Cooperation Between Software Development and Operations: A Literature Review

Floris Erich  
Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente  
f.m.a.erich@student.utwente.nl

Chintan Amrit  
Industrial Engineering and Business Information Systems, Faculty of Management and Governance University of Twente  
c.amrit@utwente.nl

Maya Daneva  
Information Systems Group, Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente  
m.daneva@utwente.nl

ABSTRACT

Many software development organizations make a structural division of their software departments. A popular one is separating software development and operations. Organizations have made this division in order to gain higher efficiency by achieving economies of scale and specialization. This paper provides an overview of the research literature related to DevOps. DevOps is a collection of principles and practices that try to improve cooperation between development and operations in organizations which separate these two functions. We found that research about DevOps is still in its infancy despite its widespread adoption in industry. We also find that how companies implement DevOps depends on contextual factors such as culture, level of automation, measurement instruments and how information is shared.

Keywords

DevOps, Continuous Delivery, Development, Operations, Organizational Culture, Automation, Measurement, Sharing, Services, Cloud Computing, Service Oriented Architecture

1. INTRODUCTION

Software development can profit from improvements in the deployment and maintenance phases. Yet little attention is paid to these phases.

DevOps improves these phases through a collection of principles and practices, centered around having developers work closely together with operations. Traditionally, developers paid little attention to issues operations faced. In turn, often organizations discovered issues only after software development transferred system ownership to the operations department responsible for maintenance.

The DevOps movement, a software development and operations professionals community, argues that the departmental division has led to cultural and communication problems. They call the problems caused by the division the DevOps gap. Companies try to address this issue through a set of principles and practices related to culture (C), automation (A), measurement (M) and sharing (S). These concepts together form the CAMS framework [13].

Various companies have started or are planning to start a DevOps effort. These firms develop software for third parties or consume IT services. Research on DevOps is relatively recent and fragmented, making it hard to clearly see and understand its scope, topics covered and challenges addressed as well as those unaddressed. In this article we perform a Systematic Literature Review (SLR) on DevOps.

The main question we like to address as part of this research is "How does the relation between development and operations influence Information System development?" We asked the following research questions:

1. What are the main concepts related to DevOps?
2. What are the problems encountered in Information System development attributed to the relation between development and operations?
3. How can the problems in RQ2 be alleviated by using DevOps?

Our review also aims at helping researchers and practitioners reason about and define DevOps clearly. We hope that in the future a framework for implementing DevOps will be created using the concepts we identified in this research.

We performed our review in the context of a larger project, which also includes interviews with industry professionals. We are producing a detailed technical report of the literature review, which will be released separately.

This paper introduces the research and our major findings. Section 2 describes the method used for answering the research questions. Sections 3 and 4 describe the various results that our SLR produced and Section 5 concludes.

2. RESEARCH METHOD

We performed the research using the guidelines for SLR of Kitchenham [15]. According to her an SLR goes through the phases of planning, conducting and reporting. She also describes subactivities during the planning and conducting phase. For the planning phase, there are two activities: (1) identification of the need for a review; and (2) development of a review protocol. There are five activities which form the conducting phase: (1) identification of research; (2) selection of primary studies; (3) study quality assessment; (4) data
We used three search terms: (1) DevOps; (2) “Continuous Delivery” AND Software; and (3) “development and operations” AND software. We applied the search terms to the databases of Scopus, Web of Science, IEEE Xplore and ACM Digital Library. Selection of articles was done by the first author and was discussed with the co-authors. Only articles subject to the following criteria were selected for the review:

1. Published in 2007 and onward;
2. Relates to problems found in the intersection of software development and software operations;

We used the year 2007 as starting date as that was the year in which the DevOps movement was formed. We performed both a forward search and a backward search using the literature selected. For the backward search we added one more criterion for inclusion of an article. At least two articles already in the study should refer to it. We have set this criterion to increase the quality of the selection. In the forward search we studied all literature referring to the literature already included in the study. This did not lead to any articles being added. As a literature review is concept-centric [30], we constructed a table describing which major concepts we encountered during the review.

3. RESULTS

As Table 1 presents, we selected 25 articles for use in this research. This was out of a total of 151 articles found. We selected 13 journal articles, 10 conference proceedings and two industry reports. After adding an article, we checked whether the scope of the review had to be changed. Based on this we iteratively build the review’s literature selection. We found that there has been little quantitative research performed on DevOps. 19 out of 25 studies were qualitative personal experience reports. We think that DevOps research could profit from more quantitative analyses.

