International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 04 Issue: 10 | Oct -2017 www.irjet.net p-ISSN: 2395-0072

An approach of improve efficiencies through DevOps Adoption

Mohd. Asfar[1], Mr. Raj kumar goel[2], Mr. pradeep kumar[3]

[1]Student of Master of Technology, Department of Computer science & Engineering, APJKTU, Lucknow, India
[2,3]Assistant professor, Department of Computer science & Engineering, APJKTU, Lucknow, India

---

Abstract - Most organizations waste 50% or more of their resources. In larger enterprises and in organizations with compliance requirements, the ratio of productive activities to waste is even more pronounced. Waste comes in several forms: unnecessary overhead, unnecessary rework, unnecessary features, and building the wrong thing. This is a well-established theme of agile methods and lean startup practices. The process measures of the past have proven to be easily gamed and too subjective to measure execution progress accurately. Objective instrumentation of the product pipeline (instead of activity pipeline) leads to more honest progress and quality feedback. Measuring the bottlenecks, throughput, volumes and delay of testable project increments enables more objective steering toward building the right value and more predictable outcomes.

Key Words: **continuous integration, continuous deployment, process based, information technology, automation**

1. INTRODUCTION

Developments and Operations (DevOps) have a significant impact on the global IT sector in the near future. Realizing "DevOps" full potential, IT vendors have been agile enough in providing new products and services under the label "DevOps inside", at an ever increasing pace. [2]

1.1 What is DevOps?

DevOps applies lean and agile principles across the life-cycle and across the enterprise with richer feedback cycles everywhere. Lean transformation enables more efficient delivery and continuous feedback enables more effective steering. DevOps adoption can balance delivery speed with trusted outcomes.

The field in consumer IT is changing quickly. For instance, Apple and Google need to keep pushing out new features in all their services, mobile operating systems and hardware. From major yearly updates to minor quarterly ones, consumers have grown accustomed to expect these releases. Bug fixes and other critical improvements are made even more quickly and deployed at once to millions upon millions of devices and software. If the providers are unable to keep up with this pace, consumers are quick to react and switch to competitors.

1.2 History of IT and DevOps

IT Performance in the context of ITSM and DevOps should reflect the concerns of both development and operations departments. Development organizations are typically measured in terms of *throughput* in delivering code, while operations prioritizes the reliability of the infrastructure and *stability* of the services running on this infrastructure [6]. When measuring throughput, one can think about software development similarly to a manufacturing plant [21], as stated previously. The first throughput attribute we captured is batch size, which we define as the size of a change that affects IT services. In the case of ITSM, the size of a change is difficult to measure, and is one of the biggest problems in applying Lean manufacturing principles to software development [15]. In fact, batch size has been shown to be difficult to measure, particularly across different contexts [e.g., 27]. Therefore, we proxy batch size using deployment frequency, which is how often code is released into production environments. An important part of DevOps is changing the organizational culture from a collection of silos into an openly collaborative way of working. It entails involving Operations personnel in the design and transition process of an application. Furthermore, they should attend necessary planning meetings, retrospectives and showcases of project teams in order to share their insights and knowledge already early on in the process. Humble and Molesky (2011) note that rotating through operations teams is necessary for developers and they should be equally available for root cause analysis and remedy in case of incidents in production.

2. PROPOSED SYSTEM

To present our differentiators in a simple, yet insightful way, and stimulate discussions with clients, we have defined a DevOps Adoption Model and various assets. Here is the overview graphic.

There are 4 adoption paths represented by the 4 rows: Steer, Develop, Deploy and Operate. The gray left hand column shows a stark description of the root cause of inefficiency in most organizations. The middle column represents the initial transformation in each adoption path with a differentiating theme of DevOps adoption:

1. Measure and steer the product, for honest insight into progress and quality
2. Accelerate develop and test feedback cycles through agile methods.

3. Automate the build and release process to enable frictionless deployment.
4. Collaborate consistently across the software supply chain for holistic efficiencies.

