] J.I. Statman, M.S. Gatti, and D.S. Bagri. "Low-cost operations concept for the DSN". In: 2007.

[123] S.a Stuckenberg, E.b Fielt, and T.a Loser. "The impact of software-as-a-service on business models of leading software vendors: Experiences from three exploratory case studies". In: 2011.

[124] D.-W. Sun. *Infrared Spectroscopy for Food Quality Analysis and Control*. 2009.

[125] G. Sybille et al. "IREQ's innovations in power system simulation [Innovations de l'IREQ en simulation de reseaux]". In: *European Journal of Electrical Engineering* 13.5-6 (2010), pp. 675–698.

[126] P.a Szegedi et al. "With evolution for revolution: Managing FEDERICA for future internet research [Topics in Network and Service Management]". In: *IEEE Communications Magazine* 47.7 (2009), pp. 34–39.

[127] G.a b Tang et al. "Stochastic versus deterministic kernel-based superposition approaches for dose calculation of intensity-modulated arcs". In: *Physics in Medicine and Biology* 53.17 (2008), pp. 4733–4746.

[128] B.a Tessem and J.b Iden. "Cooperation between developers and operations in software engineering projects". In: 2008, pp. 105–108.

[129] C.R.a Theodore and M.B.b Tischler. "Development and operation of an automatic rotor trim control system for the UH-60 individual blade control wind tunnel test". In: 2010, pp. 474–495.

[130] C.R.a Theodore and M.B.b Tischler. "Development and operation of an automatic rotor trim control system for the UH-60 individual blade control wind tunnel test". In: *Journal of the American Helicopter Society* 58.4 (2013).

[131] M.a Thirumaran et al. "A novel framework for enterprise web services change management". In: 2012, pp. 21–27.

[132] C.D.a Turnitsa et al. "Financial implications of Modeling and Simulation standards: Practical aspects and theoretical analysis". In: 2012, pp. 164–172.

[133] J.a Vegh et al. "Core analysis at Paks NPP with a new generation of VERONA". In: *Nuclear Engineering and Design* 238.6 (2008), pp. 1316–1331.

[134] A. Vilenica and W. Lamersdorf. "Benchmarking and evaluation support for self-adaptive distributed systems". In: 2012, pp. 20–27.

[135] L. Vuta and V. Piraianu. "Infoworks WS and epanet v2 - Modeling the water distribution networks". In: *UPB Scientific Bulletin, Series D: Mechanical Engineering* 70.4 (2008), pp. 91–102.

[136] Walls. *Building a DevOps Culture.* Sebastopol, CA: O'Reilly Media, 2013.

[137] C.a Wang et al. "Evaluation method of stealth aircraft's infrared radiation measurement". In: *Hongwai yu Jiguang Gongcheng/Infrared and Laser Engineering* 41.11 (2012), pp. 2891–2897.

[138] H. Wang. "Research of key technologies in development in thesis management system". In: *Advances in Intelligent and Soft Computing* 128 (2011), pp. 317–322.

[139] J.a Wettinger et al. "Integrating configuration management with model-driven cloud management based on TOSCA". In: Aachen, 2013, pp. 437–446.

[140] J. Wu. "Stress testing software to determine fault tolerance for hardware failure and anomalies". In: 2012, pp. 294–298.

[141] S. Wu and P. Chua. "Museum Collection Management on-demand". In: vol. 351. 2008, pp. 310–315.

[142] W.a Ye et al. "DIS system design and in-orbit performance assessment of NHTX-1 SAT". In: *Nanjing Hangkong Hangtian Daxue Xuebao/Journal of Nanjing University of Aeronautics and Astronautics* 44.6 (2012), pp. 797–802.