During the research we encountered the CAMS framework [13], which proved to be an effective tool to identify the concepts that each paper of our set of 25, was related to. While doing the research we also encountered some papers related to DevOps which could not be classified according to the popular CAMS framework. We therefore added the concepts of services, quality assurance and structures & standards.

3.1 Answering RQ1

We asked what the main concepts related to DevOps are. We have discovered that these are culture, automation, measurement, sharing, services, quality assurance, structures and standards. The first four of these concepts form the CAMS framework. Companies can use this framework for implementing DevOps [13]. It is a basic categorization of where an organization needs to change. Meanwhile, services and quality assurance are sectors in which DevOps delivers the most benefits. Structures and standards are an area in which more research is required [10, 13, 20].

3.2 Answering RQ2

We asked what problems are encountered in Information System development, which can be attributed to the relation between development and operations. We used CAMS to categorize the encountered problems. Cultural differences between development and operations departments causes clashes [17]. Developers for example have to get used to operation personnel not having experience in working in projects [13]. When development and operations are divided into different departments, some processes cross departmental boundaries. This makes it hard to automate these processes [8]. Measurements between development and operations differ. Development is generally measured on features delivered, while operations is generally measured on the availability of systems under their care. As releasing new features can negatively influence system availability, friction between both departments is created [20]. Finally, little knowledge sharing takes place between development and operations. But both departments benefit from having access to information from the other [5]. However, this is generally made worse by each party having limited knowledge of what is considered important by the other party.

3.3 Answering RQ3

We asked how the problems from RQ2 can be solved using DevOps. DevOps attempts to solve cultural problems by merging the two cultures [29]. Expanding Scrum teams with team members from operations leads to conflicts. For example, if a problem occurs, operations are expected to be available at any time, while developers are not expected to share the same availability [13]. Everyone should be treated equally in a team. DevOps supports automation by removing the departmental barrier and facilitating frequent and clear communication between developers and operators [13]. DevOps makes the whole team responsible for both releasing new features and maintaining system availability [13]. A team is hence given more responsibilities, but also has access to more resources. Closer collaboration between team members should increase its capabilities [9]. Sharing is facilitated by pairing developers and operations [7] or even by placing them in the same team [8].

4. DISCUSSION

Our review revealed that there is no universal term for the intersection between development and operations. This made it hard to find relevant literature. 18 papers used the term DevOps to represent this intersection, 7 papers did not name it explicitly. In general, there are a lot of studies considering development, operations, or even both. However, few studies consider how both sides should work together. At the same time it is hard to define what operations means in various contexts. Sometimes it means maintenance (e.g. in business software), actual use (e.g. in devices such as UAV’s) or something which can be completely automated (e.g. in self-adaptive systems). The papers used in this review see operations as something done by humans.

There was a lot of difference in how the papers studied defined DevOps. Some did not define DevOps at all and a lot of the definitions of DevOps were vague. In some definitions, DevOps is a principle practiced by teams. Other sources state that a specialized team practices DevOps. Still others see DevOps more as a role within teams. Some papers see DevOps as a container for various tools, techniques and/or practices. Research could benefit from a clarification of DevOps by creating a taxonomy [2]. A good starting point for such a taxonomy is the CAMS framework which some papers studied referred to. In such a taxonomy rela-
Operations related resources should be studied to elicit operational requirements. Software shouldn’t be installed by hand; this should be managed as source code. Continuous Delivery processes can be modeled using System Dynamics. Managing software configurations as source code aids operations automation. Applying Behavior Driven Development practices on operations supports DevOps. Development and operations patterns improve web applications scalability. Various tools and practices can be applied to support DevOps. DevOps should be supported by a framework of metrics. DevOps should be people focused instead of tool- and process-centric. To stay competitive, organizations have to improve their IT services using DevOps. DevOps is an actively explored topic. This could not be avoided since English was the only feasible common language for our team. Second, it might be possible that we collectively categorized a paper in a wrong way. The categorization was however reviewed by another senior researcher. We therefore think the risk of this threat is minimal.

6. CONCLUSION

This research was inspired by a large Dutch financial firm’s DevOps adoption, which the first author encountered on a business course a year ago. The first author also experienced the importance of development and operations cooperation while working for a software development company. A lot of time there was spend on communication with external operations, reducing the effectiveness of software maintenance. To answer the main question: DevOps is a major change in Information Systems development.

First, from a practical perspective. Organizations need to incorporate DevOps principles and practices into their processes. They might need to restructure their organization. Even so, organizations can use DevOps together with standards such as ITIL [20].