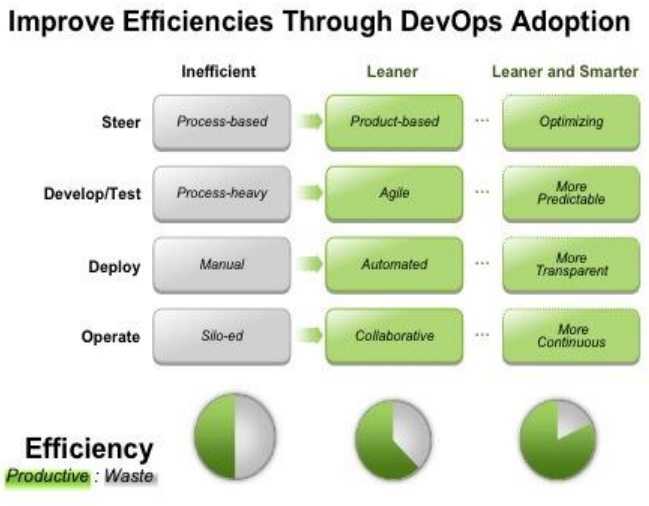
Then the 3rd column illustrates the continuous improvement theme associated with even leaner and smarter outcomes.

1. Optimizing decisions with better steering, continuous feedback and analytics.
2. Increasing the predictability of development with less waste, and better steering.
3. enhence the transparency of deployment updates with automation.
4. Improving the continuity of operations with better quality, fewer defects.

The primary objective is to move an organization to improved execution by improving both efficiency and effectiveness.

Execution has two important dimensions: effectiveness and efficiency.

- **Effectiveness** represents better steering by objectively quantifying value and cost tradeoffs through continuous feedback and better delivery analytics.
- **Efficiency** in execution is achieved through leaner processes, people and platforms.



Your change speed must be an asset, not an anchor.

We all want to passed less time in overhead work like meetings, compliance documentation, late rework, waiting and progress reporting. And we want to spend less time in the drudgery of manual tasks that can be automated. By avoiding these sources of waste AND by steering with continuous feedback and advanced analytics, we can improve the economics of software delivery. Producing more effective value, and doing it more efficiently.

2.1 factors affecting IT performance profile

DevOps practitioners cite the use of Lean methodologies as a way to change the shape of the total cost of change curve, and hence the optimal batch size, through the reduction in transaction costs either directly or through the use of innovations [15, 38]. The innovations offered by DevOps methodologies would occur in the ITSM and software delivery lifecycles in the stages following development: integration, testing, and acceptance (which we refer to as validation). Once accepted, software is either deployed to production, released to manufacturing, or published, depending on whether the software is part of a service, an embedded system, or a user-installed application (including mobile apps) [39]. When viewed through a lens of ITSM, these innovations can be divided up into two categories: validation (testing and approval), and comprehensive use of configuration management in the form of version control (a key part of what is known as "infrastructure-as-code"). DevOps practitioners [e.g., 40] and prior literature [6, 26] have also highlighted the importance of cultural transformation on achieving good throughput and performance outcomes, so we include culture in our analysis as well.

3. EXPERIMENT

3.1 Research Design

The chapter will detail how the research is designed. First, the overall research strategy and approaches taken in the study are explained. The second part describes how data collection was carried out and the reasons for selecting the approach. Finally, in the last subsection the quality of the research is discussed.

3.2 Research Strategy

This research is a factual study of a relatively new approach to software development and delivery in the field of IT. Uusitalo (1991) states that a factual study has to focus on a real world phenomenon and knowledge needs to be acquired with a systematic method. Due to a deficient amount of prior research in the field, the factual data collection for this thesis was carried out by a systemic subjective method. The thesis consists of rich insights to the topic with an aim to describe and clarify the phenomenon. This supports the view of Miles and Huberman (1994). Further, an aim of qualitative research is to describe a phenomenon or an event with the goal to understand a certain activity or give a interpretation to a phenomenon that is meaningful in a theoretic point of view.(Eskola and Suoranta 1996).

Additionally, the value of interviews also lies in the ability to ask for clarification and to find out more about specific issues mentioned. This holds especially true for themes that have not been previously explored. (Hirsjärvi et al., 2009).

Theme interviews are adaptive in the sense they support making follow-up questions and modifications if an interesting topic comes up. The aim of this research was to find out and understand potential challenges in the field of DevOps. Therefore, targeting questions and predefined discussion topics were decided with the overall goal of the research in mind (Tuomi & Sarajärvi, 2006).

3.3 Data Collection

The data collection was carried out by semi-structured theme interviews with nine Finnish IT-professionals with previous in-depth experience of DevOps initiatives. The interviews were structured by the CALMS-model described in the theory part of this thesis. Each section of CALMS was supported by targeting questions. However, conversations were allowed to develop freely in any direction in order to learn more about the different aspects of DevOps as well as to uncover new sources of challenges and points-of-view.