Second, companies will have to look at various sources for guidance in adopting DevOps. We discovered that there is not a one-size fits all DevOps method which companies can apply. Instead, there are various separate principles and practices, and there are methodologies built around these principles and practices. Disciplined Agile Delivery is an example of this. Other agile methodologies can use DevOps principles and practices. Examples of these are Scrum and

<table>
<thead>
<tr>
<th>Ref.</th>
<th>T</th>
<th>Description</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C</td>
<td>Continuous Delivery processes can be modeled using System Dynamics</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>J</td>
<td>Knowledge, Skills and Abilities allows to find and train employees to use DevOps</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>Operations related resources should be studied to elicit operational requirements</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>4</td>
<td>J</td>
<td>Three-tier knowledge management supports transition of innovations to systems</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>5</td>
<td>C</td>
<td>Development and operations patterns improve web applications scalability</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>6</td>
<td>J</td>
<td>DevOps is part of a revolution of organization-wide collaboration</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>7</td>
<td>J</td>
<td>More employee trust and responsibilities reduces need for operations personnel</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>8</td>
<td>J</td>
<td>Setting up a central supportive group aids the transition to DevOps</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>9</td>
<td>C</td>
<td>Applying Behavior Driven Development practices on operations supports DevOps</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>10</td>
<td>C</td>
<td>DevOps application life cycle toolkit supports software mass customization</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>11</td>
<td>J</td>
<td>To stay competitive, organizations have to improve their IT services using DevOps</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>12</td>
<td>J</td>
<td>DevOps should more often be considered from the perspective of operations</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>13</td>
<td>J</td>
<td>DevOps should be supported by a framework of metrics</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>14</td>
<td>J</td>
<td>To solve problems between developers and operations, they need more cooperation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>15</td>
<td>J</td>
<td>DevOps should be people focused instead of tool- and process-centric</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>16</td>
<td>C</td>
<td>Implementing Continuous Delivery can be aided through various lessons learned</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>17</td>
<td>J</td>
<td>Quality assurance can use DevOps tools for better system monitoring</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>18</td>
<td>C</td>
<td>DevOps can be used in with process standards such as CMMI and ITIL</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>19</td>
<td>C</td>
<td>Managing software configurations as source code aids operations automation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>20</td>
<td>J</td>
<td>Various tools and practices can be applied to support DevOps</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>21</td>
<td>J</td>
<td>More attention on logging aids DevOps</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>22</td>
<td>J</td>
<td>Software shouldn’t be installed by hand; this should be managed as source code</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>23</td>
<td>C</td>
<td>Software-as-a-Service has influenced the business model of organizations</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>24</td>
<td>C</td>
<td>DevOps is needed to successfully deploy and operate software systems</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>25</td>
<td>J</td>
<td>DevOps application life cycle toolkit supports software mass customization</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>26</td>
<td>J</td>
<td>To stay competitive, organizations have to improve their IT services using DevOps</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>27</td>
<td>J</td>
<td>DevOps should more often be considered from the perspective of operations</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>28</td>
<td>J</td>
<td>DevOps should be supported by a framework of metrics</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>29</td>
<td>J</td>
<td>To solve problems between developers and operations, they need more cooperation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 1: Literature selected for the review

In this table the T column shows the type of the article: J stands for journal article, C stands for conference proceedings and R stands for industry report. The last seven columns represent the concepts we have identified to be related to DevOps: (C1) Culture; (C2) automation; (C3) measurement; (C4) sharing; (C5) services; (C6) ; and (C7) structures & standards.
the Rational Unified Process.

Third, once DevOps is adopted, problems can be detected earlier. This is because of: (1) operations being involved in development leads to issues which would have shown during operations of the product being found during development; and (2) user feedback getting enabled by deployments in every iteration. This is quite different from the situation in the past, when after each Scrum sprint the software would work according to specifications. However, one could never be sure if the specifications matched the end user’s wishes.

Fourth, using DevOps, problems can be solved both more efficiently and effectively. Operations have access to actual usage data which they can share with developers. Developers consult operations in solving issues, making sure that the solution will work well in operations.

Finally, a word of caution from a theoretical perspective. We found that little quantitative research has been done to show that DevOps is beneficial. Organizations which do not yet practice an agile method, such as Scrum or Extreme Programming, should avoid implementing DevOps, as the change to DevOps would be too intensive in this case. There is also not a clear overview of DevOps principles and practices. This means companies have to experiment a lot for themselves. Hopefully more intensive research will make it easier for companies to adopt DevOps in the future.

7. REFERENCES