3.3.1 Interviews

The study consists of nine semi-structured theme interviews with industry experts. Additionally, two of the experts come from companies that work with DevOps transitions as a business. While the number of interviewees is low, they have extensive knowledge on the subject and some of them can be considered key players in the field of DevOps in Finland. Gomm et al. (2009) state the quality of the study is not influenced by statistical significance. Taking a deep scope, rather than scratching the surface, allows to concentrate on the reasons and the ways to overcome challenges in DevOps implementations, as supported by Dubois and Gadde (2002).

The data collection interviews were carried out between February 19th and March 8th of 2016. The interviewees were contacted through LinkedIn and e-mail and they were carried out at company premises or their nearby restaurants. On two occasions, two people were interviewed simultaneously. The length of the interviews varied between 47 minutes and 1 hour 7 minutes. The average length of the interviews was 59 minutes. The interviews were held in Finnish and were recorded on the permission of the interviewees. The interview questions were based on the key areas of DevOps, aiming to guide the interview.

3.3.2 Interviewees

A total of nine people were interviewed in the data collection phase. The interviewees were included by first identifying DevOps initiatives in Finland (Commitcom, 2014). Next after getting responses from the first interviewees, I got recommendations from their personal networks for additional people. All interviewees are top professionals in the Finnish DevOps-scene and have been deeply involved with DevOps transformations in their respective companies.

The remainder of this section will show an overview of the interviewees and their involvement with DevOps.

Erno Aapa

Erno Aapa is the founder of DevOps Finland, a community of 750+ members attending DevOps-related meetups and promoting discussion about the approach. Originally a developer, Aapa has experience in DevOps consulting to companies such as Rovio and Elisa. Currently Aapa is working at Sharper Shape with focus on management and team leadership.

Lauri Halkosaari

Lauri Halkosaari is the CTO/CIO of Schibster Media / Tori.fi. With a backround in full-stack development, web services and project management, Halkosaari was leading the DevOps transformation at Fonecta before going to Schibster.

Pasi Katajainen

Pasi Katajainen is the Head of Technology at Nordcloud, a leading cloud architecture and DevOps consultancy company in Finland. Originally a developer, Katajainen has experience at Nokia's HERE-unit in managing its cloud transformation where he got acquainted with challenges of Continuous Integration and DevOps generally. Nowadays working closely with DevOps-workshop-based solutions.

4. ANALYSIS

This section analyzes the information derived from the interviews. The thesis uses a data driven analysis for the purpose of creating a theoretic body of the qualitative data that is acquired by means of theoretical research. Data driven analysis can be divided into three phases (Miles & Huberman 1984). This research leans on these three phases in processing the acquired data.

1) Reduction of the data (simplification)
2) Clustering of the data (categorization)
3) Abstraction of the data (creating theoretic concepts)

In order to reduce the data into manageable entities, I first identified the broad concepts of different challenges and linked similar stories from different individuals together. I created an Excel-spreadsheet to manage the data and prioritized the challenges according to the frequency they were mentioned in interviews.

Next, the data was clustered into different categories. There were 16 sub-challenges identified. Further grouping resulted in four general-level challenges that are listed below:

Challenge 1: Lack of Awareness in DevOps

Challenge 2: Lack of support for DevOps

Challenge 3: Implementing DevOps technology

Challenge 4: Adapting organizational processes to DevOps

The following section describes each channel in detail.

4.1 Lack of Awareness in DevOps

As anticipated, the lack of clear definition of DevOps brings forth confusion. The concept of DevOps is not well understood, since the concept has not yet sufficiently matured. Also since DevOps is such a hot word currently, a certain amount of resistance to 'buzzwords' was also identified in the interviews. The lack of awareness inside an organization creates bottlenecks and people-dependencies.

4.1.1 Maturity of the concept

One of the main challenges in bringing DevOps to an organization was found to be the actual concept of DevOps. The lack of definition for DevOps was found to be confusing in six of nine interviews. This creates problems in getting organizations aboard to start embracing the approach and selling DevOps-consultancy services to customers.

Another common misconception was that DevOps is either a "DevOps guy or a DevOps team" (Katajainen) or a "separate operations team that is doing some coding" (Ylä-Anttila). Neither is it a role or a title (Aapa). The problem is that everybody is using DevOps the way they see it and how their organization has been inspired by it. Aapa stated that the usage of the term is also dependent on the size of the organization: for startups DevOps might entail radically different things than for large enterprises. For instance, startups are usually born cloud-native and are operating on a pure PaaS infrastructure, for example Heroku [3], which is closer to a NoOps-model where all operations are automated and handled by the service provider. In contrast, large enterprises might operate on a mixture of self-hosted and IaaS infrastructure that requires more configuration management, coding infrastructure and automation. To conclude, DevOps for startups might mean pure development with no operations while DevOps for enterprises might mean the whole product development process going even further over the boundaries of both development and operations to involve product owners as well as quality assurance.

4.1.2 Allergy to buzzwords

There are several interviewees who see a problem with the DevOps-buzzword. Partly due to the fact that DevOps is currently such a hot word and partly due to its ambiguity and "wild" usage, the value of the word has decreased and DevOps is now met with a certain kind of reluctance. Aapa mentioned that the "allergy" can be seen in people that are naturally against trend phenomena, people who think "it is too mainstream, I am against it even though I do not know what it is really about". This is important to mention because it affects how to sell the idea and how it is received on the other end.

Many of the interviewees suggest a strategy of dropping the umbrella term 'DevOps' and talking about reducing lead time and introducing automation instead, while others say the "hotness" of DevOps actually works to its advantage or that the reluctance can be used to educate the other party.

4.1.3 Lack of Awareness

DevOps is a new concept for many people. As mentioned before, it is a challenge to grasp the whole concept of DevOps. According to Vainio, for management, it might not be even necessary to understand DevOps but for the people inside the IT organization where the change is taking place, the concept needs to be shared and communicated in order for everyone to work together with the level of collaboration DevOps requires for it to work.

The overall lack of awareness about DevOps is viewed as a challenge by several interviewees. Even though the majority of the organization might already be DevOps-oriented, there are always areas in the organization where DevOps has not reached yet. Not limited to the concept of DevOps, the components such as Continuous Delivery or Continuous Integration may be unknown.

4.2 Lack of Support for DevOps

Another central challenge identified was the lack of support for DevOps. The lack of support may show on several levels: on the management level, on the team level or as an overall lack of trust.

4.2.1 Lack of management support

Lack of management support is considered one challenge in DevOps. Because DevOps has to do with lots of changes to the ways different teams are working on a daily basis, a high commitment and support at the top level of the organization is necessary to change the company culture. Klemetti says that The role of the managers is to "untie the knots executive personnel cannot untie themselves", to "break down the walls" and allocate more time for communication and sharing.

Managers open to ideas are usually willing to discuss an idea but they are more concerned about the financial side of the changes. It might be challenging for a development organization to translate the improvements into cost reductions and other financial gains. Therefore, argumentation must be well thought out for it to pass.

Fortunately, DevOps can be tried out with small initial commitments. Finally, managers with teflon profiles are the most difficult to deal with. In many cases, these managers are not even aware of the issues a DevOps initiative is trying to solve. Thinking that the organization is doing just fine with current methods is quite common, or the notion that current systems are so complex they cannot be developed in the proposed way. These kinds of endeavors may still end up in failure, Klemetti said, giving an example of an anonymous company where the main sponsor for DevOps was lost when the management saw the person as a burden and decided to let the person go, even though the person contributed to DevOps in that organization significantly.

Table 4: Manager profiles as described by Klemetti

Manager profile	Description	Strategy to gain support
Tech-savvy	Prior knowledge of DevOps, open to new	Reasonable argumentation
Open to ideas	No prior knowledge of DevOps, open to	Argumentation based on
Teflon	No prior knowledge, no realization of	"Guerilla warfare"

The lack of "real" support from management is also an issue once the decision to adopt DevOps has already been made. Pasi Katajainen stated that more often than not, management does not understand the scope of the change. Simply throwing money at DevOps does not lead to much, rather there are different aspects that the management needs to address. First of all, according to Katajainen, change is often mistakenly viewed as a R&D change where, for instance, an automation pipeline is set up. Pasi Katajainen explained that the change in culture needs to be facilitated by empowering different boxes with different communication tools and practices and seeing this as a value-adding measure: not just wasted time away from "actual" development work. Along with culture, staff needs to be properly trained to new processes and tools and it needs to be determined whether any 3rd party consultants are needed in any part of the change process. Moreover, the nature of product development changes from iterative work to continuous development and therefore, for example feature specifications need to generated accordingly.

4.2.2 Lack of team-level support

Lack of support can also be seen on the team level. The main reasons for lack of support lie in the change of working methods, change of roles and the organizational readiness to communicate. The change of working methods is perhaps the most profound change in DevOps. As mentioned in the literature review, developers take more responsibility in running software, operations take more responsibility in building coherent, preferably code-driven infrastructure and quality assurance can focus on actual quality and not manual testing. As many tasks are automatized, the fear of a machine replacing jobs is a common one that raises opposition on the executive level.

4.2.3 Lack of trust

The challenge is the clearest at the very top level of the organization. In the traditional model, the services are developed in steering committees and CAB meetings and other agreements where deployment dates are set and approved. However, Lauri Halkosaari noted that in a DevOps approach the batch sizes are smaller and releases more frequent so many of these meetings can only deal with more abstract, higher level changes. This shifts responsibility more towards the development team and the teams are getting more and more autonomous. In the process, a lot of the information and documents produced by the development teams to bring to these meetings are not produced anymore (such as weekly reports mentioned by Lauri Halkosaari).

4.3 Implementing DevOps technology

The key technological breakthroughs in DevOps is the creation of an automated continuous delivery pipeline. This new way of delivering software has a profound impact on the processes in development, quality assurance and operations. "Automatizing inefficient processes leads to automatizing inefficiency." (Jyrki Kasvi) Sandström used the quote to point out that setting up automation technology alone is not sufficient. While automation promises benefits in lead times, application stability etc., implementing this technology and processes is not entirely challenge-free.

4.3.1 Automated testing

7 of 9 interviewees found challenges with building automated testing inside the development process. Many of these challenges are people- and skill-related, while others are more related to the way testing is organized in the new approach. As a result, the need for manual click-based testing is reduced, testing time is minimized and possible breaking errors can be noticed more easily since tests are run more often and in the exact same manner each time.

One challenge mentioned by several interviewees was the attitudes of developers towards writing automated tests. According to Katajainen, writing tests is not considered fun nor a part of developers' role. It might even be considered waste. Finally, developers might not have adequate skills to do testing properly. For example, the lack of knowledge and know-how of different testing frameworks is quite common but somewhat surprisingly there have been cases where some developers did not know automated testing frameworks even existed.

4.3.2 Automation tool challenges

Five out of nine interviewees identified challenges with automation tools, namely determining which tools are right for the project and the maturity of automation tools. The challenges with automation software tools were not considered technical - the setting of these tools to work in a proper continuous deployment pipeline was considered a laborious task but not a challenge per se.

somebody highlighted the main decision points: will the company go with open source tools that are easier to replace or will they commit and pay for these tools? Additionally, should the tools be hosted locally in some environment or should they be cloud-based? Are there restrictions to which tools an organization can use? For instance, if an organization is already using a test framework, is it "DevOps-friendly" and if not, can it be replaced?

4.3.3 The type of the application

The final set of challenges in automation is related to the type of the application that is the target of automatization. Three interviewees of nine identified this challenge. The most common problem is that the application architecture is not at all suitable for virtualization.

4.3.4 Fragmentation of tools and practices

Another related set of challenges lie in common practices both inside and across teams. First of all, the use of various technologies and tools across teams is considered a challenge. Pasi Katajainen and Vainio claimed that having different environments for continuous integration and not having a common deployment pipeline is an anti-pattern for the very foundation DevOps is built on.

There are several ways of overcoming this. Keeping all application code in version control and available for everybody to audit and review makes all application development transparent and prevents reinventing the wheel effectively. Moreover, people from all teams can contribute to the code if they need a specific functionality implemented. This is especially true with infrastructure. Aapa explained that rather than making a different version of the infrastructure to use with a specific application, the enhancements made are available for all other applications that use the same infrastructure template as well. Subsequently, it prevents fragmentation and people dependencies.

4.3.5 Finding the right scope for monitoring

With the help of DevOps and new tools such as the cloud-based New Relic, it becomes easier to monitor different services in a consolidated view. However, when more and more operations duties are shifted towards development, "No-Ops" being at the far end of the spectrum, it becomes also increasingly important to monitor the right things.

4.4 Adapting organizational processes to DevOps

DevOps promises speed and flexibility in the delivery of software. The speed and flexibility might not be fully realized, however, if the organizational processes do not accommodate it. The processes have to do with DevOps initiation, software development mode, change management processes, metrics and team-related challenges.

4.4.1 Starting with the correct scope

A challenge that was frequently mentioned was the notion of starting with the correct scope in regard to DevOps transition. As explained in the literature review, DevOps calls for the virtualization of IT infrastructure. For existing applications, it means the server stack needs to be converted to code that is able to provision the functional equivalents of previous hardware servers in order to run the application. For new applications the process is much easier: starting with a clean slate the infrastructure can be configured without any limitations using the latest available technologies and industry standards. Furthermore, the monetary investment is a significantly cheaper than in a traditional setting since no new hardware is usually needed.

4.4.2 The mode of Software Development

There was a very common challenge associated with Scrum as an Agile software development method. The main challenge with Agile, according to Pasi Katajainen, is that many organizations are developing the software in sprints without the goal of releasing the features immediately when they're finished. "You can do agile forever without ever releasing your software. Agile itself does not translate to added value for customers", Pasi Katajainen explained. Moreover, Aapa claimed that doing Scrum by the book is too rigid for the purposes of DevOps. The model where a release package is created at the end of each (2-week) sprint does not fit well into the DevOps-approach where the goal is to release once a feature is completed, often several times a day. This is the reason why Kanban fits into the DevOps approach better.

4.4.3 Change Management Processes

The other major challenge with DevOps-style continuous delivery is how it fits into the organizations current change and release management processes. Very often enterprises follow for example ITIL processes by the book. The processes are not necessarily designed to handle the amount of changes and the velocity of development DevOps can bring to an organization.

4.4.4 Adopting new metrics

Finding common metrics for both development and operations as well as quality assurance was considered challenging. Traditional metrics such as uptime and the amount of tasks completed in a Scrum sprint, for instance, are still viable but the focus is shifting. The "DevOps" metrics such as lead time, code quality and overall system health can be measured with new kinds of tools. Once the metrics are in place, it may take time for the numbers to be comparable with other applications. According to Pasi Katajainen, there are managers who might be over-enthusiastic with these new metrics. When these metrics are first put in place for an application that has been in development for several years, it is natural for these metrics, such as code quality, to be below average. The challenge is finding the balance between improving the score of the metrics and keeping up a rapid pace in development. Over-emphasizing the metrics can have an adverse effect on the development since a lot of time goes to refactoring code and making the application perform better from a metrics-point of view.

4.4.5 Team challenges

When development and operations are brought together to form DevOps, the idea is that the amount of collaboration and communication between the two increases. Several aspects, such as location, the time spent together with other teams and product teams have some associated challenges.

5. CONCLUSION

A new concept such as DevOps is always met with a certain amount of skepticism in the beginning. For me, DevOps seemed like a great suggestion for improvement of rigid processes in an organization, although grasping the concept in its entirety took a considerable amount of time. It wasn't until the very final interview with Pasi Katajainen when I thought I knew enough about DevOps to actually write a thesis about it. Bearing this in mind, it's no wonder that understanding DevOps also proved to be one of the main challenges in the findings.

REFERENCES

[1] Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. Computer, 34(9), 120-127.

[2] Hirsjärvi, S., Remes, P., & Sajavaara, P. (2009). Tutki ja kirjoita (13th ed.). Helsinki: Tammi.

[3] Hummer, W., Rosenberg, F., Oliveira, F., & Eilam, T. (2013). Testing idempotence for infrastructure as code. Middleware 2013, 368-388. Berlin Heidelberg: Springer.

[4] Menzel, G (2015) DevOps - The Future of Application Lifecycle Automation. A Capgemini Architecture Whitepaper – 2nd Edition

[5] Halkosaari, L. (2016) Interview at Schibster. Helsinki 19.2.2016.

[6] Humble, J. (2012) There is No Such Thing as a "DevOps Team". Thoughtworks blog. Online. Available at: https://www.thoughtworks.com/insights/blog/there-no-such-thing-DevOps-team [09.04.2016]

[7] Hussaini, S. W. (2014, October). Strengthening harmonization of Development (Dev) and Operations (Ops) silos in IT environment through Systems approach. Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference, 178-183. IEEE.

[8] Hüttermann, M. (2012). DevOps for developers. New York City: Apress.

[9] Highsmith, J. A. (2002). Agile software development ecosystems (Vol. 13). Boston: Addison-Wesley Professional.